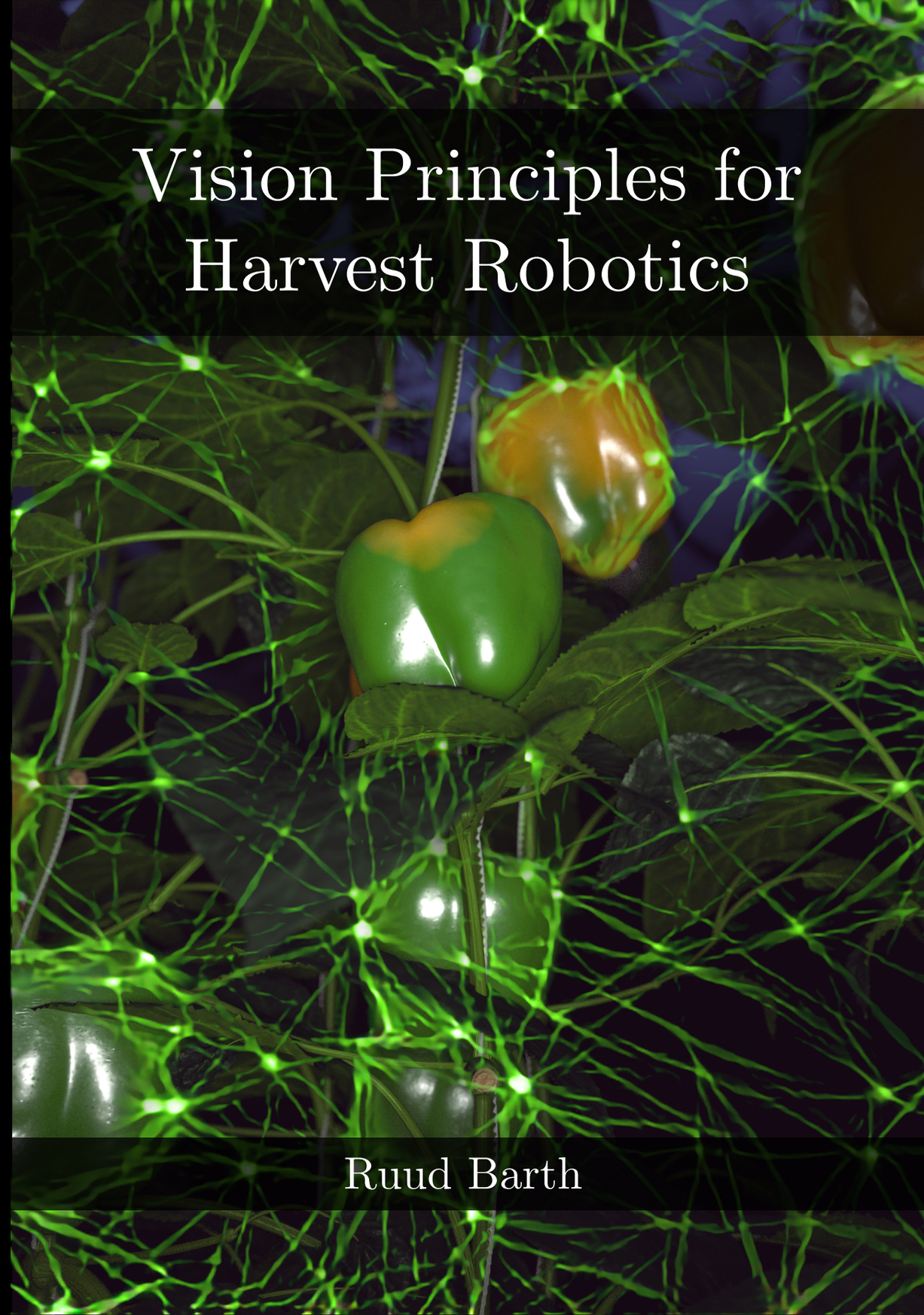
Vision Principles for Harvest Robotics

Ruud Barth

Vision Principles for Harvest Robotics



Ruud Barth

Propositions

1. Agricultural robotics will not solve the societal problem of increased caloric demand caused by the expected population growth.
   (this thesis)

2. Synthetic data generation will become superfluous when unsupervised learning takes root.
   (this thesis)

3. Human visual learning is possible because the perceivable world is very simplistic.

4. Although artificial intelligence now outperforms humans on many specific tasks, non of those systems match the learning abilities of the human brain.

5. Automatisation should not be feared for causing the loss of jobs but praised for the time they free us from working.

6. Psychology will become the most important science of the 21st century.

Propositions belong to the PhD thesis entitled,

`Vision Principles for Harvest Robotics'

Ruud Barth
Wageningen, 30 oktober 2018

Vision Principles for Harvest Robotics

sowing artificial intelligence in agriculture



Ruud Barth

Thesis committee

Promotor

Prof. Dr E.J. van Henten
Professor of Farm Technology
Wageningen University & Research

Co-promotor

Dr J. Hemming
Senior researcher, Business Unit Greenhouse Horticulture
Wageningen University & Research

Other members

Prof. Dr J. Molenaar, Wageningen University & Research
Prof. Dr H.P.J. Bruyninckx, KU Leuven, Belgium
Dr D. Perrin, Harvard University, Boston, United States of America
Dr M.J.G. van de Molengraft, Eindhoven University of Technology

This research was conducted under the auspices of the C.T. de Wit Graduate School of Production Ecology & Resource Conservation (PE&RC)

Vision Principles for Harvest Robotics

sowing artificial intelligence in agriculture

Ruud Barth

Thesis

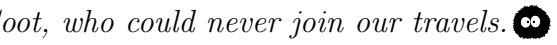
submitted in fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus
Prof. Dr. A.P.J. Mol,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on Tuesday 30 October 2018
at 11 a.m. in the Aula.

Ruud Barth
Vision Principles for Harvest Robotics
327 pages.

PhD thesis, Wageningen University, Wageningen, NL (2018)
With references, with summary in Dutch and English

ISBN 978-94-6343-318-1
DOI: https://doi.org/10.18174/456019

to Lianne, who with me endured the academic journey and journeys, from the early beginnings to Stanford, where coal broiled hamburgers were seducing through the pinewood air, to the autumn travels to Harvard where all cedar homes infused the atmosphere inspiringly, and then the visit to the CalTech that breathed revitalizing eucalyptus amongst blue skies and palm trees.

to Soot, who could never join our travels. 

Contents

		Page
Chapter 1	Introduction	5
Chapter 2	Design of an eye-in-hand sensing and servo control framework	41
Chapter 3	Data Synthesis Methods for Semantic Segmentation in Agriculture: a *Capsicum annuum* Dataset	75
Chapter 4	Synthetic Bootstrapping of Convolutional Neural Networks for Semantic Plant Part Segmentation.	119
Chapter 5	Improved Segmentation Performance by Optimising Realism of Synthetic Images using Cycle Generative Adversarial Networks	171
Chapter 6	Estimating Angles between Fruit and Stems to Support Grasping in a Sweet-Pepper Harvesting Robot	203
Chapter 7	General Discussion, Reflection and Recommendations	251
Chapter 8	Summary	277
Chapter 9	Samenvatting	289
Chapter 10	Acknowledgements	301
	Curriculum Vitae	313
	List of Publications	317
	PE&RC Training and Education Statement	323
	Colophon	327

"What labor is more severe,
what science is more wearisome,
than botany?"

Professor Carl Linnaeus



**Three-dimensional model of *Capsicuum Annuum*, textured with surface normals.
Used as a source for each chapter image.**

Chapter 1

Introduction

1.1 Aim of This Thesis

The objective of this work was to further advance technology in agriculture, specifically by pursuing the research direction of agricultural robotics for harvesting in greenhouses. Within this scope, it was previously determined that the primary cause of agricultural robotics not yet maturing was the complexity of the tasks due to inherent variations of the crops, in turn limiting performance in harvest success and time. As a solution, it was suggested to further enhance robotic systems with sensing, world modelling and reasoning, for example by pursuing approaches like machine learning and visual servo control. In this work, this suggestion was followed by researching vision principles for harvest robotics.

1.2 Prologue

Before addressing the scientific contents and contributions of this work to the research community in the following chapters, first a background is provided that brings a concise historical context of the relation between agriculture and technology. Combined, these subjects form the underlying theme of this thesis.

The background is by no means exhaustive, but does supply a compact motivation for the relevancy and necessity of the pursued research field and direction. Although in essence science is about truth finding, increased understanding or perhaps more pragmatically about progression in a specific domain, it is important to also place the work in a contemporary societal perspective and define the challenges to be faced. Therefore, this chapter might be considered more subjective in nature, as opposed to the remainder of the thesis that fully adheres to the constraints of conventional scientific discourse.

As a part of the background, an attempt was made to unravel the issues regarding the global food supply and production. However, it must be noted that by no means this can be fully addressed within the limited bounds of this thesis, as many factors are entangled in their complex interactions and only a partial perspective from an outsider can be given. The truth isn't as clear-cut and deserves more nuance by their respective experts.

1.3 Background

1.3.1 History of Agriculture & Technology

For about 90% of our species' history, humans have moved with the seasons and their food supply as nomadic hunters and gatherers (Lee & Daly, 1999). As the earth's last glacial period ended about twelve thousand years ago (Walker et al., 2009), the landscape became more moderate and in turn created a more geographically consistent supply of animals and plants. It is thought that this change allowed humans to move less and initiated a transition towards a more semi-nomadic way of living, setting up temporary seasonal camps (Stringer & Gamble, 1993).

Gradually, the development of agriculture, defined as the cultivation of organisms, made a significant impact on the way by which and scale at which the human species provided itself with its essential living needs. Agricultural development allowed for a full transition from a nomadic lifestyle towards sedentary and permanent local settlements. To enable this transition, a certain level of technology was needed not only for food production, but also to preserve and store harvests for periods of scarcity. This was a considerable investment, especially when taking into account that up to this point the natural ecosystem had generally been able to provide adequate and sustainable resources throughout the year - albeit for a limited population size. This technology-facilitated transition is thought to have occurred relatively simultaneously in about a dozen independent locations (Larson et al., 2014). At first, animals were tamed and fed with local wild grains and it is thought that only later plant crops such as wheat and rice were domesticated.

Since the establishment of agriculture, technology, defined as the practical application of accumulating knowledge, has been a key factor in improving agriculture by increasing harvest yields, quality and diversity. In turn, the latter allowed societies to not only grow in size, but also to settle in less affluent locations. Furthermore, it has enabled people to spend time on novel economic activities and leisure. It is a common myth though that technology correlates with leisure time. As the economy grew and diversified, the cultural norm of what essential needs comprised shifted as well. During the nomadic period, it was estimated each individual spent four hours of essential labour per day (Kaplan, 2000; Sahlins, 2009). In pre-industrial medieval times, peasants did see an increase in labour to about eight hours a day (Rogers, 1949), although about one-third of the year did not include any work due to holidays. With the rise of capitalism in the industrial era, holidays were almost all cut and the 70 hour workweek became the mean (Zeisel, 1958).

Regardless of this trend of increasing labour efforts, the required time spent on agricultural activities decreased over time from 75% during agrarian societies to 2% in the developed countries today (Turner, 2006). Unfortunately, the advancements of technology have not distributed equally yet. In developing nations about a third of the total available labour is still required for agriculture. However, it must be noted that the commonly used distinction between *'developed'* and *'developing'* world could be a remnant of the previous century, as according to some statistics the world has become far less polarised (Rosling et al., 2018).

1.3.2 Societal Issues in the Developed World

Although for developed countries it could be argued that further agricultural optimisation and advancement might be excessive, there remains room for further improvements that reach beyond (perhaps shallow) goals such as reducing labour costs or increasing yields for profit alone:

- There remains a part of agricultural labour that is of low quality from the perspective of the employee, e.g. being monotonous to the point of causing cognitive strain and physical wear. For example, in Dutch high-tech greenhouses the majority of labour consists of plant manipulation such as harvesting fruits or removing leafs (Jukema & Van de Meer, 2009). These tasks are generally performed under harsh climate conditions to allow optimal yields. The poor labour quality has led to societal problems. In developed countries, it becomes increasingly hard to source people that are willing to endure this type of labour and a current trend can be observed of the national workforce declining the work altogether. Consequently, international and cheaper labour is attracted from less developed economic regions. When in time also these regions economically catch up, the labour problem is further displaced. The above described problems are directly related to the generally low wages in the agricultural sector, as a higher reimbursement might incentivise local people to perform the work or could allow part-time labour to reduce strains. However, due to globalisation this solution is not economically viable, as international competition with regions with less development and lower wages could result in the elimination of certain types of local food production. Furthermore, also these regions are likely to catch up economically eventually, then facing the same underlying problem of poor quality labour. Protectionism might safeguard for higher synthetic wages and ensure local production, although this does not address the fundamental labour quality concerns.

- Further agricultural optimisation could result in lowering resource requirements and could thereby reduce the environmental and climate impact of agriculture. Resources like water, fertilisers, pesticides but also land have a significant environmental pressure and carbon footprint (Davis et al., 2016). Optimising resource use efficiency can lead to a higher level of sustainability. For example, with the advent of closed greenhouse technology (Opdam et al., 2005), water use efficiency can be increased (Katsoulas et al., 2015) or even zero-emmission can be achieved (Beerling et al., 2017).

- As a third improvement, higher quality products and fewer losses in the food production chain can be achieved by increased agricultural optimisation. For example, regarding greenhouse fruit and vegetables crops, changes in light intensity, temperature, vapour pressure deficit and carbon dioxide concentration have effects on visual characteristics, texture, vitamin and mineral content of produce (Gruda, 2005). Market-driven grading and harvesting could also reduce losses (Bac, 2015).

1.3.3 Global Societal Issues

If one looks beyond issues in developed countries, where population growth appears to stabilise (Espenshade et al., 2003), the observed challenges for food production are perhaps overshadowed by the effects of projected population growth in developing countries. In 2050, the world society is estimated to have increased with 35% of the current population. In combination with the expected diet shift of the developing world from cereal based meals to include meats, the global crop production could be demanded to double (Godfray et al., 2010). It can therefore be expected that the pressure on agricultural development will continue to rise in order to meet this increasing demand. This global issue could be addressed by the following, non-exhaustive list of means:

- Along the total production and distribution chain, about a quarter of the cereals, half of the fruit and vegetables and a quarter of all meat is lost (Gustavsson et al., 2011). For developing countries this is mostly early in the supply chain, due to poor infrastructure, transportation and absence of refrigeration. For developed countries, most losses are later in the chain due to (over-)selective sorting during food processing, poor environmental conditions during display and with consumers' lack of planning and limited focus on waste. Reducing these losses could relieve a part of the expected pressure on agriculture.

- There is a global imbalance in the distribution of food. Although about a billion people are currently suffering from obesity, another 800 million are underfed and another 2 billion lack essential nutrients (von Grebmer et al., 2014). If food would be distributed more evenly, (mal)nutrition issues could be improved, more people could be fed and this would result in a reduced demand of agriculture.
- Changing the diet to reduce or exclude meat consumption. The caloric efficiency of meat production is on average about 7.5% (Shepon et al., 2016). Furthermore, up to 2,700 Mha of pasture and 100 Mha of cropland could be re-purposed for either non-meat food production or as carbon uptake sinks by regrowing vegetation (Stehfest et al., 2009). If all the grain currently fed to livestock in the United States alone would be subverted for human consumption, it could feed close to an additional 800 million people. Globally, it is estimated that the world's cattle consume food equal to the caloric needs of 8.7 billion people (Gold, 2004).

1.3.4 Technologic Answer

Regarding the global societal issue of increasing agricultural pressure, the previously proposed solutions are for a large part non-technological. Applying technology directly could be another partial solution. For example, produce yield and nutrient quality can be improved by applying the right type of artificial light (Olle & Viršile, 2013; Lin et al., 2013).

Concerning technology for meat production, in-vitro meat cultivation could increase the caloric efficiency of meat production (Post, 2012) and therefore reduce the burden on the use of resources by agriculture.

In the case of the most significant crops like maize, rice, wheat, and soybean, on average the yearly yield growth due to direct technology is about 1,2% (Ray et al., 2013), e.g. by increasing resource use efficiency through genetic engineering (Christou et al., 1990; Hu & Xiong, 2014). Unfortunately, this growth rate itself is not sufficient to meet the expected future caloric demand.

Regarding societal issues in the developed world and to improve the aspects of labour and product quality, production loss, resource use and climate impact, technology can also be one of the solutions.

One of the proven agricultural technologies is mechanics, which enables automation of a wide range of crop tasks, performing them faster, more accurately and efficiently whilst on a larger scale than solely with human labour. Since the beginning of agriculture, mechanical tools have been developed and continuously improved. For example mechanical aids to work the soil, ranging from the first known plough around 2800 B.C. (Lal, 2003) to the current tractors pulling disk ploughs. Other tasks, like post-processing the harvest into consumables were also mechanised, e.g. threshing the grains using a flail in the beginnings (Zakiuddin et al., 2012) to the modern combine which integrates harvesting and threshing into a single machine.

Furthermore, the way mechanics were powered changed over time, resulting in increasingly higher speeds and capacity of the machines. The first mechanical tools were powered by human hands, after which domesticated animals such as workhorses and oxen took over. In time, water and wind powered mechanics automated a subset of the agricultural tasks, for example the gristmill for grinding grain into flour (Wilson, 2002). Later, machines got powered using steam engines in the industrial era, upgraded to fossil fuel powered mechanics that are used in modern times. Currently, fully electric agricultural machinery are being researched.

To further specify mechanical technology, five general system levels are distinguished on a range of the dimension of (artificial) intelligence. These levels can be mapped to natural systems, or their mechanical counterpart ranging between *classical* mechanics (I) to self-aware robotics (IV).

0 Systems that passively interact with the environment. For example in mechanics, the plow or in nature, a virus.

I Systems purely reactive on basic sensory input. For example in mechanics, precision spraying based on the presence of green leaves or in nature, chemotactic bacteria such as *Escherichia Coli* (Delbrück & Stent, 1988).

II Systems with the extension of sensors and memory, including software that is able to form a world representation and can accordingly act *autonomously*. For example in mechanics, unmanned aerial vehicles with cameras to navigate through orchards or in nature, bees performing a similar task.

III Systems that can form a theory of mind, i.e. the ability to attribute beliefs, intents and desires to itself and other agents. For example in mechanics, the fictional Cyberdyne 800 Terminator, or in nature (though to a slight degree) great apes (Krupenye et al., 2016) or dolphins (Tomonaga et al., 2010).

IV Systems that can form not only a theory of mind but also representations about themselves and possess self-awareness. For example in nature, humans.

For the domain of agriculture, the current commercially available mechanisation falls primarily in Levels 0 and I. Regarding solving the aforementioned societal issues, a significant part could be determined by improving the present agricultural mechanisation technology to Level II, also coined as agricultural robotics.

1.3.5 Agricultural Robotics

Agricultural robotics may further improve efficiency by the premise of automating complex tasks that could not previously be solved through Level I classical mechanics and thus currently still require human labour (van Henten, 2006; Pekkeriet & van Henten, 2011; Blackmore et al., 2005). However, few to no Level II systems have reached the full market yet and they remain in development, like autonomous tractors (Aravind et al., 2017). Only a handful of pilot robots are currently to a limited extent deployed, for example in protected strawberry cultivation (Hayashi et al., 2010a) or in field spraying (Blue River Technology, 2018). Although many solutions are not yet mature, the demand for harvest robots is currently present regardless. In a recent survey amongst 1300 growers (Alpha Brown, 2018), 27% indicated considering purchase of robotics to aid in harvesting. Amongst greenhouse growers, the interest was the highest at 34%.

Specifically for harvest robotics, a recent review of 50 systems concluded that no system has matured enough to be practically used (Bac et al., 2014). The reported average performance in this domain for localisation success was 85% and a detachment success rate of 75%. When all factors were combined, the average harvest success was 66% with a cycle time of 33 s. Remarkably, the performance did not seem to have increased the past thirty years.

Previous research (Bac et al., 2014) concluded that the following causes, ordered by priority, can be identified:

- Variation on many levels in the crop and environment makes harvesting a difficult task.
- The context or environment was rarely adapted to simplify the task of the robot.
- Lack of detailed robot requirements were set and therefore no optimisation towards constraints was made.
- Lack of detailed performance measures were defined and therefore determining bottlenecks was hard.
- Lack of detailed descriptions of best practices and used hardware restrained progress in the research domain.

1.3.6 Conclusion

The substantial societal issues surrounding (both sufficient and high quality) global food production pose significant pressure on rapid agricultural technological improvement. As highlighted before, although no single approach will likely be able to solve the issues in full, advancing agricultural technology is a promising route to alleviate at least *some* of the issues of global food production. This thesis specifically aims to contribute through transcending classical mechanics to agricultural robotics, thus facilitating a higher level of artificial intelligence in agricultural robotics.

To reach this goal, the primary bottleneck for performance should be first addressed, namely finding mechanisms to cope with the variation in the crop and environment. One previously suggested approach, was to further enhance robotic systems with more advanced sensing, world modelling and reasoning. Therefore, in this thesis, this suggestion was followed by researching vision principles for agricultural robotics.

On a critical note, one must be careful not to overstate the priority of new technologies like agricultural robotics to solve some of these issues. Although their impact is hard to quantify yet, other answers that are targeted at meat consumption and food waste might contribute far more significantly. Unfortunately, these two topics seem to be big elephants in the room. Regarding meat consumption, the current consumer price[1] does not provide an incentive to change and might not accurately reflect the real (e.g. due to subsidies)

[1] €3.00 per kilogram of pork and €1.20 per kilogram of chicken, January 2018, Pasadena, CA, USA.
€5.25 per kilogram of beef and €4.50 per kilogram of chicken, January 2018, Nijmegen, NL.

and future (e.g. environmental) cost. Claims that agricultural robotics can and will solve the issues solely, should be taken with grains of salt.

Agricultural robotics is likely not to contribute significantly to meeting the future caloric demand, as the main relevant crops (soy, wheat, rice and maize) are already highly automated. One must search for its primary contribution in crops that provide other essential nutrients, e.g. fruits and vegetables. Agricultural robotics is therefore about ensuring the quality of food and nourishment of a growing population. Secondary to that, agricultural robotics could further improve on labour quality and contribute to reaching an exemplary equilibrium of sustainable agricultural production. In that sense, this new level of mechanisation can become an invaluable part of our future society. Perhaps in time, after a global zero population growth is established (Davis, 1973) and nation development has equalised through acceleration of technological transfer (Krugman, 1979; Onwude et al., 2016), most pressure on agriculture can be finally relieved.

1.4 Scientific Context

Before addressing and delineating the underlying reasoning for the work in this thesis in the next section, first a brief historical scientific context is provided in this section on the surrounding technological topics of this thesis, notably on i) agricultural robotics, ii) computer vision in agriculture and iii) as it is an important requirement for the latter; the modelling of plants.

1.4.1 Agricultural Robotics

One of the earliest mentions of the suggestion of research in agricultural robotics seems to have been made around three decades ago. Around that time, the possibility of increasing productivity through improved handling technology was discussed, concentrating particularly on converting industrial robotics for agriculture (Arndt, 1985). It was thought that such robotics research would center around the development of sensors and control software to reduce labour requirements that couldn't yet be solved by classical machines (Isaacs, 1986). Hence, the general idea of upgrading level I mechanics with a basic form of artificial intelligence to level II systems, was born (Baylou, 1987).

Looking back, this new research domain started out with great ambition, as shown by statements that practical intelligent agricultural robotics should already be possible (Sistler, 1987). Expectations were set for agricultural robotics to become common within the next decade. Time proved that although indeed classical level I mechanics were upgraded across the board with all kinds of sensors and software, a breakthrough in either a sufficient level of autonomy or speed was not achieved (Bac et al., 2014).

However, within that first decade, research on essential components for agricultural robotics did begin to emerge. For example, to interpret the information provided by image sensors, a procedure was developed for computing the orientation of bell peppers (Wolfe & Swaminathan, 1987). Shortly after, such information was already successfully coupled to robotic manipulators, e.g. in the application of vision guided planting of dissected micro-plants (Tillett, 1990).

It was at the end of that decade that the perception on the ambition changed. For fruit harvesting, the added requirement was raised that although robotics might be technically feasible, such a system must be economically sound to form an alternative for current (level I) mechanical harvesting systems or manual picking (Sarig, 1993). The challenge

of developing a cost-effective robotic system became a new requirement for agricultural robotics that even until this day remains difficult to meet.

As the scope in this thesis is pointed towards improved automated harvesting in greenhouses, the advancements of such systems are briefly reviewed with examples in the following sections.

Cucumber Harvesting Robot

After the first decade in the early nineties, in Wageningen the research commenced on the robotisation of the harvest of high-wire greenhouse cucumber (*Cucumis sativus*, cv. Korinda) (Arima & Kondo, 1999; Van Henten et al., 2002). An autonomous vehicle was designed that included a robotic manipulator, a specialised end-effector for grasping and cutting the fruit, a control scheme that generates collision free motions for the manipulator during harvesting and two computer vision systems for fruit detection and 3D imaging of the fruit.

First and foremost, during the design the economics were taken into account. One of the main defined requirements was a harvest cycle time per fruit of at most 10 seconds. In turn, this would translate to the need of one robot per hectare during peak season. However, there is a direct correlation between what a robot may cost and how fast it operates[2], given a certain fixed investment space that growers might have. As a shorter cycle time would result in fewer robots needed, this in turn would also allow for the application of more advanced and therefore more costly technology to be used. Vice versa, a larger cycle time pressures robots to be produced at lower cost.

This trade-off between, for example, cost and cycle-time is often set arbitrarily. The balance between engineering and economics can be seen as somewhat of a catch-22 dilemma of causality (e.g. *'How much a robot may cost depends on its cycle time, but the cycle time depends on how much the robot may cost'*). Nonetheless, the grower's space for investment remains an important guideline for the requirements of economical design. First and foremost, a system must be proven to function after which research can iterate to reach a viable solution.

[2] Other factors also play a role. Amongst others, depreciation time is directly correlated with cost and maintaince or residual labour costs correlate with reliability and harvest success performance respectively.

At the turn of the century, the cucumber harvesting robot was proven to be technically a success (Henten et al., 2003). The computer vision system was able to detect more than 95% of the cucumbers and by using geometric models the ripeness of the cucumbers could be determined. The A^* based motion planner assured collision free eye-hand coordination towards the fruit. Under unmodified greenhouse conditions, the robot achieved a harvest success rate of 74%, without any human interference.

However, regarding the requirement of time, on average the robot still needed 65 seconds to harvest a cucumber, resulting in an uneconomical design. Faster hard- and software for image processing and motion planning was thought to be required as well as a reduction of the motion time of the manipulator itself.

Sweet-Pepper Harvesting Robot

A decade later, as hard- and software costs decreased whilst their speeds were increased significantly, another moment arrived to attempt to address the economical requirement of atomisation. One of the efforts made, was by a European consortium within a 7th Framework Programme for Research and Technological Development project called Clever Robots for Crops (*Crops*) (Wageningen University and Research, 2018a).

The goal of *Crops* was to develop a highly configurable, modular and clever carrier platform that includes modular parallel manipulators and intelligent tools (e.g. sensors, algorithms, sprayers and end-effectors) that can be easily installed onto the carrier and are capable of adapting to new tasks and conditions. Several use-cases were pursued for high value crops like those in greenhouses (sweet-pepper), orchards (apples), and vineyards (grapes). Two main tasks were investigated for automation, namely site-specific spraying and selective harvesting of fruit.

The performance of the automated harvest of sweet-peppers was investigated (Bac et al., 2017). An economic analysis for robotised Dutch sweet-pepper harvesting showed that a cycle time of six seconds would be required, corresponding to a robot price of €100,000 and a 50% harvest success rate (Pekkeriet, 2011). After development, a greenhouse performance evaluation was held and showed an overall harvest success rate of 6% with an average cycle time of 94 seconds. Unfortunately it was therefore concluded that this part of the project's goal had not been reached, whilst mentioning the cause to be the inherent complexity of the sweet-pepper crop. Simplifying the crop conditions (e.g. removing leafs and fruit clusters) indeed increased performance to 33%, although this might not be a realistic real-life crop requirement to achieve production yields.

Attempts by others during the same timeframe on robotic sweet-pepper harvesting (Lehnert et al., 2017; Sa et al., 2017), used a similar approach where the crop was first scanned to obtain a 3D scene whereafter an optimal grasp pose was determined for an open-loop motion control system towards the fruit. The system achieved on average a 46% harvest success rate for unmodified crop, and 58% for modified crop, with a picking time of 40 seconds on average. Barring the difference in cultivation system with the one that was targeted by *Crops*, these results indicate technical feasibility towards (partial) robotic harvesting of a complex crop such as sweet-pepper, although time-wise an order of magnitude of improvement remains required.

Other investigations of robotic greenhouse harvesting around the same period, achieved comparable results as for sweet-pepper under simplified conditions. For tomato (*Solanum lycopersicum*), a robot was developed to harvest whole clusters of fruit in high density crops (non high-wire) (Kondo et al., 2010). A harvest success rate of 50% was achieved, although cycle time was not reported. For strawberry (*Fragaria* × *ananassa*), a robot achieved on average a 41.3% harvest success rate with a cycle time of 11.5 seconds per fruit (Hayashi et al., 2010b).

Follow-up Sweet-Pepper Harvesting Robot

Another effort in sweet-pepper was launched in the H2020 project *Sweeper* (Wageningen University and Research, 2018b) by a consortium of European partners. As opposed to *Crops*, the work only focussed on one use-case and one task. The aim of the research was to advance individual system components from the *Crops* project with technological readiness level 6 (technology demonstration) towards an integrated market-ready system with level 9 (system test and launch). At the time of writing, this project is still running. Most reported efforts of this thesis are part of the *Sweeper* project.

The main differences of this pepper robot as compared to the one in *Crops* are i) the use of an industrial manipulator, ii) implementing an eye-in-hand strategy for local fruit scanning, iii) visual servo control (Barth et al., 2016) and iii) the implementation of deep learning for plant part segmentation to facilitate grasping (Barth et al., 2017, 2018a) and iv) the aim of autonomous navigation in the greenhouse.

1.4.2 Agricultural Computer Vision

Since the first emerging computer vision approaches for agricultural robotics three decades ago, methods that interpret digital images have become a key sensing technology. The first reports on this new sub-domain were on sweet-pepper orientation detection (Wolfe & Swaminathan, 1987), market quality analysis for sorting tomatoes (Sarkar & Wolfe, 1985), peach and apple detection in orchards (Sites & Delwiche, 1988) and vision based guidance of self-driving tractors (Reid & Searcy, 1987).

Specifically for plant or fruit image detection, a recent historical survey of image processing techniques showed that up to a few years ago the majority of methods was based on colour index-based or threshold-based segmentation (Hamuda et al., 2016; Kapach et al., 2012) (similar to the exemplary approach in Chapter 2). Such basic methods have significant drawbacks however. Setting the right index or threshold for segmentation works primarily for static image distributions with high color and/or intensity differences between classes. When the scene is not static these methods tend to fail, e.g. changes due to illumination conditions, differing backgrounds or because the plant parts change color through the season, or due to wetness or fruit ripeness.

One way to overcome these drawbacks is to apply learning based methods that can tune to dynamic, or multiple input distributions. Furthermore, such methods are often able to generalise to new situations. These approaches, such as for example supervised or unsupervised machine learning where previously sparsely applied in computer vision for agriculture. In the beginning, unsupervised methods were popular, e.g. fuzzy clustering on excessive greens (Meyer et al., 2004). Later, initial learning methods like (classical) artificial neural networks were applied for green segmentation (Zheng et al., 2009).

Meanwhile, the parent domain of computer vision as a whole developed more advanced machine learning methods for image recognition and segmentation, by finding overlap with the domain of artificial intelligence. Most notably at first random forests for image classification (Bosch et al., 2007) and later convolutional neural networks (CNN) (Krizhevsky et al., 2012). The latter has been the predominant approach for computer vision since (LeCun et al., 2015).

With some delay, deep learning has been steadily adopted for agricultural computer vision applications, e.g. for plant species classification (Yalcin & Razavi, 2016), classifying Fusarium wilt of radish from unmanned aerial vehicles (Gwan et al., 2017) or weed detection using synthetic images (Pearlstein et al., 2016).

The work presented here made an effort to further apply state-of-the-art developments in deep learning to agricultural computer vision, beyond detection or segmentation of plants alone. An attempt was made to increase the class resolution towards segmenting individual plant parts and to reduce the requirement of manual annotation of images.

1.4.3 Plant Modelling

An important part of this thesis comprises the modelling of plants, which in itself is a fairly recent addition to botany or the scientific study of plant physiology, structure, genetics, ecology, distribution, classification, and also economic importance.

In the work presented here, plant models will be used to provide additional high-level domain knowledge to allow pre-training of machine learning algorithms that require large datasets, such that recognition of plant parts can be accelerated and improved.

Long before botany was formalised, plants were first collected and traded for centuries. One of the earliest remaining accounts of the hunt for plants was by queen Hatshepsut, the 5th pharaoh of the 18th dynasty of Egypt, who ordered five ships in 1482 B.C. to the Land of Punt to collect eleven specimens of (*Boswellia*) frankincense trees (Tyler-Whittle, 1970). Recordings of this travel were made by mural carvings in the queen's temple gardens at Karnak. Today, wooden stumps of these trees still remain at her mortuary temple.

It was not until the fourth century B.C. that the first (known) plant catalogues appeared. Initiated by Aristotle's observations in nature, the science of botany was founded. However, his original works were lost. His writing did survive through a summary and extension by one of his pupils, Theophrastus. His works, *Historia Plantarium* from about 300 B.C are the earliest surviving attempt to describe the uses and classifications of plants, based on how they reproduced.

Besides physically collecting plant specimens or writing about them, plants have also been botanically described in illustrations. Around the year 60, Pedanius Dioscorides wrote the first known book of herbals and their medicinal power and included an illustrated botanical work (Pedanius & Beck, 2005). Botanical illustrations offer a form of scientific description, which are not necessarily photorealistic, but are stylised in order to highlight unique plant features that enable a botanist to distinguish between between plant species. In Figure 1.1, one of the first botanical illustrations of *Capsicuum Annuum* is shown.

It was not until 1532 that the herbalist Ulisse Severini da Cingoli created the first known physical plant collection. Such herbaria were created with the intent to index and preserve specimens for scientific study. In Figure 1.2 a herbarium sample of *Capsicuum Annuum* is displayed. Today, over 3000 herbaria have been created containing an estimated total of 350,000,000 specimens, ironically requiring an index of indexes (Thiers, 2018).

Another approach to preserve botanical information was issued by George Lincoln Goodale, the first director of Harvard's Botanical Museum. At the time, for scientifc study there were only either paper pressed herbaria which faded in color and had a collapsed three-dimensional structure, or coarse papier-mâché and wax models. However, a gap in realism remained between such models and their real world plant counterparts. After Goodale saw the photorealistic artisanal glass sculptures of the Balschka family, he requested Leopold and Rudolf Blaschka in 1886 to make botanical models. Today, the Ware Collection of Blaschka Glass Models of Plants, also known as *"the Glass Flowers"*, features over 4,000 models of more than 847 plant species. In Figure 1.3 a glass model of *Capsicuum Annuum* is shown.

More recently, plants have been modelled in a more abstract sense; e.g. functionally, structurally, or a combination of both (Vos et al., 2007a, 2010). Functional plant models describe the interaction of internal and external plant processes, whereas structural plant models focus only on the physical description of appearance. When both types of models are combined into functional-structural plant models, a higher-order description is formed of the development over time of the three-dimensional structure of plants, within the context of their physiological processes and as influenced by environmental factors (Vos et al., 2007b).

Our botanic modelling work in this thesis combines the structural modelling approach with the 3D virtue of glass models. Based on empirical plant parameter measurements, a methodology for photorealistic and structurally accurate three-dimensional models was created, albeit digital, that can be used as a tool for improving the learning of machines.



Figure 1.1: Botanical illustration of *Capsicuum Annuum*. From Medical Botany, page 144, William Woodville. London, 1793.



Figure 1.2: Herbarium of *Capsicuum Annuum*. From the collection of the University of Neuchâtel, reference number 093295. Printed with permission.



Figure 1.3: Blaschka glass model of Capsicum annuum (Model 448), 1894. Stereoscopic Slides A-J: The Archives of Rudolf and Leopold Blaschka and the Ware Collection of Blaschka Glass Models of Plants. Printed with permission of the Harvard University Herbaria.

1.5 Thesis Rationale

The objective of the work in this thesis was to further advance technology in agriculture, specifically by researching steps towards maturing agricultural robotics for harvesting in high-tech greenhouses. Currently, many agricultural robotics research prototypes exist, though they are not practically used due to poor harvest success rates and high cycle times. As discussed in the last section, the main bottleneck that was previously identified is caused by the inherent variation in both the crop and environment. Past sensing and manipulation control methods were not fully able to handle this variation (Gongal et al., 2015; Nasir et al., 2012). Explicitly,

from the hypotheses that:

'Agrobotics has not yet matured primarily because its sensing and control cannot yet handle all crop variation.'

&

'Computer vision is the most effective sensing solution to handle the crop variation.',

followed the main research question:

'Which novel and proven vision principles can improve agrobotics performance by coping with the crop variation and interaction?',

of which it was hypothesised that:

'State-of-the-art machine learning, notably deep learning based semantic segmentation, could improve sensing crop variation for agrobotics.'

&

'Visual servo control can help to sense more of the crop variation and provide feedback to make corrections for interaction with the crop.'

To further address the second hypothesis of computer vision being an important solution to handle the crop variation, it is important to sketch the still most common operational pipeline of a basic robot. Historically, greenhouse robotics applications followed the hierarchical sense-plan-act control paradigm (see Figure 1.4). In such an open-loop paradigm, a robot cycles through a sequence of using its sensors to build a world model, whereafter it plans its next move given some goals, after which its carries out the next act (Nilsson, 1984; Brooks, 1986). In time, the *'sense-plan-act'* and similar paradigms are likely to be replaced by methods that are more based on constraint optimisation, which links both continuous, discrete and symbolic knowledge at runtime continuously (Schutter et al., 2007).

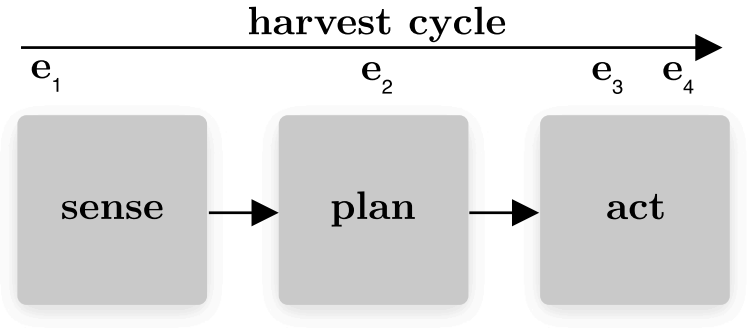


Figure 1.4: Schematic of an open-loop, hierachical sense-plan-act control paradigm primarily used in agricultural robotics. A single harvest cycle timeline is shown, consisting of the components *'sense'*, *'plan'* and *'act'* that are serially executed. During the harvest, the errors e_1, e_2, e_3 and e_4 can occur. See the accompanying text for further specification of these errors.

The problem with such a serial paradigm, amongst others, is twofold. First, the performance of each component is highly dependent on the performance of its prior. Early errors tend to have a larger effect on the final performance than errors in later components, because errors will accumulate and multiply with other inaccuracies along the chain of actions. For example, a 1 cm error in a noisy depth image (e_1 in Figure 1.4) is likely to cause a larger positioning offset at the end of a harvest cycle than a 1 cm overshoot error during motion execution (e_4).

Second, such a paradigm does not allow reactive behaviour within one task cycle when the robot acts, although that might be required when the scene changes. For example in agrobotics, the manipulator and crop interaction may result in dislocating the target

after sensing (e_2 in Figure 1.4) or during the motion towards the target (e_3. At that point in time, it is required to sense the scene change and re-plan the action.

To overcome the posed problems, two main topics in this work were investigated. Regarding error e_2, one would like to change the paradigm to allow more parallel active perception and replanning to steer the current action. Several paradigms exist to achieve this, one of which is to introduce a closed-loop pipeline that uses feedback during the robot's actions to re-plan and re-adjust if necessary. One popular method is visual servo control through eye-in-hand sensing. Although this approach is not novel for robotics, nor specifically for agriculture (Mehta & Burks, 2014; De-An et al., 2011; Hayashi et al., 2010b), most previous methods were tailored to a specific crop task and could not be generalised easily to new agricultural use-cases. Therefore, part of this work (Chapter 2) aimed to develop a modular and flexible framework for eye-in-hand sensing and visual control that could be applied to any use-case.

Regarding errors e_1 and e_3, it is essential that the sensing is as accurate as possible at the earliest stage possible to safeguard the final performance later on. Therefore, the second topic researched (Chapter 4) was the early component of sensing and how the robot segments or sub-divides the visual scene into meaningful regions. Previous manually crafted computer vision algorithms were unable to cope well with the crop and environment variation, as finding optimal image and object features that encompass all possibilities that could occur proved a hard task. Current state-of-the-art methods use machine learning, specifically convolutional neural networks, to be able to identify where objects or parts thereof are in an image (LeCun et al., 2015). Therefore, in this work it was investigated whether and how state-of-the-art machine learning, specifically deep learning, could be applied successfully to the agricultural domain.

Because such deep learning approaches need large annotated training datasets that are hard to manually obtain, methods for plant modelling and data synthesis were also an important research topic in this work (Chapters 3 and 5). Without proper and large datasets, machine learning cannot prevail. By modelling plant and agricultural scenes, higher level domain knowledge can be injected into the learning process, potentially increasing the recognition performance.

1.6 Outline of this Thesis

In Chapter 2, an eye-in-hand sensing and visual control framework was investigated. The goal of this work was to provide methods to overcome issues of occlusion and image registration that were previously introduced when sensing was performed externally from the robot manipulator (Bac et al., 2017; Henten et al., 2003). The hypothesis is that having local sensing information should resolve occlusion and image registration issues, e.g. by using cameras within the robot's end-effector itself where manipulation with the crop takes place. Moreover, this should also improve the performance in harvest time per fruit because most actions can be performed within adjacent hardware and during motion, thereby reducing redundant actions. Briefly, this Chapter also explored the possibility of adding simultaneous localisation and mapping (SLAM) (Engel et al., 2014) to obtain a three-dimensional world model using a monocular color camera.

Chapter 3 zoomed in on the sensing component of the framework postulated in the previous chapter. It was identified that in order to bring a new level of artificial intelligence to sensing for agricultural robotics, large amounts of annotated training data would be required to satisfy the learning needs of convolutional neural networks (CNNs, also known Deep Learning (LeCun et al., 2015)). However, manual annotation can quickly become a bottleneck, as time per image for our use-case averaged 30 minutes and potentially hundreds of images would be required. To overcome this issue, this Chapter presents a methodology to synthetically generate large sets of automatically annotated images, specifically for agricultural scenes. Based on a set of empirically measured plant parameters, a plant generation model was used to generate random instances of 3D meshes. Using render software, realistic scenes were generated that corresponded to the real world greenhouse architecture. It is hypothesised that the similarity between empirical photographs and the synthetic sets was both quantitatively and qualitatively high. An annotated dataset of 10,500 synthetic and 50 empirically photographed images was created and publicly released (Barth et al., 2018b). Furthermore, it is hypothesised and briefly investigated that synthetic images can be used to bootstrap semantic segmentation CNNs that fine-tune on a small set of empirical images, thereby also improving performance over other learning strategies.

The dataset was then used in Chapter 4 to explore a method to segment individual plant parts on a per-pixel level. Specifically, the main objective was to find an approach that minimised the requirement of annotated empirical images. It was further built on the hypothesis from Chapter 3 that only a small manually annotated empirical dataset would

be sufficient for fine-tuning a convolutional neural network that was bootstrapped with synthetic images. Furthermore, the following aspects were investigated: i) multiple deep learning architectures, ii) the correlation between synthetic and empirical dataset size on part segmentation performance, iii) the effect of post-processing using conditional random fields (CRF) and iv) the generalisation performance on other related datasets. For this seven main experiments were performed using different combinations of models, settings and combinations of learning data.

Although the synthetic images were modelled in the work of Chapter 3 to have an high similarity with the empirical situation, a realism gap still remained. Possibly, by using these synthetic images for bootstrapping, the plant part classification generalisation performance to empirical images was held back. To investigate this and to improve performance further on empirical images. In Chapter 5 a cycle consistent generative adversarial network (cGAN) was applied to our dataset with the objective to generate more realistic synthetic images by translating them to the feature distribution of the empirical domain. It is hypothesised that plant part image features such as color and texture become more similar to the empirical domain post translation. Furthermore, seven experiments were performed using convolutional neural networks with different combinations of synthetic, synthetic translated to empirical and empirical images. It is hypothesised that the translated images can be used for improved empirical learning and that without any empirical fine-tuning, improved empirical learning can be achieved when trained with translated images compared to only using synthetic images.

The aim of Chapter 6 was to bring all previous chapters into practice by estimating angles between fruit and stems to support visual servo control grasping in a sweet-pepper harvesting robot. It is hypothesised that from color images, such angles in the horizontal plane can be derived under unmodified greenhouse conditions whilst meeting the end-effector positioning requirements. For this it was further hypothesised that the location of a fruit and stem could be inferred from the image using sparse semantic segmentations. The work was separated into four sub-tasks and experiments were performed on 45 images per condition under simplified laboratory conditions, simplified greenhouse conditions and unmodified greenhouse conditions.

The thesis is concluded by a discussion and reflection, with a summary thereafter.

References

Alpha Brown (2018). Agricultural robotic harvesting solutions. url: `https://www.alphabrown.com/blank-2/robotic-harvesting-u-s-market-study`.

Aravind, K., Raja, P., & Pérez-Ruiz, M. (2017). Task-based agricultural mobile robots in arable farming: A review. *Spanish Journal of Agricultural Research*, *15*, 02–01. doi: 10.5424/sjar/2017151-9573.

Arima, S., & Kondo, N. (1999). Cucumber harvesting robot and plant training system. *Journal of Robotics and Mechatronics*, *11*, 208–212.

Arndt, G. (1985). Technology transfer and agricultural robotics. *CIRP Annals*, *34*, 381 – 386. doi: https://doi.org/10.1016/S0007-8506(07)61794-6.

Bac, C. W. (2015). *Improving obstacle awareness for robotic harvesting of sweet-pepper*. Ph.D. thesis Wageningen. WU thesis 5954.

Bac, C. W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, . doi: 10.1002/rob.21709.

Bac, C. W., Van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Barth, R., , Hemming, J., & van Henten, E. (2018a). Estimating angles between fruit and stems to support grasping in a sweet-pepper harvesting robot.. . Submitted to the Journal of Computers and Electronics in Agriculture.

Barth, R., Hemming, J., & van Henten, E. J. (2016). Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, *146*, 71 – 84. doi: https://doi.org/10.1016/j.biosystemseng.2015.12.001. Special Issue: Advances in Robotic Agriculture for Crops.

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. V. (2017). Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture*, . doi: https://doi.org/10.1016/j.compag.2017.11.040.

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. V. (2018b). Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Computers and Electronics in Agriculture*, *144*, 284 – 296. doi: https://doi.org/10.1016/j.compag.2017.12.001.

Baylou, P. (1987). Agricultural robots. *IFAC Proceedings Volumes*, *20*, 111 – 119. doi: https://doi.org/10.1016/S1474-6670(17)55251-9. 10th Triennial IFAC Congress on Automatic Control - 1987 Volume V, Munich, Germany, 27-31 July.

Beerling, E., van Os, E., van Ruijven, J., Janse, J., Lee, A., & Blok, C. (2017). Water-efficient zero-emission greenhouse crop production: a preliminary study. In *ActaHortic.* 1170 (pp. 1133–1140). International Society for Horticultural Science (ISHS), Leuven, Belgium.

Blackmore, S., Stout, B., Wang, M., & Runov, B. (2005). Robotic agriculture – the future of agricultural mechanisation? In *Proceedings of the 5th European Conference on Precision Agriculture*.

Blue River Technology (2018). Blue river technology. url: `http://www.bluerivertechnology.com/`.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Image classification using random forests and ferns. In *2007 IEEE 11th International Conference on Computer Vision* (pp. 1–8). doi: 10.1109/ICCV.2007.4409066.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, *2*, 14–23. doi: 10.1109/JRA.1986.1087032.

Christou, P., McCabe, D. E., Martinell, B. J., & Swain, W. F. (1990). Soybean genetic engineering - commercial production of transgenic plants. *Trends in Biotechnology*, *8*, 145 – 151. doi: https://doi.org/10.1016/0167-7799(90)90160-Y.

Davis, K. (1973). Zero population growth: The goal and the means, . *102*, 15–30.

Davis, K. F., Gephart, J. A., Emery, K. A., Leach, A. M., Galloway, J. N., & D'Odorico, P. (2016). Meeting future food demand with current agricultural resources. *Global Environmental Change*, *39*, 125 – 132. doi: https://doi.org/10.1016/j.gloenvcha.2016.05.004.

De-An, Z., Jidong, L., Wei, J., Ying, Z., & Yu, C. (2011). Design and control of an apple harvesting robot. *Biosystems Engineering*, *110*, 112 – 122. doi: https://doi.org/10.1016/j.biosystemseng.2011.07.005.

Delbrück, M., & Stent, G. S. (1988). Mind from matter? an essay on evolutionary epistemology. *Journal of Genetics*, *67*, 173–178. doi: 10.1007/BF02927828.

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision*

ECCV 2014 (pp. 834–849). Springer International Publishing volume 8690 of *Lecture Notes in Computer Science*. doi: 10.1007/978-3-319-10605-2_54.

Espenshade, T. J., Guzman, J. C., & Westoff, C. F. (2003). The surprising global variation in replacement fertility. *Population Research and Policy Review*, *22*, 575–583. doi: 10.1023/B:POPU.0000020882.29684.8e.

Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F., Pretty, J., Robinson, S., Thomas, S. M., & Toulmin, C. (2010). Food security: The challenge of feeding 9 billion people. *Science*, *327*, 812–818.

Gold, M. (2004). *The global benefits of eating less meat: a report for Compassion in World Farming Trust*. New Delhi: Navodanya in collaboration with Compassion in World Farming Trust.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, *116*, 8 – 19.

von Grebmer, K., Saltzman, A., Birol, E., Wiesman, D., Prasai, N., Yin, S., Yohannes, Y., Menon, P., Thompson, J., & Sonntag, A. (2014). *2014 Global Hunger Index: The challenge of hidden hunger*. Number 978-0-89629-958-0 in IFPRI books. International Food Policy Research Institute (IFPRI).

Gruda, N. (2005). Impact of environmental factors on product quality of greenhouse vegetables for fresh consumption. *Critical Reviews in Plant Sciences*, *24*, 227–247. doi: 10.1080/07352680591008628.

Gustavsson, J., Cederberg, C., Sonesson, U., van Otterdijk, R., & Meybeck, A. (2011). Global food losses and food waste. *Food and Agriculture Organization of the United Nations*, .

Gwan, J. et al. (2017). Deep convolutional neural network for classifying fusarium wilt of radish from unmanned aerial vehicles, . *11*.

Hamuda, E., Glavin, M., & Jones, E. (2016). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, *125*, 184 – 199. doi: https://doi.org/10.1016/j.compag.2016.04.024.

Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., & Kurita, M. (2010a). Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, *105*, 160 – 171. doi: http://dx.doi.org/10.1016/j.biosystemseng.2009.09.011.

Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., & Kurita, M. (2010b). Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, *105*, 160 – 171. doi: https://doi.org/10.1016/j.biosystemseng.2009.09.011.

van Henten, E. J. (2006). Greenhouse mechanization: State of the art and future perspective. In *ActaHortic.* 710 (pp. 55–70). International Society for Horticultural Science (ISHS), Leuven, Belgium.

Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., Bontsema, J., & Os, E. V. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering*, *86*, 305 – 313. doi: http://dx.doi.org/10.1016/j.biosystemseng.2003.08.002.

Hu, H., & Xiong, L. (2014). Genetic engineering and breeding of drought-resistant crops. *Annual Review of Plant Biology*, *65*, 715–741. doi: 10.1146/annurev-arplant-050213-040000.

Isaacs, G. W. (1986). Robotic applications in agriculture. In *ActaHortic.* 187 (pp. 123–128). International Society for Horticultural Science (ISHS), Leuven, Belgium.

Jukema, G., & Van de Meer, R. (2009). Labor costs in arable farming and greenhouse horticulture. *The Netherlands: Landbouw Economisch Instituur (LEI)*, .

Kapach, K., Barnea, E., Mairon, R., Edan, Y., & Shahar, O. (2012). Computer vision for fruit harvesting robots—state of the art and challenges ahead, . *3*, 4–34.

Kaplan, D. (2000). The darker side of the "original affluent society". *Journal of Anthropological Research*, *56*, 301–324. doi: 10.1086/jar.56.3.3631086.

Katsoulas, N., Sapounas, A., Zwart, F. D., Dieleman, J., & Stanghellini, C. (2015). Reducing ventilation requirements in semi-closed greenhouses increases water use efficiency. *Agricultural Water Management*, *156*, 90 – 99. doi: https://doi.org/10.1016/j.agwat.2015.04.003.

Kondo, N., Yata, K., Iida, M., Shiigi, T., Monta, M., Kurita, M., & Omori, H. (2010). Development of an end-effector for a tomato cluster harvesting robot. *Engineering in Agriculture, Environment and Food*, *3*, 20 – 24. doi: https://doi.org/10.1016/S1881-8366(10)80007-2.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc.

Krugman, P. (1979). A model of innovation, technology transfer, and the world distribution of income. *Journal of Political Economy*, *87*, 253–266. doi: 10.1086/260755.

Krupenye, C., Kano, F., Hirata, S., Call, J., & Tomasello, M. (2016). Great apes anticipate that other individuals will act according to false beliefs. *Science*, *354*, 110–114. doi: 10.1126/science.aaf8110.

Lal, B. B. (2003). *Excavations at Kalibangan, the early Harappans, 1960-1969*. Archaeological Survey of India.

Larson, G. et al. (2014). Current perspectives and the future of domestication studies. *Proc Natl Acad Sci U S A*, *111*, 6139–6146. doi: 10.1073/pnas.1323964111. 201323964[PII].

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning, . *521*, 436–44.

Lee, R., & Daly, R. (1999). *The Cambridge Encyclopedia of Hunters and Gatherers*. Cambridge University Press.

Lehnert, C., English, A., McCool, C., Tow, A. W., & Perez, T. (2017). Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, *2*, 872–879. doi: 10.1109/LRA.2017.2655622.

Lin, K.-H., Huang, M.-Y., Huang, W.-D., Hsu, M.-H., Yang, Z.-W., & Yang, C.-M. (2013). The effects of red, blue, and white light-emitting diodes on the growth, development, and edible quality of hydroponically grown lettuce (lactuca sativa l. var. capitata). *Scientia Horticulturae*, *150*, 86 – 91. doi: https://doi.org/10.1016/j.scienta.2012.10.002.

Mehta, S., & Burks, T. (2014). Vision-based control of robotic manipulator for citrus harvesting. *Computers and Electronics in Agriculture*, *102*, 146 – 158. doi: https://doi.org/10.1016/j.compag.2014.01.003.

Meyer, G. E., Neto, J. C., Jones, D. D., & Hindman, T. W. (2004). Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images. In *Computers and Electronics in Agriculture* (pp. 161–180).

Nasir, A., Rahman, M., & Mamat, A. (2012). A study of image processing in agriculture application under high performance computing environment. *International Journal of Computer Science and Telecommunications*, *3*.

Nilsson, N. J. (1984). *Shakey the robot*. Technical Report, SRI INTERNATIONAL MENLO PARK CA.

Olle, M., & Viršile, A. (2013). The effects of light-emitting diode lighting on greenhouse plant growth and quality. *Agricultural and Food Science*, *22*, 223–234.

Onwude, D. I., Abdulstter, R., Gomes, C., & Hashim, N. (2016). Mechanisation of large-scale agricultural fields in developing countries – a review. *Journal of the Science of Food and Agriculture*, *96*, 3969–3976. doi: 10.1002/jsfa.7699.

Opdam, J. J. G., Schoonderbeek, G. G., Heller, E. M. B., & de Gelder, A. (2005). Closed greenhouse: A starting point for sustainable entrepreneurship in horticulture. In *ActaHortic.* 691 (pp. 517–524). International Society for Horticultural Science (ISHS), Leuven, Belgium.

Pearlstein, L., Kim, M., & Seto, W. (2016). Convolutional neural network application to plant detection, based on synthetic imagery. In *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1–4). doi: 10.1109/AIPR.2016.8010596.

Pedanius, D., & Beck, L. (2005). *De Materia Medica*. Altertumswissenschaftliche Texte und Studien. Olms-Weidmann.

Pekkeriet, E. J. (2011). Crops project deliverable 12.1: Economic viability for each application.

Pekkeriet, E. J., & van Henten, E. J. (2011). Current developments of high-tech robotic and mechatronic systems in horticulture and challenges for the future. In *ActaHortic.* 893 (pp. 85–94). International Society for Horticultural Science (ISHS), Leuven, Belgium.

Post, M. J. (2012). Cultured meat from stem cells: Challenges and prospects. *Meat Science*, *92*, 297 – 301. doi: https://doi.org/10.1016/j.meatsci.2012.04.008. 58th International Congress of Meat Science and Technology (58th ICoMST).

Ray, D. K., Mueller, N. D., West, P. C., & Foley, J. A. (2013). Yield trends are insufficient to double global crop production by 2050. *PLOS ONE*, *8*, 1–8. doi: 10.1371/journal.pone.0066428.

Reid, J., & Searcy, S. (1987). Vision-based guidance of an agriculture tractor. *IEEE Control Systems Magazine*, *7*, 39–43. doi: 10.1109/MCS.1987.1105271.

Rogers, J. E. T. (1949). Six centuries of work and wages. (pp. 542–43). London: Allen and Unwin.

Rosling, H., Rosling, O., & Rönnlund, A. (2018). *Factfulness*. Hodder & Stoughton.

Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., & Perez, T. (2017). Peduncle detection of sweet pepper for autonomous crop harvesting combined color and 3-d information. *IEEE Robotics and Automation Letters*, *2*, 765–772. doi: 10.1109/LRA.2017.2651952.

Sahlins, M. (2009). Hunter-gatherers: Insights from a golden affluent age. *Pacific Ecologist*, . Http://www.pacificecologist.org/archive/18/pe18-hunter-gatherers.pdf.

Sarig, Y. (1993). Robotics of fruit harvesting: A state-of-the-art review. *Journal of Agricultural Engineering Research*, *54*, 265 – 280. doi: https://doi.org/10.1006/jaer.1993.1020.

Sarkar, N., & Wolfe, R. (1985). Computer vision based system for quality separation of fresh market tomatoes. *Transactions of the ASAE*, *28*, 1714.

Schutter, J. D., Laet, T. D., Rutgeerts, J., Decré, W., Smits, R., Aertbeliën, E., Claes, K., & Bruyninckx, H. (2007). Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, *26*, 433–455. doi: 10.1177/0278364907078091O7. arXiv:https://doi.org/10.1177/0278364907078091O7.

Shepon, A., Eshel, G., Noor, E., & Milo, R. (2016). Energy and protein feed-to-food conversion efficiencies in the us and potential food security gains from dietary changes. *Environmental Research Letters*, *11*, 105002.

Sistler, F. (1987). Robotics and intelligent machines in agriculture. *IEEE Journal on Robotics and Automation*, *3*, 3–6. doi: 10.1109/JRA.1987.1087074.

Sites, P., & Delwiche, M. (1988). Computer vision to locate fruit on a tree. *Transactions of the ASAE*, *31*, 257.

Stehfest, E., Bouwman, L., van Vuuren, D. P., den Elzen, M. G. J., Eickhout, B., & Kabat, P. (2009). Climate benefits of changing diet. *Climatic Change*, *95*, 83–102. doi: 10.1007/s10584-008-9534-6.

Stringer, C., & Gamble, C. (1993). *In Search of the Neanderthals: Solving the Puzzle of Human Origins*. Thames and Hudson.

Thiers, B. (2018). Index herbariorum: A global directory of public herbaria and associated staff. url: `http://sweetgum.nybg.org/science/ih/`.

Tillett, R. (1990). Vision-guided planting of dissected microplants. *Journal of Agricultural Engineering Research*, *46*, 197 – 205. doi: https://doi.org/10.1016/S0021-8634(05)80126-X.

Tomonaga, M., Uwano, Y., Ogura, S., & Saito, T. (2010). Bottlenose dolphins' (tursiops truncatus) theory of mind as demonstrated by responses to their trainers' attentional states. *Int. J. Comp. Psychol*, *23*, 386–400.

Turner, M. (2006). Feeding the world: an economic history of agriculture, 1800 - 2000. *The Economic History Review*, *59*, 661–663.

Tyler-Whittle, M. (1970). *The Plant Hunters: Being an Examination of Collecting with an Account of the Careers and the Methods of a Number of Those who Have Searched the World for Wild Plants*. Horticulture Garden Classics Series. Lyons & Burford.

Van Henten, E., , Hemming, J., Van Tuijl, B., Kornet, J., Meuleman, J., Bontsema, J., & Van Os, E. (2002). An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, *13*, 241–258. doi: 10.1023/A:1020568125418.

Vos, J., Evers, J. B., Buck-Sorlin, G. H., Andrieu, B., Chelle, M., & de Visser, P. H. B. (2010). Functional–structural plant modelling: a new versatile tool in crop science. *Journal of Experimental Botany*, *61*, 2101–2115. doi: 10.1093/jxb/erp345.

Vos, J., Marcelis, L. F. M., Visser, P., Struik, P. C., & Evers, J. (2007a). *Functional-Structural Plant Modelling in Crop Production*. Springer Netherlands.

Vos, J., Marcelis, L. F. M., de Visser, P. H. B., Struik, P. C., & Evers, J. B. (2007b). *Functional-Structural Plant Modelling in Crop Production*. (1st ed.). Springer Publishing Company, Incorporated.

Wageningen University and Research (2018a). Clever robots for crops : Crops. url: `http://www.crops-robots.eu/`.

Wageningen University and Research (2018b). Sweet pepper harvesting robot. url: `http://www.sweeper-robot.eu/`.

Walker, M. et al. (2009). Formal definition and dating of the gssp (global stratotype section and point) for the base of the holocene using the greenland ngrip ice core, and selected auxiliary records. *Journal of Quaternary Science*, *24*, 3–17. doi: 10.1002/jqs.1227.

Wilson, A. (2002). Machines, power and the ancient economy. *The Journal of Roman Studies*, *92*, 1–32.

Wolfe, R., & Swaminathan, M. (1987). Determining orientation and shape of bell peppers by machine vision. *Transactions of the ASAE*, *30*, 1853.

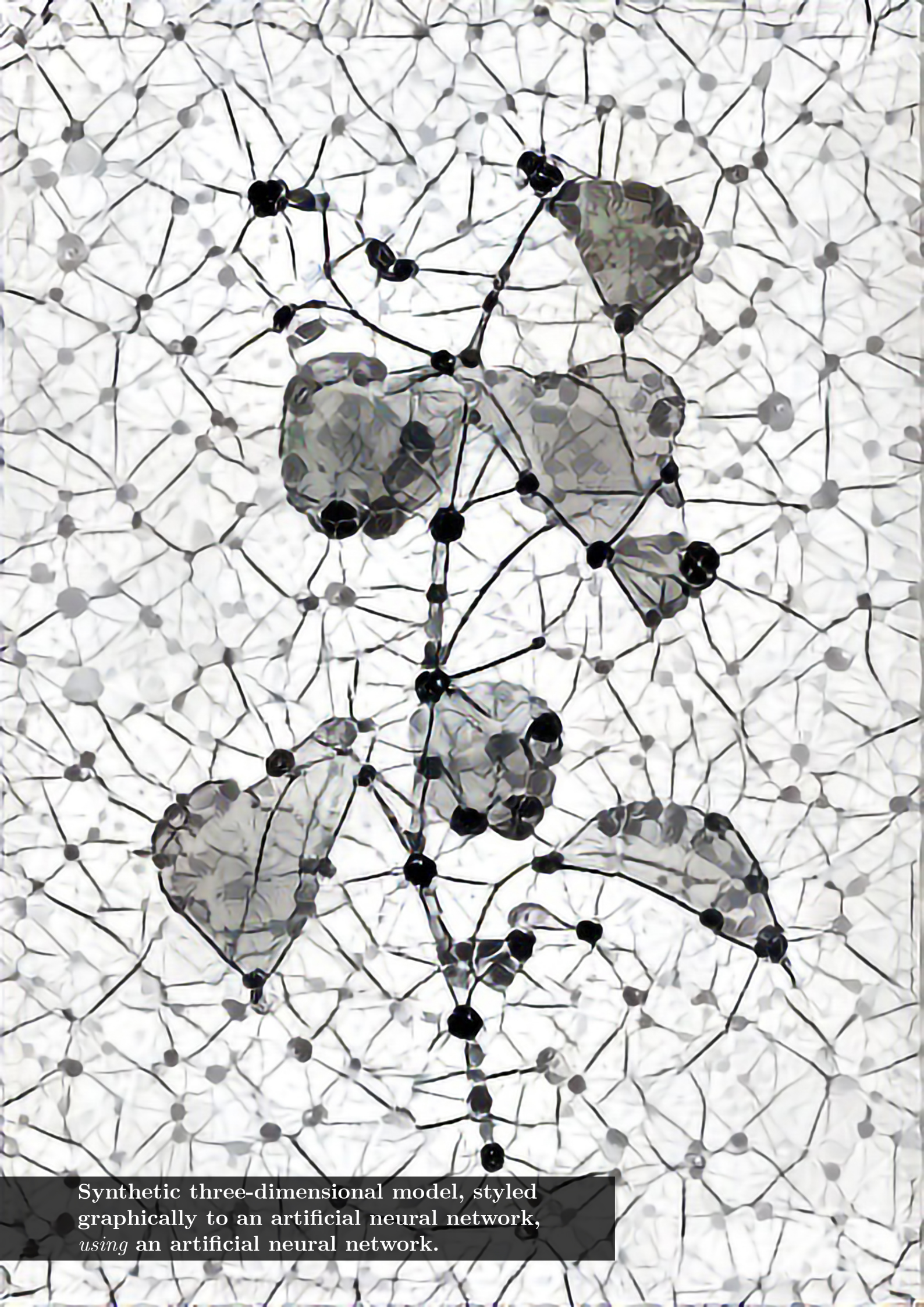
Yalcin, H., & Razavi, S. (2016). Plant classification using convolutional neural networks. In *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* (pp. 1–5). doi: 10.1109/Agro-Geoinformatics.2016.7577698.

Zakiuddin, K. S., Sondawale, H. V., Modak, J. P., & Ceccarelli, M. (2012). History of human powered threshing machines: A literature review. In T. Koetsier, & M. Cecca-

relli (Eds.), *Explorations in the History of Machines and Mechanisms* (pp. 431–445). Dordrecht: Springer Netherlands.

Zeisel, J. (1958). The workweek in american industry, 1850-1956. In *Monthly Labor Review* (pp. 23–29).

Zheng, L., Zhang, J., & Wang, Q. (2009). Mean-shift-based color segmentation of images containing green vegetation. *Computers and Electronics in Agriculture*, *65*, 93 – 98. doi: https://doi.org/10.1016/j.compag.2008.08.002.



Synthetic three-dimensional model, styled graphically to an artificial neural network, *using* an artificial neural network.

Chapter 2

Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation.

This chapter is based on:

Barth, R., Hemming, J., and Henten, E.J. van (2016). Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, 146, 71 – 84. Special Issue: Advances in Robotic Agriculture for Crops.

Awarded with the *'Outstanding Paper Award 2018'* from the *'European Society of Agricultural Engineers'*.

Abstract

We present a modular software framework design that allows flexible implementation of eye-in-hand sensing and motion control for agricultural robotics in dense vegetation. Harvesting robots in cultivars with dense vegetation require multiple viewpoints and online trajectory adjustments in order to reduce the amount of false negatives and correct for fruit movement. In contrast to specialised software, the framework proposed aims to support a wide variety of agricultural use cases, hardware and extensions. A set of Robotic Operating System (ROS) nodes was created to ensure modularity and separation of concerns, implementing functionalities for application control, robot motion control, image acquisition, fruit detection, visual servo control and simultaneous localisation and mapping (SLAM) for monocular relative depth estimation and scene reconstruction. Coordination functionality was implemented by the application control node with a finite state machine. In order to provide visual servo control and simultaneous localisation and mapping functionalities, off-the-shelf libraries ViSP and LSD-SLAM were wrapped in ROS nodes. The capabilities of the framework are demonstrated by an example implementation for use with a sweet-pepper crop, combined with hardware consisting of a Baxter robot and a color camera placed on its end-effector. Qualitative tests were performed under laboratory conditions using an artificial dense vegetation sweet-pepper crop. Results indicated the framework can be implemented for sensing and robot motion control in sweet-pepper using visual information from the end-effector. Future research to apply the framework to other use-cases and validate the performance of its components in servo applications under real greenhouse conditions is suggested.

2.1 Introduction

During the design of robotic applications for harvesting horticultural products, two key challenges need to be solved. The first is the detection of a target location of the fruit. The second is moving the end-effector towards that location with precision to perform a harvest action. There are several ways to address each of these challenges. For example, one approach solves fruit detection by using one or few viewpoints from a sensing module located externally from the robot. However, in crops with a high vegetation density, using a low number of viewpoints results in false negatives due to a high amount of occluding leaves and branches (Hemming et al., 2014b). Furthermore, the external placement of the sensor(s) require one or multiple frame transformations. Slight errors therein accumulate, resulting in inaccurate target coordinates. When a location of the target fruit is acquired, moving the end-effector there can be solved by executing a planned motion trajectory without additional sensing. However, dislocation of the target can occur as the robot enters and interacts with a dense crop. Both the frame transformation errors and dislocation of the target can result in poor end-effector placement at the target (Hemming et al., 2014a; Henten et al., 2003). An example implementation of external sensing and planned motion control was tested during the European 7th Framework Programme project Clever Robots for Crops (CROPS) (GA no. 246252). During this project, a proof-of-principle harvesting robot was created for a dense sweet-pepper crop. A sensing module dislocated from the robot provided fruit detection from a single viewpoint. Thereafter a motion trajectory was executed without further sensing. It was concluded that these approaches were one of the causes of low harvest performance, both in cycle time as in fruit detection rates (Bac, 2015). Another example of a cucumber harvesting robot uses a similar approach (Van Henten, E. J. et al., 2002), where a single viewpoint in the workspace of the robot provided fruit positions. In both the CROPS and cucumber robot, additional sensing could be performed to refine fruit positions with a second set of cameras on the end-effector. However, in both field tests this feature was not used. This extra single sensing step before the final motion execution is also known as look-and-move (Hutchinson et al., 1996). When a camera is attached to the end-effector, it is often named an eye-in-hand sensor (Hutchinson et al., 1996). For a strawberry harvesting robot, a similar eye-in-hand look-and-move approach was used (Hayashi et al., 2010).

A different approach is to solve fruit detection and motion control using primarily eye-in-hand sensing. External sensors are not necessarily excluded in this paradigm,

though the application is not dependent on this additional secondary sensing source. For the fruit detection, the internal location of the sensor(s) reduces the number of coordinate frame transformations to a single one. Moreover it allows the application to sense the scene from multiple viewpoints with pose changes of the end-effector, expected to decrease the number of false negative detections in a dense crop. For the motion towards the target, this approach allows for continuous incremental visual feedback and corrections, also known as visual servo control (Hutchinson et al., 1996; Marchand et al., 2005). Examples of robotic harvesters in horticulture using visual servo control are numerous. For a sweet-pepper harvesting robot in Japan, a visual servo control algorithm positioned the end-effector near the fruit using stereo images (Kitamura & Oka, 2005). Although the camera was not part of the end-effector, it was placed within its workspace and aligned with the optical axis. In a strawberry harvesting robot, a set of external sensors first provided rough fruit position after which an eye-in-hand system moved towards it using visual servo control (Han et al., 2012). For applications of an apple harvesting robot (De-An et al., 2011) and a citrus harvesting robot (Mehta & Burks, 2014), eye-in-hand visual servo control systems were created. Another application for an apple harvesting robot also applied eye-in-hand sensing, however did not implement a full visual servo control. Instead look-and-move corrections were performed multiple times during the fruit approach (Baeten et al., 2008).

The aim of our research was to provide a flexible modular framework for eye-in-hand sensing and motion control in robotic harvest applications as a standardised approach. In the aforementioned previous research, the designs of sensing and visual servo control algorithms were ad hoc and application specific, therefore hard to migrate to other use-cases. Our approach aims to provide a consistent approach, designed to cope with a wide variety of applications. The framework is primarily designed for dense crops in agri- and horticultural robotics, where a single viewpoint is not sufficient for sensing. Other frameworks for sensing and visual servo control focus on the design of a single low level function (Bachiller et al., 2003; Mahony, 2011; Jara et al., 2014; Marchand et al., 2005). Our aim is to provide a higher level framework architecture that spans the functionality required for a full robot application, as suggested in previous research (Bachiller et al., 2006).

This chapter firstly provides the general design of the framework in section 2.2.1 by describing the required functionalities and architecture of the software and its components. An example implementation for a sweet-pepper use case is then described

in section 4.3, along with qualitative tests under laboratory conditions. The primary aim of these tests was to i) demonstrate the framework can execute an eye-in-hand sensing and visual servo control sequence and ii) extending the functionality of sensing with 3D scene reconstruction. The performance of eye-in-hand sensing and motion control libraries are not validated. Section 5.6 describes the results of our research followed by a discussion in section 3.4.

2.2 Materials

2.2.1 Software

The functions of a robot can be divided into three broad primitives: sensing, planning and acting (Murphy, 2000). To organise the robotic behaviour with these primitives, one of several paradigms can be implemented in a software architecture. For a visual servo control task, a reactive paradigm is most applicable because it routes sensor information directly to actions. However, this omits any planning that an application may need. The hybrid deliberative/reactive paradigm introduces the planning primitive whilst also supporting reactive behaviour. In this paradigm, a global planner executes sub-task that can be either planned or reactive, acting as an intermediate coordinator of sensing information (Murphy, 2000). For our framework this paradigm is chosen because both planning and reactive tasks were used.

A software architecture, or framework, that implements the hybrid deliberative/reactive paradigm should describe a set of components and their interaction (Dean & Wellman, 1991). For our framework five required functionalities were differentiated that fall into the three primitives of sensing, planning and/or acting: (i) image acquisition (ii) fruit detection (iii) application control, (iv) visual servo control and (v) robot control. The functions (i), (iii) and (v) fall into a single primitive. However, functions (ii) and (iv) overlap in the planning primitive because they also process, analyse and plan with data.

In Fig. 2.1 an overview of the functions for the framework is provided, divided over the robotic primitives. Flexibility of the framework results from the functional implementation in independent modules. Through such a design pattern, functionality is replaceable and expandable with new features without revisions of other modules. This in contrast to creating a single library that entangles all functionality, resulting in poor affordance to

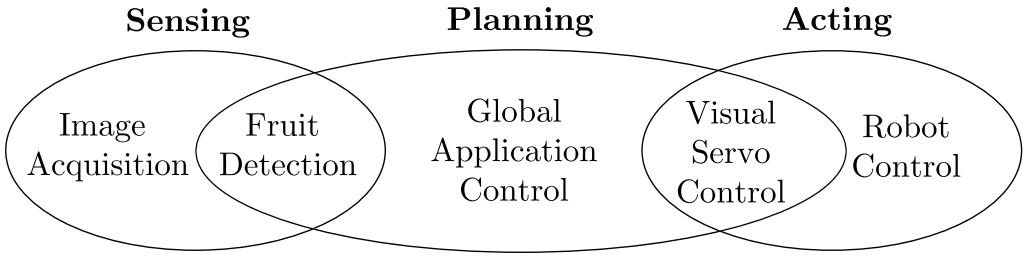


Figure 2.1: Venn diagram of the framework's required functions, divided over the robot behaviour primitives sensing, planning and acting.

substitution of components and a limited separation of concerns (Felix & Ortin, 2014).

The functions were implemented in the middleware 'Robotic Operating System' (ROS) (Quigley et al., 2009). ROS allows the creation of a modular networks of nodes that perform dedicated subsets of the computation and organises the communication between them. Furthermore, a shared stack of robotic libraries are available to all nodes to facilitate computations for robotics, such as for timing, coordinate frame transformations and robot motion simulations. ROS also provides a set of basic communication policies such as services, publishers and subscriptions as well as a more advanced policy where actions can be monitored or preempted during a continuous feedback loop.

In Fig. 2.2 the suggested interaction architecture between the ROS nodes of the framework is displayed. Functionalities in the framework were explicitly separated to facilitate replacement of nodes and the extension of new functionalities. Furthermore, centralised functionalities avoid functional duplication across nodes. An additional function of simultaneous localisation and mapping (SLAM) was added to show the extendibility of the framework. The central position of the application control node in the communication allows for flexible coordination as opposed to distributed control over several interacting nodes. In the following sections, the required functionality of each node will described in detail.

Application Control

The application control node was to implement the functionality of coordination by communicating with all other nodes and processing their feedback information. This goal was achieved in this node by implementing a finite state machine (FSM), based on previous research experience (Hellstrom & Ringdahl, 2013; Barth et al., 2014), which has similarity with other FSM approaches like ROS Commander (Nguyen et al., 2013) and SMACH (Bohren & Cousins, 2010). The FSM is a modular collection of states and their respective

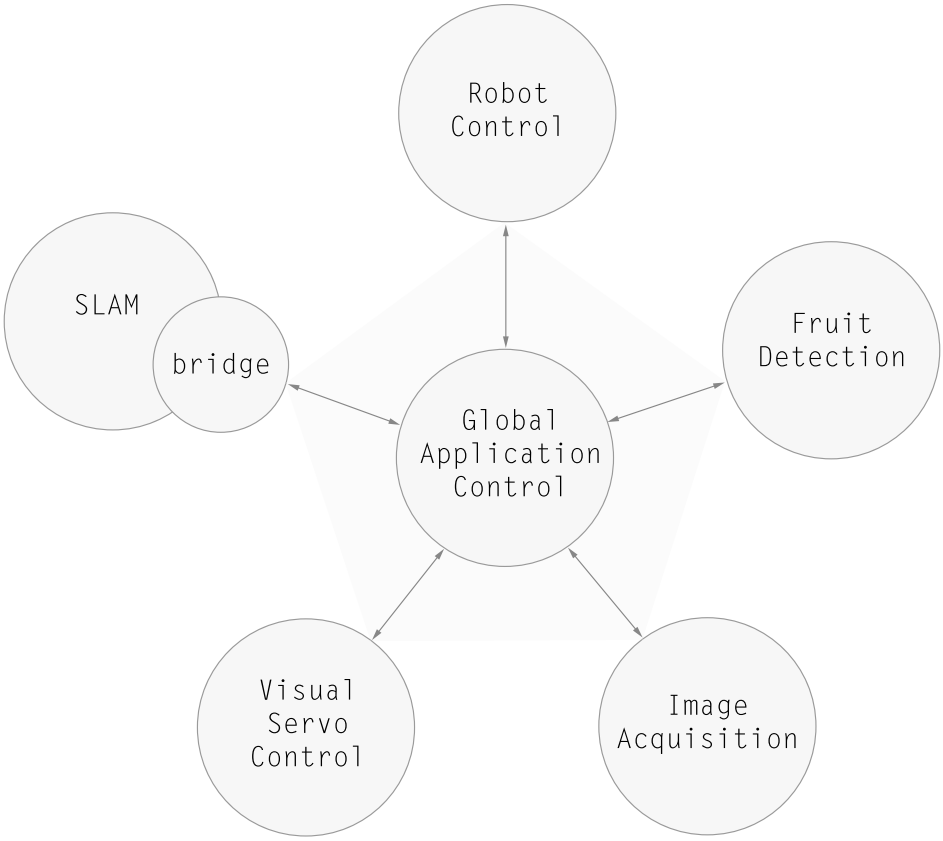


Figure 2.2: ROS nodes architecture of the eye-in-hand sensing and motion control framework. Links indicate communication interactions.

state transitions. In each state a certain subtask of the robot application can be executed. The use of a state machine gave the framework another layer of flexibility by facilitating the reuse of states, smooth addition of states and rerouting of transitions without requiring recompilation of the framework. The concern of coordination was separated in this node from the other nodes.

Image Acquisition

The image acquisition node provided the framework with the functionality of creating a connection to a camera and grabbing color and monochrome images upon request. ROS required the images to be sent in the ROS image format. The node requires exposure settings and gains for each channel, which can be set in the ROS launch file. To obtain the gain parameters, a color calibration procedure should be performed by manually adjusting the gain levels until the red, green and blue values of all pixels are equal given a recorded image of a grey calibration reference object. Rectification of the images before sending is required to allow a relation between image coordinates in pixels and real world coordinates in meters accordingly. For this, the camera parameters should be known.

Fruit Detection

The goal of the fruit detection node was to provide information about the fruit in a given image. Note that the function of fruit detection is a broad term, to which at least 3 sub-functions can be distinguished that are relevant for harvesting robots: finding fruits, localising fruits in 3D and determining ripeness and/or harvestability. Depending on which sub-functions are required by the application, each sub-function should provide a service that returns a set of features of an image. These features can be descriptive, like surface areas, or geometrical like the position of the largest fruit in the image. In this framework, the visual servo control constrains the image analysis computation time and should be below 100 ms.

Visual Servo Control

The functionality to be provided by this node was to use image features to control the motion of a robot, using a continuous correctional feedback loop (Hutchinson et al., 1996) or on-line trajectory generation(Kröger, 2010). Image information can be used from one of more cameras, either located on the gripper or external from the robot. Independent of the camera configuration, the task requires a set of geometrical visual features s to be extracted from the acquired image(s). In our framework the fruit detection node provides this functionality. To use the geometrical features for correcting the motion of the robot, a control law must be designed that realises the desired feature values s^* by minimising the error (s - s^*). For this an interaction matrix L_s , also known as the image Jacobian, needs to be approximated that models the relationship between the time variation of the features and the camera velocity v (Marchand et al., 2005). Vector v is also known as the kinematics screw vector, encoding the required variation in pose of the camera relative to the object. The general case of an eye-in-hand control law where camera velocities are computed is defined by:

$$v = -\lambda \widehat{L_s^+}(s - s^*) \,, \tag{2.1}$$

where λ is the proportional coefficient of the exponential convergence of the error and $\widehat{L_s^+}$ is the pseudo-inverse of the estimation of the interaction matrix, which is parameterised by intrinsic camera parameters (focal length, image sensor format, and principal image point) and feature location information (m), relative to the camera frame.

To add robot motion control to the framework, the Visual Servoing Platform library (ViSP) was wrapped in a ROS node (Marchand, 1999; Marchand et al., 2005), allowing for rapid prototyping of visual servo control algorithms and specifically designed/developed for high-level applications. ViSP features a set of elementary tasks in which visual features can be combined. It is designed to be modular, hardware-independent, extendable and portable, making this library highly suitable as a key component for our highly flexible and modular framework. Under the assumption that visual features are defined upon geometrical primitives, such as points or lines, ViSP can approximate the interaction matrix analytically using a previously proposed method (Espiau et al., 1992).

In each iteration of the servo control loop, this ROS node computes the velocity vector $\mathbf{v}$ given (i) a set of desired geometric features s^*, (ii) a set of current geometric features $\mathbf{s}$ and (iii) the convergence coefficient, ranging from 0 to 1. The velocity vector encodes the required change in pose applied to the end-effector to converge to the desired feature values. Note that in some cases convergence and stability problems may occur (Chaumette, 1998).

Robot Control

The robot control node should provide the functionality to move the end-point of a robot to a desired pose, receiving a real-world Cartesian coordinate and returning a movement status. Visual servo control requires updating the end-effector goal pose multiple times a second. This can, for example, can be achieved by goals that can be pre-empted or when joint motions of the robot are directly accessible. In section 4.3 the way this was achieved with the Baxter robot is described.

Simultaneous Localisation and Mapping

The aim of this node is to provide additional sensing information from images by implementing a simultaneous localisation and mapping (SLAM) method. Largely used for unknown and unmapped environments, this approach allows for camera pose estimations and three-dimensional scene reconstruction (Durrant-Whyte & Bailey, 2006). From this information, relative depth between objects in the scene can be derived.

For this purpose, the Large Scale Direct SLAM (LSD-SLAM) was used (Engel

et al., 2014). In contrast to other methods, it runs real-time on modern CPU's and uses a featureless approach working directly on monocular image intensities. The library was already wrapped in a ROS Indigo node, hence no modifications for our framework were required. However, by default only the visual odometry (estimated camera pose) in combination with relative depth map keyframes were published in the ROS system. Therefore the three-dimensional reconstruction was not available outside the LSD-SLAM node. To add this feature in ROS, a bridge node was implemented which concatenated multiple depth keyframes after transformation to the same world frame using the SLAM's published odometry information.

2.2.2 Hardware

The hardware used for testing the framework consisted of a camera attached to the end-effector of a robot. The camera was connected to the computer through USB and the robot was connected to the computer through an ethernet connection.

Robot

The framework was applied and tested on a Baxter robot of Rethink Robotics (Fitzgerald, 2013), depicted in Fig. 2.3. The robot was designed to mimic and replace workers on a production line, performing tasks such as sorting or picking and placing parts. The human sized robot consists of 2 mirrored 7 degrees of freedom arms, of which only one was used here. The robot was chosen for its out-of-the box ROS support. Baxter runs a ROS master core to which target joint angles can be published, executed by an internal controller. Baxter also provides inverse kinematics (IK) service for calculating joint angles given a 3 dimensional Cartesian coordinate relative to the robot frame. Furthermore, Baxter publishes the pose of the end-effector and all joints. The standard Baxter end-effector was used.

Camera

A USB CMOS color Autofocus Camera (DFK 72AUC02-F, TheImagingSource, Germany) was attached on top of the tool centre point of the robot, to which also the standard end-effector was also mounted. Images were grabbed by the ROS image acquisition node with a rolling shutter at a resolution of 640x480 pixels. A M12x0.5 mount lens with a focal length of 4.6 mm was attached. Exposure was set to 50 ms, allowing for a frame rate of 20 images s^{-1}. The autofocus feature was not available under the Linux operating system and was not used.



Figure 2.3: Baxter robot (Rethink Robotics) consisting of 2 mirrored 7 degree of freedom arms.

Computer

The framework was run on a MacBook Pro, 2.4 GHz Intel Core i5 with 8GB of DDR3 memory operating on Ubuntu 12.04 Precise Pangolin.

2.3 Methods

To validate the design of the eye-in-hand sensing and motion control framework, we implemented the software functionalities described in Section 2.2.1 with the hardware described in section 3.2.1 for the dense sweet-pepper crop use case. Fur this purpose, nodes for the robot control, image acquisition, fruit detection, visual servo control and the application control were implemented to provide the required functionality. All nodes were implemented in C++ ROS version Indigo. For the image acquisition and fruit detection nodes, the machine vision library of MVtec Halcon 11.0 (MVTec Software GmbH, 2015) was used by a wrapped ROS Indigo node around the Halcon HDevEngine. Upon initialisation of the ROS nodes, a set of custom Halcon procedures were loaded into memory. The functions were not hardcoded in the source, but specified in the ROS launch file, allowing functions to be updated or replaced without recompilation of the framework. Note that open source image processing libraries, e.g. OpenCV (Culjak et al., 2012), can replace the commercially licensed Halcon library with minor efforts. The framework was tested with

the hardware in combination with an artificial dense sweet-pepper crop under laboratory conditions. In the following section, the use case is further specified, first describing the use-case specific software implementation of the framework, followed by the experimental setup of the laboratory tests.

2.3.1 Use Case Description

Sweet-pepper (*Capsicum Annum*) is a high value crop, which is currently manually harvested in high wired greenhouse cultivation systems. Due to their organised, repetitive structure, as seen on the left in Fig. 2.4 these systems are suitable for implementing robots. Unlike other crops, the fruit visibility is low in a single viewpoint due to occlusions by other plant parts (Hemming et al., 2014b). This can be see in in Fig. 2.4, where the stem and wire partially occlude the left sweet-pepper on the foreground. A green pepper on the foreground on the right is partially occluded by leaves. Furthermore, a location of a sweet-pepper can be ambiguous, as a red patch in an image can either be a wholly visible sweet-pepper in the background or a highly occluded sweet-pepper in the foreground. Solving the visibility problem in dense crops requires an approach that uses



Figure 2.4: Example photographs of a sweet-pepper crop in a Dutch high-wire greenhouse cultivation system. In the left image a front view of the cultivation system is shown. Double plant rows are separated by a workspace with a rail system to allow access to the plants. The right image shows a side view taken from that workspace facing towards the plants.

multiple viewpoints, either before or during the approach to the target fruit. Moreover, whilst executing the motion towards the target, corrections to the path are required because the dense vegetation is easily moved and the target displaced by a robot entering

the crop. Target reachability is another issue, as individual plants are spaced between 0.1-0.3 m (Bac, 2015). Obstacle avoidance therefore may be required.

2.3.2 Experimental Setup

An artificial sweet-pepper crop section was created using painted plastic imitation fruit and leaves. Although color and reflective properties were similar to real fruit for the human eye, they differ in other material properties such as hyper-spectral information and firmness. However, the materials sufficed for our purposes since only RGB analysis was required. The main veins of the leaves were fitted with a metal wire, allowing the leaves to be shaped. The leaves and the fruit were attached to a vertically placed thin pole of wood that represented a stem. By shaping the leaves, different amounts of occlusion could be realised. In Fig. 2.5 a 360° view at 45° increments of a typical setup is displayed. The visibility of the fruit depended on the perspective; the fruit were fully visible in one view and entirely occluded in another. In many views, leaves or the stems partially occluded the fruit. The objective of this experiment was to show that the robot could



Figure 2.5: Typical views at 45° increments around an artificial sweet-pepper crop used in the experiment. The backdrop is removed for clarity.

find and access the fruit. Therefore, it was sufficient that the end-effector stopped just in front of the fruit. This was assured by placing the target fruit just out of reach of the maximum robot arm stretch at 1.05 metres. The test crop was placed around 10 various of such locations, within an arc of around 0.5 m in front of the robot. The workspace towards the front of the robot, and therefore the number of test locations, was limited

because the robot's workspace was primarily designed for pick and place operations in the horizontal plane. At each location, the occlusion of the test crop was varied and multiple state machine cycles were executed. Qualitative results of the framework's performance on a dense crop were registered and these results will be discussed in the next section.

2.3.3 Use Case Specific Framework Implementation

Robot Control

Two methods for motion control of the Baxter robot were implemented. The first is a ROS actionlib service, which enabled pre-emptable tasks. With this method the status of longer movement actions could be tracked and aborted, suitable for moving to waypoints. The second method is a direct robot joint angle control, allowing to continuously change the rotation of each individual joint. This method provides short motions that can be updated during the movement, suitable for visual servo control. Both methods call Baxter's inverse kinematics ROS service. For this service, a desired pose in real world coordinates can be specified for the end-effector. The service will return a set of joint angles to move the arm to the target location, or gives feedback when unreachable or collisions are expected.

Image Acquisition

The image acquisition node implemented a connection with the camera through Halcon. A service was provided to send color or monochrome ROS images upon request. Grabbed Halcon format images were efficiently bridged to the ROS image format before sending. Furthermore this node also visualized grabbed images. In order to rectify the image, Halcon's default procedures were used for multi-view 3D calibration. For this purpose, a set of 100 images were taken of a calibration plate in various locations and orientations. Internal camera parameters were calculated and applied to a new image to remove lens distortion.

Fruit Detection

During the research project CROPS, an end-effector was developed that does not require the orientation of the fruit nor an exact position thereof for a successful harvest (Van Tuijl, B. et al., 2013; Hemming et al., 2014c). This reduces constraints on the fruit detection, allowing for more simplistic and fast approaches which are suited for visual servo control. Other approaches that calculate exact poses or use three-dimensional object matching are generally more time consuming and therefore less appropriate for

visual servo control. However, such approaches can be effectively applied, for example in grasp synthesis using active vision (Çalli, 2015; Çalli et al., 2011; Yazicioglu et al., 2009).

A previous approach (Song et al., 2014) for sweet-pepper detection classified image features. A color based classification provided the regions of interest in multiple images from which maximally stable color region features (Forssen, 2007) were extracted. Because the computational complexity and temporal performance was not reported, it is unknown if this approach can meet the time constraint in visual servo control.

To implement a simplistic and fast fruit detection for sweet-pepper, an advanced blob detection was created. It starts the analysis by converting the image from a RGB to a CIELab colorspace using the equations 2.2, 2.3, 2.4,2.5 and 2.6. Contrary to colorspaces that encode a single axis for color, CIELab has two axis for color *a,b* and one for luminosity *l*. Because the *a* axis encodes a spectrum separating green from violet-red, this channel provided distinctive contrast between red sweet-pepper and the green surroundings. Note that for other use cases or colors of sweet-pepper, a transformation to the HSI colorspace might be more suitable.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.357580 & 0.180432 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.2}$$

$$l = 116f(Y) - 16 \tag{2.3}$$

$$a = 500(f(X) - f(Y)) \tag{2.4}$$

$$b = 200(f(Y) - f(Z)) \tag{2.5}$$

where

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & \text{otherwise} \end{cases} \tag{2.6}$$

For segmentation of the sweet-pepper blobs, the a channel values ranging from 25-70 were selected. This segmentation generally contains noise, which was filtered out by a region opening operation (equal to dilation of an erosion) using a 5 pixel round element, twice as large as the noise to be filtered out. From the largest remaining region, the size and image coordinates of the centre of gravity were calculated and returned to the application control node as features. In Fig. 2.6 intermediate results of the image processing pipeline applied to an example image are displayed.

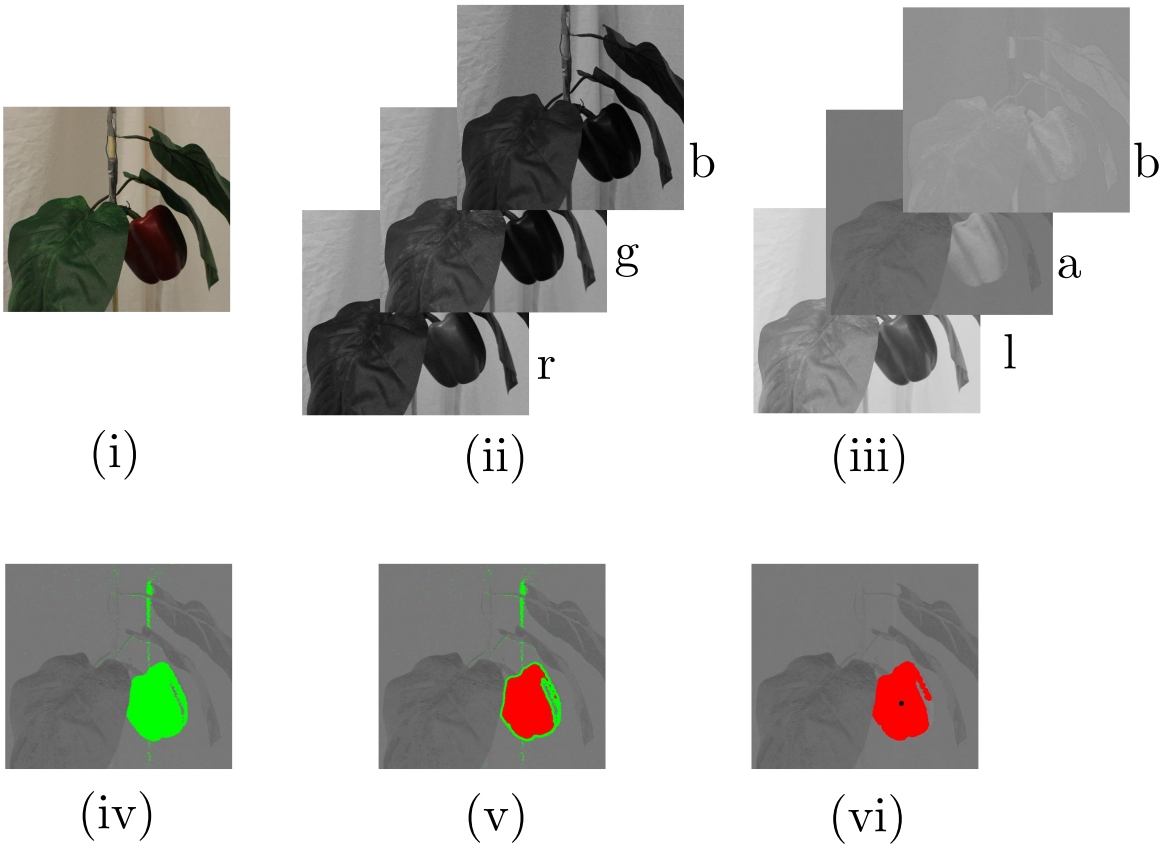


Figure 2.6: Intermediate results of the sweet-pepper detection pipeline on (i) an example color image. (ii) The image is separated in channels for red (r), green (g) and blue (b). (iii) RGB channels are converted to CIELab channels l, a and b. (iv) A threshold operation on channel a segments the sweet-pepper and some noise. (v) Noise removal by region erosion operation using an element twice as large as the noise. (vi) A region dilation operation of the same size as the shrinking operation segments the whole pepper. The size and centre of this region are returned as result features.

Visual Servo Control

To implement visual servo control for this use case, geometric features needed to be defined. However, high occlusion rates restrict the approach of deriving an object pose as a reliable feature. Instead, only parts of the fruit can be seen from a subset of all viewpoints. The centre of gravity image coordinates of the largest segmented sweet-pepper part was returned as a feature, as described in section 2.3.3. This can be used as a geometrical feature as it defines a point in two dimensional space. The desired value of this feature was set in the centre coordinates of the image.
In this application the control law for image-based visual servo control and eye-in-hand tasks in Eq. 2.1 is used. In ViSP the estimation of the interaction matrix for a 2D image feature is given by:

$$\widehat{L_s} = \begin{bmatrix} -1/z & 0 & x/z & yx & -(1+x^2) \\ 0 & -1/z & y/z & 1+y^2 & -xy \end{bmatrix}, \tag{2.7}$$

where z is either a known or estimated feature depth in the camera frame. In our application, the estimation of z had a starting value of 0.40 metres, as this was the starting distance of the crop scanning as described in section 2.3.3. This value should be updated in each visual servo cycle.

The interaction matrix also requires the positions of visual features expressed in meters rather than image pixel coordinates. For this the previously obtained camera parameters (Section 2.2.1) were used for a perspective projection without distortion model. The parameters were x-coordinate of image centre u_0, y-coordinate of image centre v_0, horizontal sensor pixel size p_x and vertical sensor pixel size p_y. If we define (u, v) as the position of a pixel in the image, then the position of that pixel in meters in the camera frame can be obtained by:

$$x = (u - u_0)/p_x \tag{2.8}$$

$$y = (v - v_0)/p_y \tag{2.9}$$

The coefficient of the exponential convergence of the error λ in the control law was set to the default value of 0.3.

Application Control

The application control node's FSM was implemented for the sweet-pepper use-case. Six states were created that each executed a sub-task in the program. The states and their transitions are displayed in Fig. 2.7. The program started in the ColdBoot state

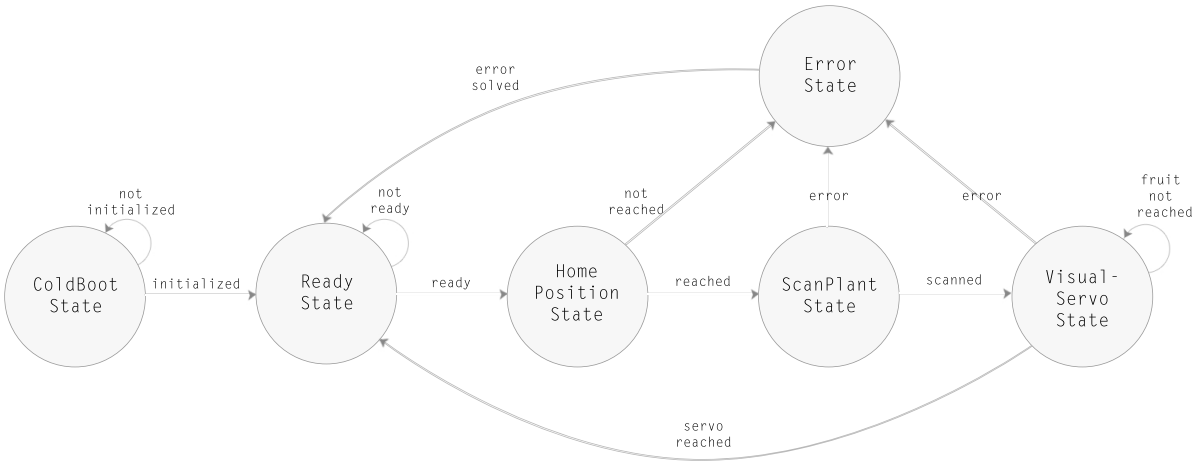


Figure 2.7: Flowchart of states and their transitions that was implemented in the finite state machine of the application control node.

where all hard- and software modules were initialised. When all modules were initialised, the state machine advanced to the Ready state that waited for an external trigger to start a harvest cycle. First, the robot moved to an initial home position, defined as 0.40 metres in front of the plant. When this position was reached, the state machine started a sensing procedure in the ScanPlant state. During this procedure the end-effector moved to predefined waypoints in the horizontal plane whilst the end-effector remained facing the plant. In Fig. 2.8 a visual representation of the procedure is provided. The waypoints were chosen following from a set of constraints consisting out i) the robot workspace, ii) representative greenhouse conditions such as a maximum distance of 0.40 meters from the target and a restricted maximum angle of approach around 90° and iii) to have at least one viewpoint with a fully occluded fruit and one viewpoint with a fully visible fruit. The motion planning involved during the plant scanning phase is a closed-loop execution of a planned motion trajectory, provided by the inverse kinematics solver of Baxter as described in section 2.2.2. In parallel, the image acquisition node was continuously triggered at 20 Hz to obtain images for (i) the SLAM node and (ii) the fruit detection node. The ScanPlant state continuously saved the pose in which the largest fruit part is detected. This pose was set as the visual servo start pose after the plant was fully scanned, under the assumption that a starting pose with a large fruit visibility from the end-effector would result in a more effective final positioning thereof.

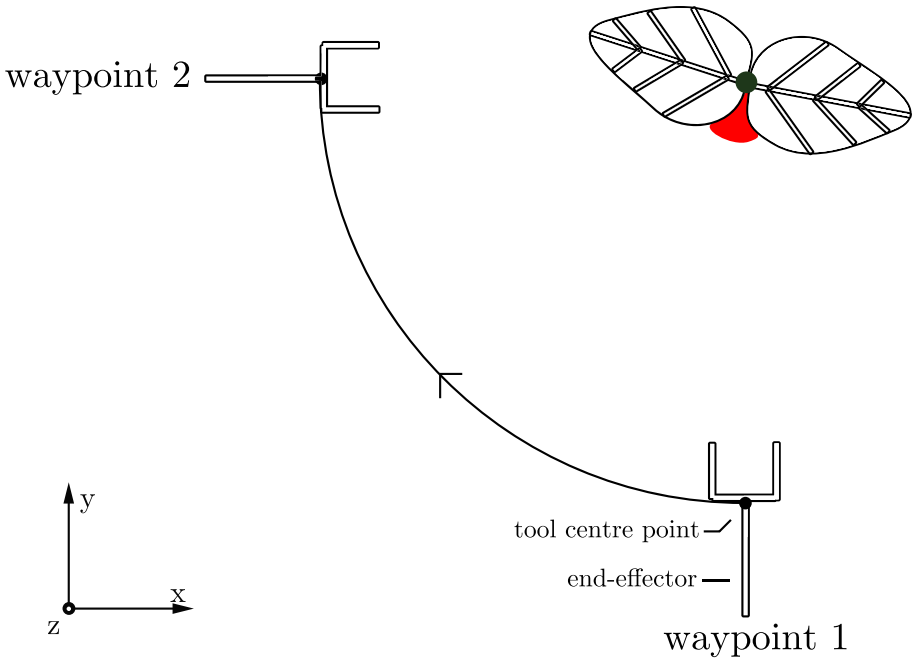


Figure 2.8: Top view of the experimental setup during the ScanPlant state. The end-effector starts at waypoint 1 and follows a trajectory towards waypoint 2 whilst the end-effector remains facing the plant and only moves in the horizontal plane *y,x*.

If no fruit was found, the state returned to the Ready state. Otherwise the visual servo control loop would commence in the Visual Servo state. Each loop cycle triggered and analysed an image for geometrical features. These features were used to calculate the pose correction vector in the camera frame and when applied to the end-effector, centres the camera with the fruit. For reaching towards the fruit, the end-effector moved along its z-axis with a constant speed of 0.03 m s^{-1}. Depth information was therefore not required. Because the fruit target was placed just outside the workspace of the robot, the fruit position was determined as reached when the arm was fully extended and the camera was centred with the fruit. The state machine then returned to the Ready state.

It is assumed that the control law improves the positioning of the end-effector in regards to the fruit centre and therefore the fruit does not leave the view during the visual servo approach. However, the latter cannot be excluded and we have yet to implement a feature that either reverts to a last known pose that includes a view on a fruit or to resume the plant scanning state.

All states after the Ready state could transition to an Error state. For example, in the Home and ScanPlant state this could occur when the given waypoint pose could

not be reached. The Visual Servo state returned to the Error state when no fruit was found during scanning. In the Error state each error could be handled depending on the transition. In this implementation it automatically returned to the Ready state whilst prompting the cause of failure.

2.4 Results

The framework was implemented and executed for a dense sweet-pepper crop use case. Software source code of the implementation of the framework is available under the the BSD License at the repository found at :
`https://github.com/rbrth/framework`.

The execution of the application resulted in the robot (i) scanning the plant for fruit first and (ii) a movement towards the centre of a fruit. A video example can be found at: `http://vimeo.com/113503359`. In Fig. 2.9 the last frame of this video is displayed, showing the fruit detection segmentation in the top left and the relative depth estimation from the SLAM node at the bottom left. On the right, the final pose of the end-effector is shown, centred with the camera towards the fruit. The execution time of a single successful execution of all states (excluding the error state) was approximately 45 s.

2.4.1 Plant Scanning

During the plant scanning state, the fruit detection node continuously analysed images from the end-effector. The achieved rate of analysis was 20 hertz, equal to the image acquisition exposure time. In most cases the visited waypoints provided sufficient viewpoints to find a suitable starting location for the visual servo control, meaning that a surface of the fruit was found.

2.4.2 Servo Control

During the visual servo control the fruit became more visible, often entirely. The end pose of the end-effector was always centred with the fruit, except for the instances in which the inverse kinematics solver of the robot failed to find a solution of the joint positions. These occasions were characterised by the robot arm already



Figure 2.9: Last frame from video at `http://vimeo.com/113503359`. The centre of the camera on the end-effector is aligned with the centre of the fruit. The top left image shows the fruit detection node's segmentation. The bottom left image shows the LSD-SLAM node's relative depth estimation. The background in the movie was removed for clarity.

being to the fully extended to its limits, but not yet horizontally or vertically aligned with the fruit. In section 3.4 the cause and solutions to this phenomenon will be discussed.

In a case where a small patch of the fruit surface was made visible from all viewpoints in the horizontal plane of the fruit, the servo control moved the end-effector in the vertical plane to centre whilst revealing more fruit surface. The result was a curved motion trajectory towards the centre of the fruit. When plant dislocation by a robot entering the crop was simulated by moving the fruit within view of the camera during visual servo control, the algorithm corrected the end-effector accordingly by following the centre of the fruit.

2.4.3 Simultaneous Localisation and Mapping

During all motions of the robot, the SLAM node ran in parallel to obtain three-dimensional information of the scene and a current estimated pose of the camera. The average publishing rate of respectively new keyframes and pose estimation was on average 5 and 10 Hz As shown in Fig. 2.10, depth estimations are primarily found on edges in the image.

This result is native to the LSD-SLAM library because it operates on image intensity differences. In Section 3.4 the usability of this result will be discussed. The bridge node

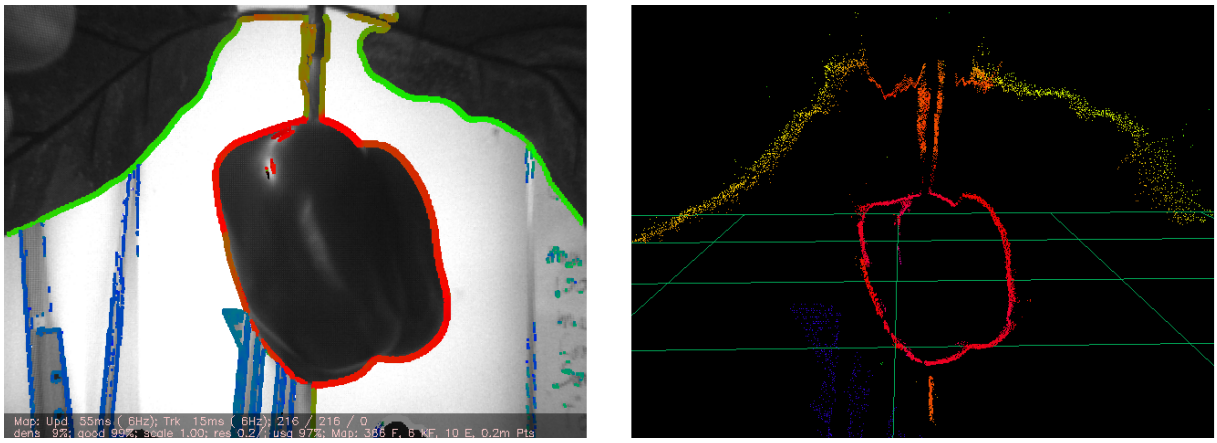


Figure 2.10: The left figure displays a monochrome image with a colored relative depth estimation overlay from the SLAM node. The figure on the right displays the respective keyframe's pointcloud in ROS RVIZ. The color gradient indicates relative depth from the end-effector.

merged and aligned multiple pointclouds from keyframes of the LSD-SLAM node, as displayed in Fig. 2.11. Again object edges are most discernible, relative depth information within objects without high texture gradients is sparsely available.

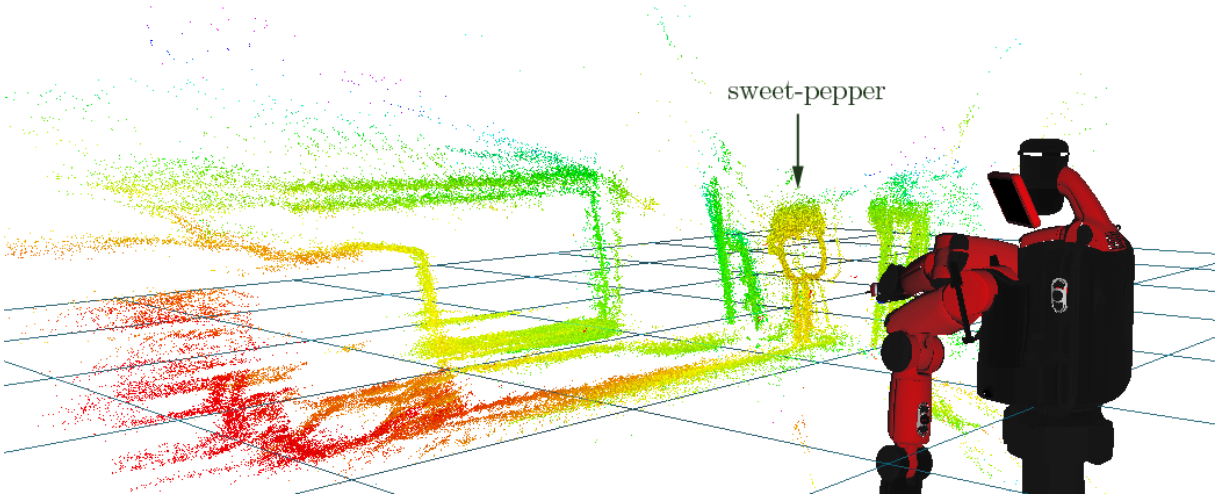


Figure 2.11: Three-dimensional scene reconstruction. Multiple LSD-SLAM keyframes from different perspectives from the end-effector were merged in a pointcloud. Visualised in ROS RVIZ with Baxter robot pose. An artificial sweet-pepper is placed in front of the end-effector, indicated by the arrow. Other structures in the background can be recognized by their edges. The color gradient indicates relative depth from the end-effector.

2.5 Discussion

The primary aim of this research was to design a framework for eye-in-hand sensing and motion control to facilitate the development of new robotic harvest applications, especially in dense crops. On a low level, many stand-alone and functionally dedicated libraries are already available to solve parts of this challenge, e.g. ROS, ViSP and LSD-SLAM. Our framework coherently integrates these parts to provide a higher level functional implementation. It can replace custom solutions by providing a standardised approach that supports a variety of use cases. Flexibility is achieved through separation of functional concerns in different modules (Felix & Ortin, 2014). One of the key aspects of the framework design was to add a high degree of implementation flexibility to meet specific use-case constraints. The secondary aim of our research was to demonstrate a framework implementation for a dense sweet-pepper crop use-case. The example implementation shows the framework can be applied for a sweet-pepper use-case with custom hardware. The added sensing functionality of 3D scene reconstruction shows the extendibility of the framework, without interference with the original software. Results also indicate that the framework can be effective for solving sensing and robot motion control in a dense crop from visual information from the end-effector, although this should be further explored in a quantitative study under greenhouse conditions.

Other approaches for solving robot sensing and motion control are not always suitable for agri- and horticulture applications due to dense crop vegetation. For sensing fruit, the occlusions of other plant parts can result in false negatives when only a single viewpoint is used (Hemming et al., 2014b; Van Henten, E. J. et al., 2002). Furthermore, using multiple poses during harvest attempts increases success rates (Henten et al., 2003). Our framework allows for the acquisition of multiple viewpoints by using the motion of the robot in combination with an eye-in-hand approach. During the motion control towards the fruit, corrections may be required when the robot displaces a target after interacting with the dense crop. It is hypothesised that additional viewpoints during motion towards the target can help resolving these problems, as well as providing more detailed information of the target. Eye-in-hand sensing is preferred as the perspective from the end-effector is likely to face the target. Some approaches already devote one or more discrete look-and-move actions to correct for displaced targets or refine rough estimated target positions (Van Henten, E. J. et al., 2002; Hayashi et al., 2010). The approach implemented in our framework uses continuous corrective actions through visual servo control, thereby able to provide more corrections and target information.

On an abstract level, the design of the framework provided a software architecture guideline to approach robot harvest system implementations. The key element of this design was to separate functional concerns to enable modularity, resulting in an system that facilitates extensions and replacements. For example, substituting a camera only affected a single node and could be achieved by replacing a single line of code without the need of recompilation. Adding a node that analyses shared resources does not require other nodes to be changed. Expanding on previous research (Hellstrom & Ringdahl, 2013; Barth et al., 2014), our implementation of this abstract level was done in the ROS middleware, which is innately modular but it did not separate concerns, or functions, automatically. Here each concern was assigned to an individual ROS node, e.g. the concern of coordination that in itself implements a modular FSM. Thus the implementation of the framework remained dependent on ROS and Linux and therefore the flexibility and usability was constrained. For ROS developers this framework is most interesting, for others it provides a useful abstract architectural design guideline. Although modules were functionally separated, they were not fully functionally independent. A notable example is the fruit detection node that was time constrained by the visual servo node. Such dependencies should be identified and avoided, for example by distributing the computation for visual servo control (Wu et al., 2010). The framework was extended with a SLAM node for depth estimation and 3D reconstruction. The nature of the LSD-SLAM algorithm is to look for differences in image intensities, therefore finding matches at texture edges depending on scene contrasts. Information is however sparse. For optimal scene reconstruction stationary scenes are needed, scene reconstruction should therefore only be used during stationary scenes, e.g. during plant scanning. Further optimisation of scene reconstruction could be implemented by pointcloud registration methods (Rusu & Cousins, 2011). Our framework provides a guideline and implementation for robotic harvest applications that enables access to visual servo control and real-time sensing in a coherent and flexible approach. The relevance of the framework is to be confirmed in other use-cases, under real conditions and with further extensions.

The framework was successfully implemented for a sweet-pepper use case and tested under controlled laboratory conditions. Results showed that the application was able to scan an occluded crop for fruit and move the end-effector towards the detected centre of a part or whole of the fruit. A single geometric feature for the servo control algorithm proved sufficient to centre the end-effector and reach the fruit. Furthermore, no depth information was required for a successful servo motion. This indicates that a simplistic

and straightforward approach can solve the motion challenge. However, relative depth estimation or descriptive image features, like fruit size, may be required to determine whether the centre of the fruit has been reached. Another possible method includes an air pressure sensor in the suction cup of the end-effector, which triggers upon fruit contact as described in (Hemming et al., 2014c). The execution time per harvest cycle can be improved. Although the Baxter robot has a maximum speed of 0.6 m s^{-1}, a high speed resulted in oversteering during our experiments. When the goal was reached, the spring kinematics of the joints damp the movement, which resulted in brief imprecise positioning. In a visual servo control loop, consecutive over- and understeering therefore produce an increasing spatial oscillation. Decreasing the speed in our experiments resulted in a more accurate movement with no oversteer, eliminating oscillations. For a real world application, it is suggested that a robot with more precise and faster joint controls is required. The use case implementation and experiments showed that eye-in-hand sensing and motion control is a viable approach for robotic harvesting of a dense crop like sweet-pepper. To further validate the use case implementation, a more advanced study under real greenhouse conditions is suggested.

Although no quantitative data was collected, the qualitative performance of the visual servo control library indicated that a more advanced study under real greenhouse conditions is viable. Whilst the artificial crop setup provided a good reflection of the occlusion problems faced in everyday practice, it remains a simplification of the real crop situation, as occlusions from stems and fruit clusters were not taken into account. Nonetheless the framework implementation showed that where a small patch was made visible, the robot managed to find the fruit and move the end-effector towards the fruit centre. This indicates that our approach can be effective under real greenhouse conditions.

The framework can potentially be implemented for robotic harvest use-cases in greenhouses like sweet-pepper, cucumber, tomato or strawberry, or in a more agricultural setting of for instance apple, citrus or broccoli. The feature of using multiple sensing viewpoints is especially functional for dense crops, as multiple viewpoints may be required for fruit detection. Although a distinction can be made between hard and soft obstacles (Bac et al., 2013), where the former (e.g stems) must be avoided at all costs but the latter (e.g. leaves) can be displaced by the robot to a certain extent, the use of visual servo control may be restricted by the presence of obstacles. Our use-case was successfully implemented using a single geometric point as feature for the visual servo control. For use-cases that require a fixed end-pose, multiple geometric features can be

used. However, this requires absolute depth information to model the interaction matrix in Eq. 2.1. The LSD-SLAM library provides relative depth information because it does not know the scale of the image. To obtain absolute depth information with this library, a calibration procedure can be performed during the initial start of the application by scanning a structure with known dimensions. In the current growing practice crops are frequently revisited to harvest newly ripened fruit, a possible future extension of this research could therefore be to use scene reconstruction to create a world model. The resulting model could be used to i) retry failed harvesting approaches, ii) skip sensing at harvested points, iii) use a crop growth model to extrapolate the position of ripened fruit and iv) update the model. For creating a world model, reconstruction could be limited to relatively stationary plant parts (e.g. fruit and stems) as opposed to frequently moving parts (e.g. leaves that follow the sun). For creating only a subset reconstruction, plant part segmentations could be used (Bac et al., 2014).

2.6 Conclusion

The significance of low level libraries that are available to the robot research community to share and build upon common functionality is evident, but individual libraries tackle one isolated concerns, e.g. ROS as communication middleware or ViSP for visual servo control algorithms. At a higher level, a framework can combine these building blocks coherently to provide an orderly structure and a new dimension of utility for a specific set of use-cases. The aim of our research was to provide such a framework for solving two key issues in robot harvesting applications. The first was sensing in dense crops with high fruit occlusions, which requires multiple viewpoints to lower the amount of false fruit detection positives. The second was the motion execution using a visual feedback loop. Implementation of this framework for a sweet-pepper use case with dense vegetation indicated viability of the framework and provided insights for further development. Future research should focus on testing the framework under real greenhouse conditions, different use-cases and by extending functionality.

2.7 Acknowledgments

We gratefully acknowledge the support of Lianne Heleen Scholtens, Joris IJselmuiden and Silke Hemming for the manuscript reviews and helpful comments. Furthermore, we would like to thank the team at RightHand Robotics; Leif Jentoft, Yaroslav Tenzer and Lael Odhner for their insights given during the research. We would also like to thank all other affiliates at the Harvard Biorobotics Laboratory and WUR Greenhouse Horticulture for providing valuable input. Special thanks go to Rob Howe, Douglas Perrin and Pierre Frederic Villard. This research was partially funded by the European Commission in the 7th Framework Programme (CROPS GA no. 246252), in the Horizon2020 Programme (SWEEPER GA no. 644313) and by the Dutch horticultural product board (PT no. 14555).

References

Bac, C. W. (2015). *Improving obstacle awareness for robotic harvesting of sweet-pepper*. Ph.D. thesis Wagenigen University & Research Centre Wageningen.

Bac, C. W., Hemming, J., & Van Henten, E. J. (2013). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture*, *96*, 148 – 162. doi: http://dx.doi.org/10.1016/j.compag.2013.05.004.

Bac, C. W., Hemming, J., & Van Henten, E. J. (2014). Stem localization of sweet-pepper plants using the support wire as a visual cue. *Computers and Electronics in Agriculture*, *105*, 111 – 120. doi: http://dx.doi.org/10.1016/j.compag.2014.04.011.

Bachiller, M., Cerrada, C., & Cerrada, J. A. (2006). *Designing and Building Controllers for 3D Visual Servoing Applications under a Modular Scheme*. Industrial Robotics: Programming, Simulation and Applications. InTech.

Bachiller, M., Cerrada, J., & Cerrada, C. (2003). A modular scheme for controller design and performance evaluation in 3d visual servoing. *Journal of Intelligent and Robotic Systems*, *36*, 235–264. doi: 10.1023/A:1023096511738.

Baeten, J., Donna, K., Boedrij, S., Beckers, W., & Claesen, E. (2008). Autonomous fruit picking machine: A robotic apple harvester. In C. Laugier, & R. Siegwart (Eds.), *Field and Service Robotics* (pp. 531–539). Springer Berlin Heidelberg volume 42 of *Springer Tracts in Advanced Robotics*.

Barth, R., Baur, J., Buschmann, T., Edan, Y., Hellstrom, T., Nguyen, T., Ringdahl, O., Saeys, W., Salinas, C., & Vitzrabin, E. (2014). Using ros for agricultural robotics : design considerations and experiences. In *Proceeding of the International Conference on Robotics and associated High-technologies and Equipment for Agriculture and forestry (RHEA)* (pp. 509–518).

Bohren, J., & Cousins, S. (2010). The smach high-level executive. *IEEE Robotics and Automation Magazine*, *17*, 18–20. doi: 10.1109/MRA.2010.938836.

Çalli, B. (2015). *Active Grasp Synthesis for Grasping Unknown Objects*. Ph.D. thesis Delft University of Technology Delft.

Çalli, B., Wisse, M., & Jonker, P. (2011). Grasping of unknown objects via curvature maximization using active vision. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 995–1001). doi: 10.1109/IROS.2011.6094686.

Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, & A. Morse (Eds.), *The confluence of vision and control* (pp. 66–78). Springer London volume 237 of *Lecture Notes in Control and Information Sciences*. doi: 10.1007/BFb0109663.

Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012). A brief introduction to opencv. In *MIPRO, 2012 Proceedings of the 35th International Convention* (pp. 1725–1730).

De-An, Z., Jidong, L., Wei, J., Ying, Z., & Yu, C. (2011). Design and control of an apple harvesting robot. *Biosystems Engineering*, *110*, 112 – 122. doi: http://dx.doi.org/10.1016/j.biosystemseng.2011.07.005.

Dean, T. L., & Wellman, M. P. (1991). *Planning and control*. The Morgan Kaufmann series in representation and reasoning. Los Altos, Calif. M. Kaufmann Publishers.

Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE*, *13*, 99–110. doi: 10.1109/MRA.2006.1638022.

Engel, J., Schops, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision ECCV 2014* (pp. 834–849). Springer International Publishing volume 8690 of *Lecture Notes in Computer Science*.

Espiau, B., Chaumette, F., & Rives, P. (1992). A new approach to visual servoing in robotics. *Robotics and Automation, IEEE Transactions on*, *8*, 313–326. doi: 10.1109/70.143350.

Felix, J., & Ortin, F. (2014). Aspect-oriented programming to improve modularity of object-oriented applications. *Journal of Software*, *9*.

Fitzgerald, C. (2013). Developing baxter. In *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on* (pp. 1–6). doi: 10.1109/TePRA.2013.6556344.

Forssen, P.-E. (2007). Maximally stable colour regions for recognition and matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (pp. 1–8). doi: 10.1109/CVPR.2007.383120.

Han, K.-S., Kim, S.-C., Lee, Y.-B., Kim, S.-C., Im, D.-H., Choi, H.-K., & Hwang, H. (2012). Strawberry harvesting robot for bench-type cultivation. *Biosystems Engineering*, *37*, 65 – 74. doi: 10.5307/JBE.2012.37.1.065.

Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., & Kurita, M. (2010). Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, *105*, 160 – 171. doi: http://dx.doi.org/10.1016/j.biosystemseng.2009.09.011.

Hellstrom, T., & Ringdahl, O. (2013). A software framework for agricultural and forestry robots. *Industrial Robot: An International Journal*, *40*, 20–26. doi: 10.1108/01439911311294228.

Hemming, J., Bac, C. W., Van Tuijl, B., Barth, R., Bontsema, J., Pekkeriet, E. J., & Van Henten, E. J. (2014a). A robot for harvesting sweet-pepper in greenhouses. In *Proceedings of the International Conference of Agricultural Engineering*.

Hemming, J., Ruizendaal, J., Hofstee, J. W., & Van Henten, E. J. (2014b). Fruit detectability analysis for different camera positions in sweet-pepper. *Sensors*, *14*, 6032–6044. doi: 10.3390/s140406032.

Hemming, J., Van Tuijl, B., Gauchel, W., & Wais, E. (2014c). Field test of differend end-effectors for robotic harvesting of sweet-pepper. In *Proceedings of the the 29th International Horticultural Congress*.

Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., Bontsema, J., & Os, E. V. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering*, *86*, 305 – 313. doi: http://dx.doi.org/10.1016/j.biosystemseng.2003.08.002.

Hutchinson, S., Hager, G., & Corke, P. (1996). A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, *12*, 651–670. doi: 10.1109/70.538972.

Jara, C. A., Pomares, J., Candelas, F. A., & Torres, F. (2014). Control Framework for Dexterous Manipulation Using Dynamic Visual Servoing and Tactile Sensors' Feedback. *Sensors (Basel, Switzerland)*, *14*, 1787–1804.

Kitamura, S., & Oka, K. (2005). Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture. In *Mechatronics and Automation, 2005 IEEE International Conference* (pp. 1807–1812 Vol. 4). volume 4. doi: 10.1109/ICMA.2005.1626834.

Kröger, T. (2010). *On-Line Trajectory Generation in Robotic Systems* volume 58 of *Springer Tracts in Advanced Robotics*. Berlin, Heidelberg, Germany: Springer.

Mahony, R. (2011). Modular design of image based visual servo control for dynamic mechanical systems. In *Proceedings of International Symposium on Robotics Research* (pp. 1–16). Springer.

Marchand, E. (1999). Visp: a software environment for eye-in-hand visual servoing. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* (pp. 3224–3229 vol.4). volume 4. doi: 10.1109/ROBOT.1999.774089.

Marchand, E., Spindler, F., & Chaumette, F. (2005). Visp for visual servoing: a generic software platform with a wide class of robot control skills. *Robotics Automation Magazine, IEEE*, *12*, 40–52.

Mehta, S., & Burks, T. (2014). Vision-based control of robotic manipulator for citrus harvesting. *Computers and Electronics in Agriculture*, *102*, 146 – 158. doi: http://dx.doi.org/10.1016/j.compag.2014.01.003.

Murphy, R. R. (2000). *Introduction to AI Robotics*. (1st ed.). Cambridge, MA, USA: MIT Press.

MVTec Software GmbH (2015). Halcon. url: `http://www.halcon.com/`.

Nguyen, H., Ciocarlie, M., Hsiao, K., & Kemp, C. (2013). Ros commander: Flexible behavior creation for home robots. In *ICRA*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T. B., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Rusu, R., & Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 1–4). doi: 10.1109/ICRA.2011.5980567.

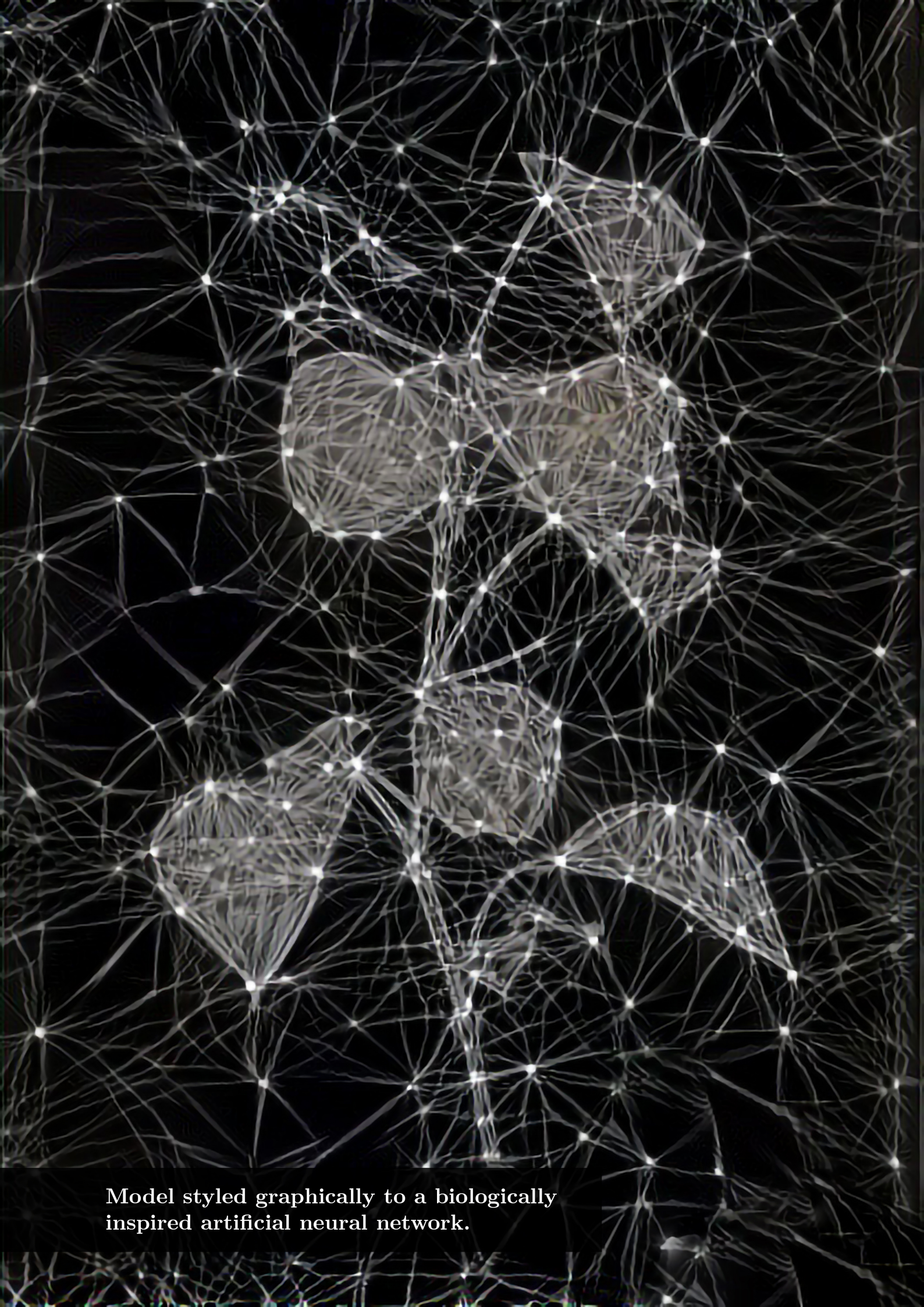
Song, Y., Glasbey, C., Horgan, G., Polder, G., Dieleman, J., & van der Heijden, G. (2014). Automatic fruit recognition and counting from multiple images. *Biosystems Engineering*, *118*, 203 – 215. doi: http://dx.doi.org/10.1016/j.biosystemseng.2013.12.008.

Van Henten, E. J., Hemming, J., Van Tuijl, B., Kornet, J., Meuleman, J., Bontsema, J., & Van Os, E. (2002). An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, *13*, 241–258. doi: 10.1023/A:1020568125418.

Van Tuijl, B., Wais, E., & Yael., E. (2013). Methodological design of an end-effector for a horticultural robot. In *Proceedings of 4th Israeli Conference on Robotics*.

Wu, H., Lou, L., Chen, C.-C., Hirche, S., & Kuhnlenz, K. (2010). A framework of networked visual servo control system with distributed computation. In *Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on* (pp. 1466–1471). doi: 10.1109/ICARCV.2010.5707837.

Yazicioglu, A., Çalli, B., & Unel, M. (2009). Image based visual servoing using algebraic curves applied to shape alignment. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (pp. 5444–5449). doi: 10.1109/IROS.2009.5354310.



Model styled graphically to a biologically inspired artificial neural network.

Chapter 3

Data Synthesis Methods for Semantic Segmentation in Agriculture: a *Capsicum annuum* Dataset

This chapter is based on:

Barth, R., IJsselmuiden, J., Hemming, J., and van Henten, E.J. van (2018b). Data synthesis methods for semantic segmentation in agriculture: A *Capsicum annuum* dataset. *Computers and Electronics in Agriculture*, 144, 284 – 296.

Abstract

This chapter provides synthesis methods for large-scale semantic image segmentation datasets of agricultural scenes with the objective to bridge the performance gap between state-of-the art computer vision performance and computer vision in the agricultural robotics domain. We propose a novel methodology to generate renders of random meshes of plants based on empirical measurements, including the automated generation per-pixel class and depth labels for multiple plant parts. A running example is given of *Capsicum annuum* (sweet or bell pepper) in a high-tech greenhouse. A synthetic dataset of 10,500 images was rendered through Blender, using scenes with 42 procedurally generated plant models with randomised plant parameters. These parameters were based on 21 empirically measured plant properties at 115 positions on 15 plant stems. Fruit models were obtained by 3D scanning and plant part textures were gathered photographically. As reference dataset for modelling and evaluate segmentation performance, 750 empirical images of 50 plants were collected in a greenhouse from multiple angles and distances using image acquisition hardware of a sweet pepper harvest robot prototype. We hypothesised high similarity between synthetic images and empirical images, which we showed by analysing and comparing both sets qualitatively and quantitatively. The sets are publicly released with the intention to allow performance comparisons between agricultural computer vision methods, to obtain feedback for modelling improvements and to gain further validations on usability of synthetic bootstrapping and empirical fine-tuning. We provide a brief perspective on our hypothesis that related synthetic dataset bootstrapping and empirical fine-tuning can be used for improved learning.

3.1 Introduction

3.1.1 Research aim

In recent years the need of robotisation in agriculture has been growing notably to keep up with the increasing demand of productivity and quality of food production whilst decreasing the pressure on resources required (Bac et al., 2014b). Although mechanisation has been an ongoing human effort for centuries, the next leap forward to achieve these higher goals is by adding a degree of artificial intelligence to harvesting and crop management systems to enable increased selectivity, precision and robustness.

We identified one of the main current bottlenecks for introducing robotics in agriculture as the computer vision performance in this domain. In the past decade, the general field of computer vision made significant progress in object localisation and consecutively was successfully applied in many domains. However, this performance has not been matched for sensing solutions in agriculture (Gongal et al., 2015; Nasir et al., 2012). We argue that one of the main reasons is the absence of detailed and large annotated agricultural datasets that current state-of-the-art methods require, but are infeasible to obtain manually

Accordingly, to contribute to solving this bottleneck and move the field forward, we provide a method for artificial agricultural data synthesis. We hypothesise it is possible with this approach to generate synthetic images highly similar to empirical images. Specifically, this chapter introduces a method for the generation of large-scale semantic segmentation datasets on a plant-part level of realistic agriculture scenes, including automated per-pixel class and depth labeling. One purpose of such synthetic dataset would be to bootstrap or pre-train computer vision models, which are fine-tuned thereafter on a smaller empirical image dataset (Dittrich et al., 2014; Kondaveeti, 2016). Our methodology is designed to be extended to other plants, but for reference a running example is given for a *Capsicum annuum* species, also known as sweet (or bell) pepper. An empirical photographic dataset was gathered and partially annotated to i) use as reference for modelling and ii) to verify the performance of any computer vision method that used the synthetic data for bootstrapping and applied to the empirical images.

In the following sections we report on the methodology for the data synthesis, with the requirement of similarity with the empirical dataset. The method starts with the modelling of a structural plant model using empirical plant parameter measurements and images. This model was used for generating randomised mesh instances of plants, which were

imported into render software to mimic scenes of a commercial agricultural environment. To synthesise color and per-pixel label and depth data, these scenes were rendered with similar characteristics as the hardware used in a harvesting robot prototype.

In the results section examples are given for both the synthetic and empirical datasets. Although subjectively these sets are comparable, the sets were analysed for differences in distributions of colors per class to verify our first hypothesis of similarity. To test our second hypothesis that the datasets can be used for improved learning, 5 experiments were performed using a basic semantic segmentation deep learning network. However, the scope of this chapter was primarily on the synthetic image generation methodology. Therefore we discussed the machine learning part briefly to give future perspective on our follow-up companion Chapter 4. In that chapter we will more extensively discuss the impact of synthetic data for semantic segmentation of plant parts and the requirements. At the end of this chapter we discussed the challenges and limitations of this approach for realistic data generation and its potential use for computer vision. We conclude the chapter by making the used scripts, models and datasets publicly available. The objective of this release is to i) enable comparison of state-of-the-art computer vision methods for this domain, ii) further validate the approach of synthetic modelling and empirical fine-tuning and iii) gain knowledge on modelling deficiencies and improvements.

3.1.2 Research context

In the domain of agriculture, progress in image classification performance has been lagging behind state-of-the-art results in other domains (Gongal et al., 2015; Nasir et al., 2012). Although some progress in high level object detection or localisation have lately been accomplished (Sa et al., 2016), lower level detailed object part recognition in scenes reflecting realistic structural object complexity, remains unsolved. The challenges of computer vision in the agricultural domain come from the high amount of variation within object classes and changing environment conditions, throughout the day and seasons (e.g. light, growth stages). To overcome this variability, large annotated and detailed datasets are needed for capture all different situations. Although collecting image data can be automated (van der Heijden et al., 2012), it remains required and time consuming to manually annotate.

Synthetic image dataset generation methods are emerging as an important tool in the computer vision community to automatically create annotated training data for bootstrapping machine learning models (Dittrich et al., 2014; Kondaveeti, 2016). Consecutively, such models can be fine-tuned by and applied to empirical image data. Recent examples showing improved object recognition performance can be found in multiple domains, e.g. urban scene segmentation (Ros et al., 2016), 3D human pose estimation from depth images (Shotton et al., 2013) and multi-modal magnetic resonance imaging for pathological cases (Cordier et al., 2016).

Previous work on methods for plant architecture modelling have been also successful for synthetic plant image generation. For example, OpenAlea (Pradal et al., 2015, 2017) is able to generate anatomical and functional plant models and furthermore be used to simulate images with a virtual camera. Other approaches such as ElonSim (Benoit et al., 2014) provide a simulator of plant growth, specifically root systems, and a simulator of the image acquisition to generate synthetic images including ground truth. The simulator uses plant and camera parameters. Furthermore, recently a method was created for automatic model based synthetic dataset generation for crop and weeds detection on a per-pixel level (Cicco et al., 2016), though no plant parts could be differentiated.

The required level of labeling detail depends on the task and in turn determines how much annotation effort is needed. One approach is to only label images on a high level using a single class or a few keywords per image in order to classify an image globally or give a shortlist of objects in the image (Everingham et al., 2015). This can be partially automated through combined image and label retrieval using context from search engines (Fergus et al., 2005). For other datasets like ImageNet or PASCAL VOC, manual annotation was performed using crowd sourcing (Everingham et al., 2010; Russell et al., 2008). A second approach is to weakly label the data with bounding boxes around objects or their parts (Papandreou et al., 2015). However, some computer vision tasks require a lower, per pixel level labeling of the image, also known as semantic segmentation. Specifically for agriculture, per-pixel segmentations are required for localisation in robotics for harvesting (Bac et al., 2013), disease detection (Polder et al., 2014) and phenotyping (van der Heijden et al., 2012). For example in harvest robotics, obstacle maps on the plant part level resolution improves successful motion planning. (Bac et al., 2016, 2014a). Registered depth images can provide an additional dimension for motion control (Barth et al., 2016).

With the advent of state-of-the-art machine learning methods for computer vision, most notably convolutional neural networks for image classification and segmentation, the train-

ing dataset size requirement has been further increased (Najafabadi et al., 2015). Such learning models can have up to 10^{11} free parameters (Dean et al., 2012), which depend upon a large number of distinct data samples for the optimisation to converge properly without overfitting to occur (Trask et al., 2015). Without access to large datasets, domains such as agriculture previously used traditional computer vision methods using manual feature crafting (Bolón-Canedo et al., 2013) whilst capturing a limited subset of the variability that occurs. Our aim is to facilitate the agricultural computer vision domain with the benefits of state-of-the-art machine learning, e.g. the supervised hierarchical feature representation learning and the performance increase that comes with large datasets (LeCun et al., 2015).

3.2 Materials and Methods

In Figure 3.1 our method to obtain the synthetic and empirical datasets is shown in a flowchart. Empirical data was a cornerstone for two objectives. First, it was used as a reference to create both a realistic model and conditions to render the synthetic dataset. Second, to provide fine-tuning data and a verification test set for computer vision methods that use the synthetic dataset for bootstrapping. This section provides some intermediate results as prerequisite for consecutive methods; the final results of the synthetic and empirical datasets are reported on in Section 5.6.

3.2.1 Empirical Reference Dataset and Scans

The empirical photographic image dataset was acquired using imaging hardware of a sweet pepper harvest robot prototype, consisting of a uEye SE industrial camera (UI-5250RE-C-HQ PoE Rev.2, GigE, Germany) with resolution of 1600x1200 pixels and a lens with focal length of 4.16 mm (CMFA0420ND, Lensagon, Germany). The scene was illuminated with a matrix of white LEDs, flashed for 50 μs, producing a light level of approximately 200.000 lx at a distance of 50 cm from the crop. The distribution of the light was highly centered with a sharp falloff towards the edges in the field of view of the camera. By using the flash, the global illumination was suppressed, although this resulted in dark images.

From distances ranging from 50 cm to 10 cm (in 10 cm increments) in front of the plant stems, images were captured from -45, 0 and 45 degree orientations with the horizontal plane. Furthermore, the camera was angled 20 degrees upwards as previous research

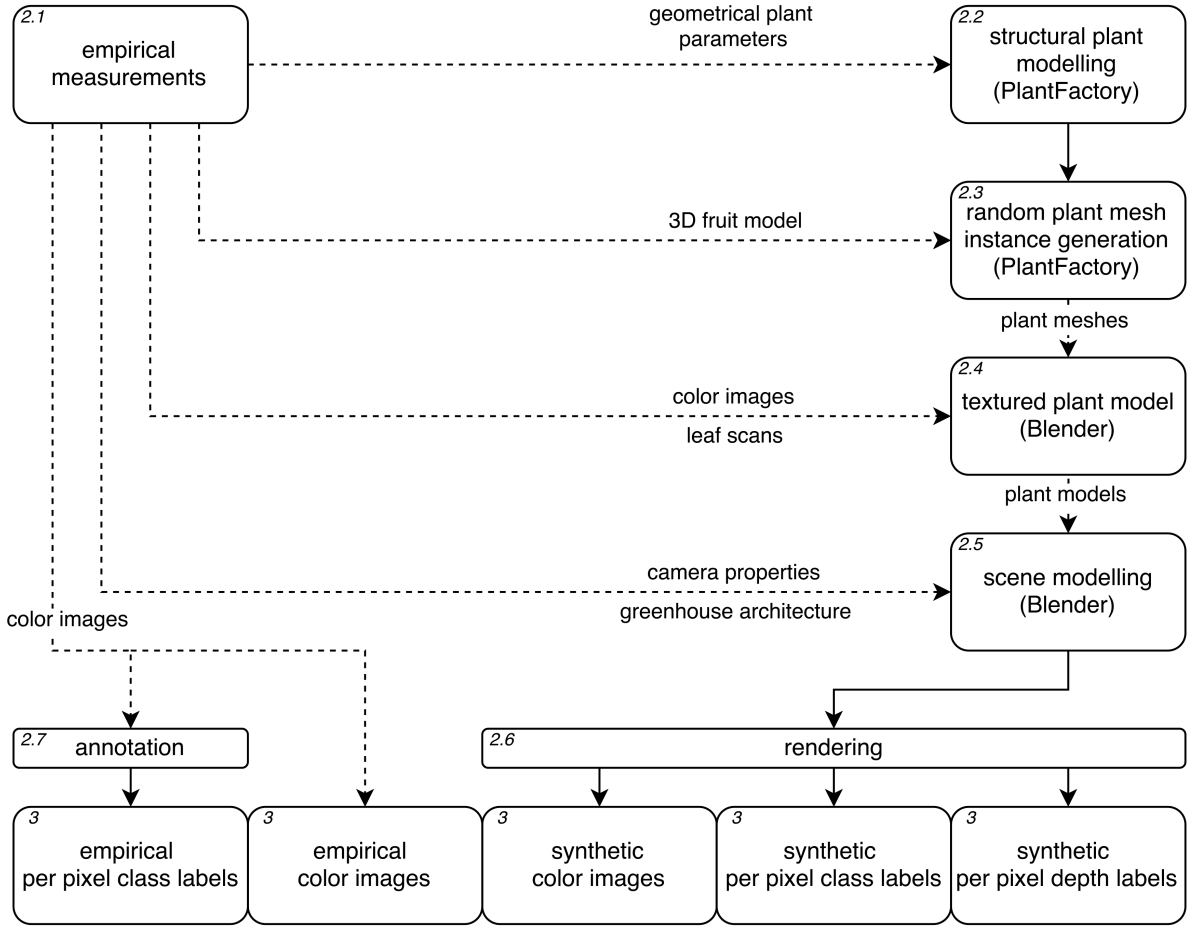


Figure 3.1: Methodological flowchart to obtain empirical and synthetic datasets for agricultural plant scenes. Empirical measurements feed information to the plant and scene modelling processes. First a structural model of the plant was created using geometrical plant parameters. The structural model was then used to create instances of polygonal plant meshes that included a 3D mesh scans of fruit. The meshes were imported to Blender, where color images and leaf scans textured the model. Multiple plant models were thereafter included in scenes which mimiced the greenhouse architecture. Camera and illumination properties were added as well. The scenes were then rendered to obtain the sythetic data. A subset of the empirical color images was annotated intended for computer vision fine-tuning and verification test material. Each box is described in a section as referred by its top left number.

suggested this would reduce occlusions (Hemming et al., 2014). In total 50 incremental positions of 20 cm along the row of plants were imaged in a 800x600 binned pixel resolution under two conditions; i) facing towards and ii) away from the sun. The increment size of 20 cm along the row was chosen to reflect the plant spacing in the greenhouse in order to approximate one new plant in the field of view per image. Overall weather conditions were clear and sunny with occasional clouds (cumulus humilis). At the position of the camera, the average irradiance was 6,000 lx under a clear sky and 5000 lx when the sun was occluded by a cloud.

3D meshes were obtained of 3 sweet pepper fruit, cultivar Kaite (E20B.0073, Enza Zaden, the Netherlands) with a Spider 3D scanner (Artec, Luxembourg) with a 3D point accuracy of 0.05 mm. Scanning was performed manually by covering all perspectives using a rotating platform. Bottom occlusions were solved by using multiple poses of the same fruit and merging the resulting meshes automatically with Artec's software package. A set of 10 leaves were flattened and scanned using a consumer flatbed color scanner to obtain their shape, color and texture. At the nodes of the plant, occasionally there were cuts present where fruit were harvested. Frontal photographs were taken to obtain textures for this plant part and similarly for the stem.

3.2.2 Structural Plant Modelling

A plant can be modelled functionally, structurally, or both (Vos et al., 2007, 2010). Functional plant models represent the interaction of internal and external plant processes. On the other hand, structural plant models focus solely on the physical appearance. When both types of models are combined, the influences of processes on the plant structure are taken into account. The scope of the current dataset was purely structural, as only fixated images irrespective of other influences were modelled.

Structurally, a plant consists of elements of various types and shapes. Based on these elements, a plant architecture can be defined globally or modularly. A global architecture is considered as a single shape and such an architecture inhibits plant variability on a detailed plant part level. In contrast, a modular plant architecture consists of the combination of three types of information; (i) the decomposition information that describes which components a plant consist of, (ii) the topological information that characterises the hierarchy and connection of components with others and (iii) the geometrical information that describes the sizes and poses of the components irrespective of other plant parts (Godin, 2000). With this decomposition either a regular or a multi-scale represen-

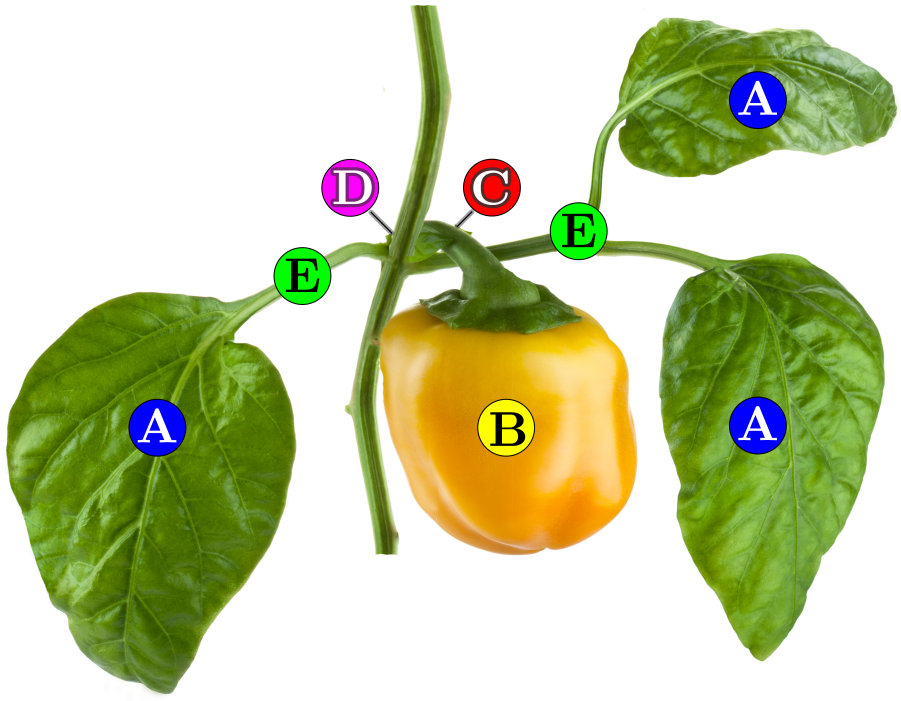


Figure 3.2: A section of a Capsicum Annuum plant with one node in the center from which all plant parts grow. Plant parts are: A) leaf, B) fruit, C) peduncle, D) a node at a stem section and E) sideshoot or leaf stem.

tation of a plant can be created. In the latter case self-similarity at different levels in the plant hierarchy can occur. In our approach, we created a regular structural modular plant model with a multi-scale representation for side shoots of the plant. This enabled detailed part modelling in which empirical measurements could be included and variability could be expressed.

Decomposition of Capsicum annuum

We decomposed the sweet pepper plant in the following plant parts: stem section, nodes, sideshoot, leaf stem, leaf, peduncle, fruit and flower. To facilitate robotic harvesting, our model focussed on the generative stage of the crop only. At this stage most flowers have been pollinated and only fruit remain. Therefore the flower was omitted from the model.

Topology of Capsicum annuum

In Figure 3.2, a typical section of a sweet pepper plant is shown, with a node in the center. From this empirical situation, we defined our hierarchy of plant components, as shown in Figure 3.3.

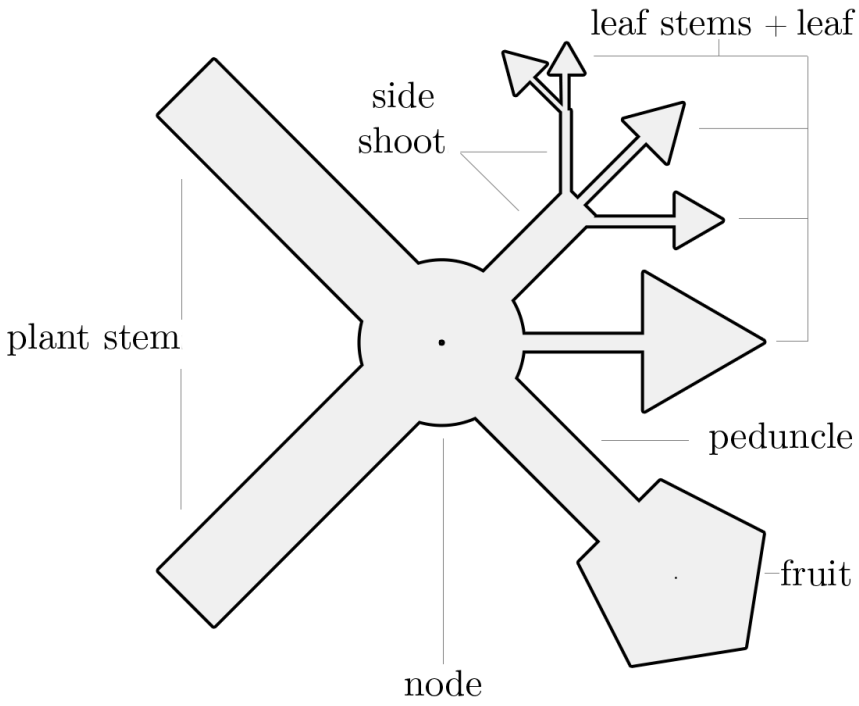


Figure 3.3: Structural modular plant topology of *Capsicum annuum*, showing the hierarchy and connection of plant elements. We considered the node as the central part of each section of the plant. The node joins 2 stem sections and connects to (i) sideshoots, (ii) leaf stems and leaves, (iii) peduncle and fruit. A side-shoot can have up to 3 leaves or new side shoots at it's end. This topology omits the flower, which has grown into a peduncle and fruit.

Geometry of Capsicum annuum

The geometry of a plant describes its parts in terms of dimensions and poses irrespective of other parts. Similar to other approaches for *Capsicum annuum* (Ballina-Gomez et al., 2013; , IPGRI), 22 relevant plant parameters were identified that capture the geometry between plant parts. These parameters were measured in the early production season (April) on 15 plants of the same cultivar as used for the collection of the empirical image dataset and the scanned fruit and leaves. The collection of the measures included length, width, diameter and angles. Top view angles of plant parts were measured around the stem, counter-clockwise starting from the anterior side of the plant (hence perpendicular from the aisle towards the plant row) as depicted in Figure 3.4. Side view angles were measured counter-clock-wise, perpendicular with the floor as reference plane as shown in Figure 3.5. Results of the measurements are provided in Table 3.1 with the angular distributions plotted in Figure 3.6.

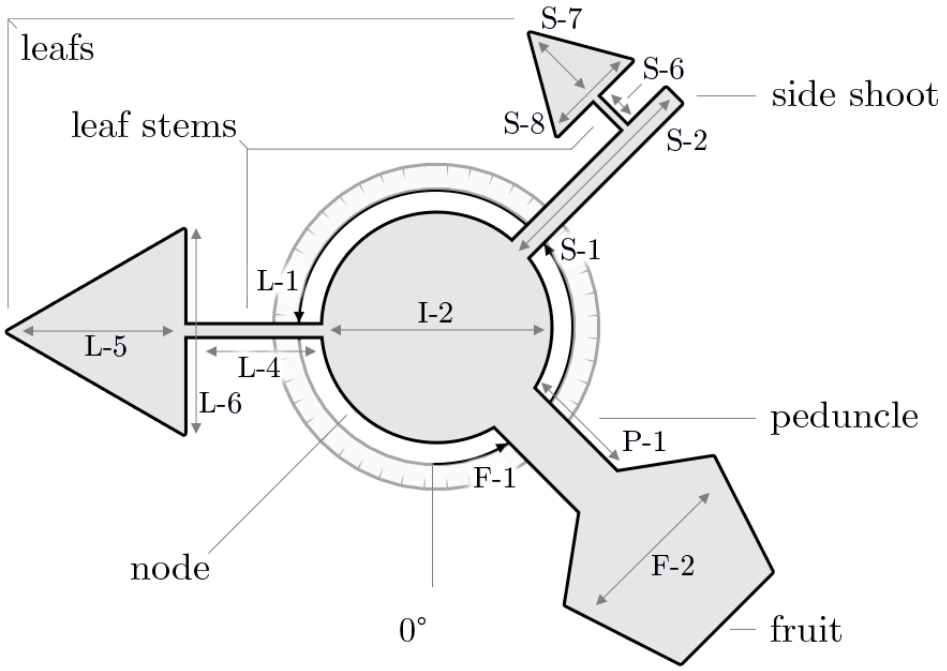


Figure 3.4: Schematic top view of *Capsicum annuum*. An intersection is shown at the level of an node of the plant. Parts are connected around the plant: i) peduncle and fruit, ii) leaf stem with leaf, iii) side shoot with leaf stem and leaf. Geometrical plant parameter measures are depicted, as reported in Table 3.1. Angles F-1, S-1 and L-1 are measured counter-clockwise starting from the anterior side of the plant in respect to the greenhouse aisle.

Name	Plant Part	Measure	Average (mm)	SD (mm)
I-1	NODE	internode length	99	14
I-2	NODE	node width	16	2
L-4	LEAF	stem length	99	14
L-5	LEAF	leaf length	16	2
L-6	LEAF	leaf width	112	11
S-1	SIDESHOOT	length	48	30
S-6	SIDESHOOT	leaf stem length	97	12
S-7	SIDESHOOT	leaf length	123	16
S-8	SIDESHOOT	leaf width	102	12
P-1	PEDUNCLE	length	46	6
F-2	FRUIT	diameter	84	11
P-1	PLANT	stem diameter	9	1

Table 3.1: Geometrical *Capsicuum annuum* plant parameters per plant part that were measured in 15 plants at 115 node positions. Averages and standard deviations were used for modelling in PlantFactory. Descriptive names are displayed in Figures 3.4 and 3.5.

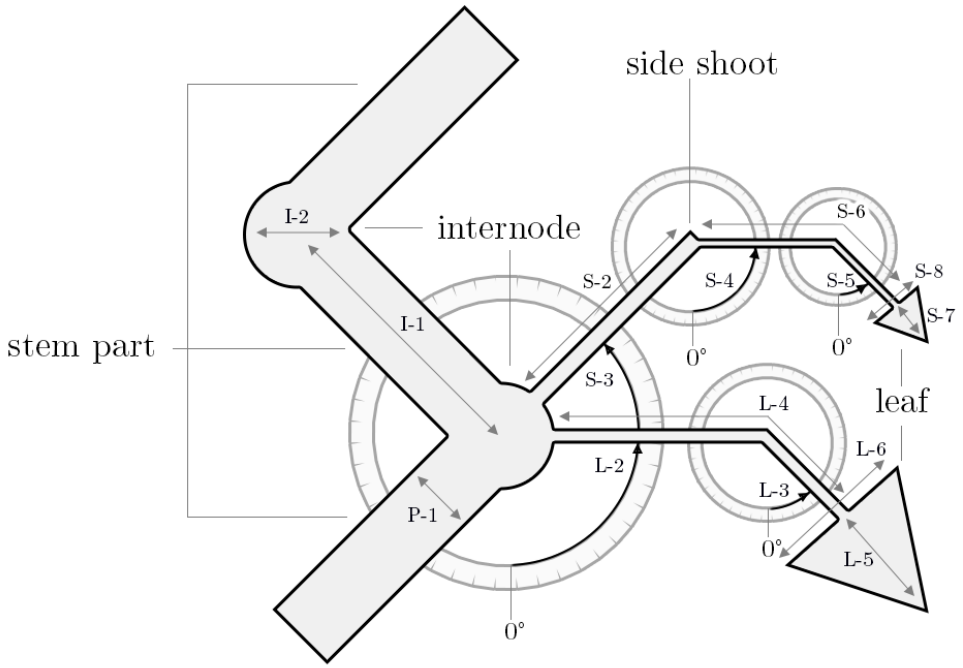


Figure 3.5: Schematic side view of *Capsicum annuum*. Geometrical plant paramter measures described in Table 3.1 are depicted. Angles L-2, L-3, S-3, S-4 and S-5 are measured perpendicular to the ground and counter-clockwise.

3.2.3 Plant Instance Modelling

To model the plant and create object meshes, the commercial software PlantFactory 2015 Studio (EON Software, 2016) on OSX 10.10 was used. Originally intended for realistic game and video production modelling, it includes functionality to generate randomised plant instances based on plant parameter distributions, which can be exported as mesh models.

Randomised instances of sweet pepper plant meshes were generated by a procedural algorithm within PlantFactory 2015. The structural modular plant topology of Figure 3.3 was used as a guideline. In Figure 3.7 the procedural structure of the algorithm is shown. Plant part parameters of the obtained geometry from Section 3.2.2 were used. The metric parameters were based on the averages and standard deviations. For the angular values, their distributions were used as shown in Figure 3.6, imported to PlantFactory as a curve. Per plant, 40 stem parts were generated and concatenad, resulting in a mesh of approximately 4 m in height. Mesh instances were manually checked for the occasional inconsistency when a fruit intersected with other meshes. Those instances were replaced. Each mesh instance was exported in the open Wavefront OBJ 3D model format. In Figure 3.8 example meshes are shown with 5 stem parts.

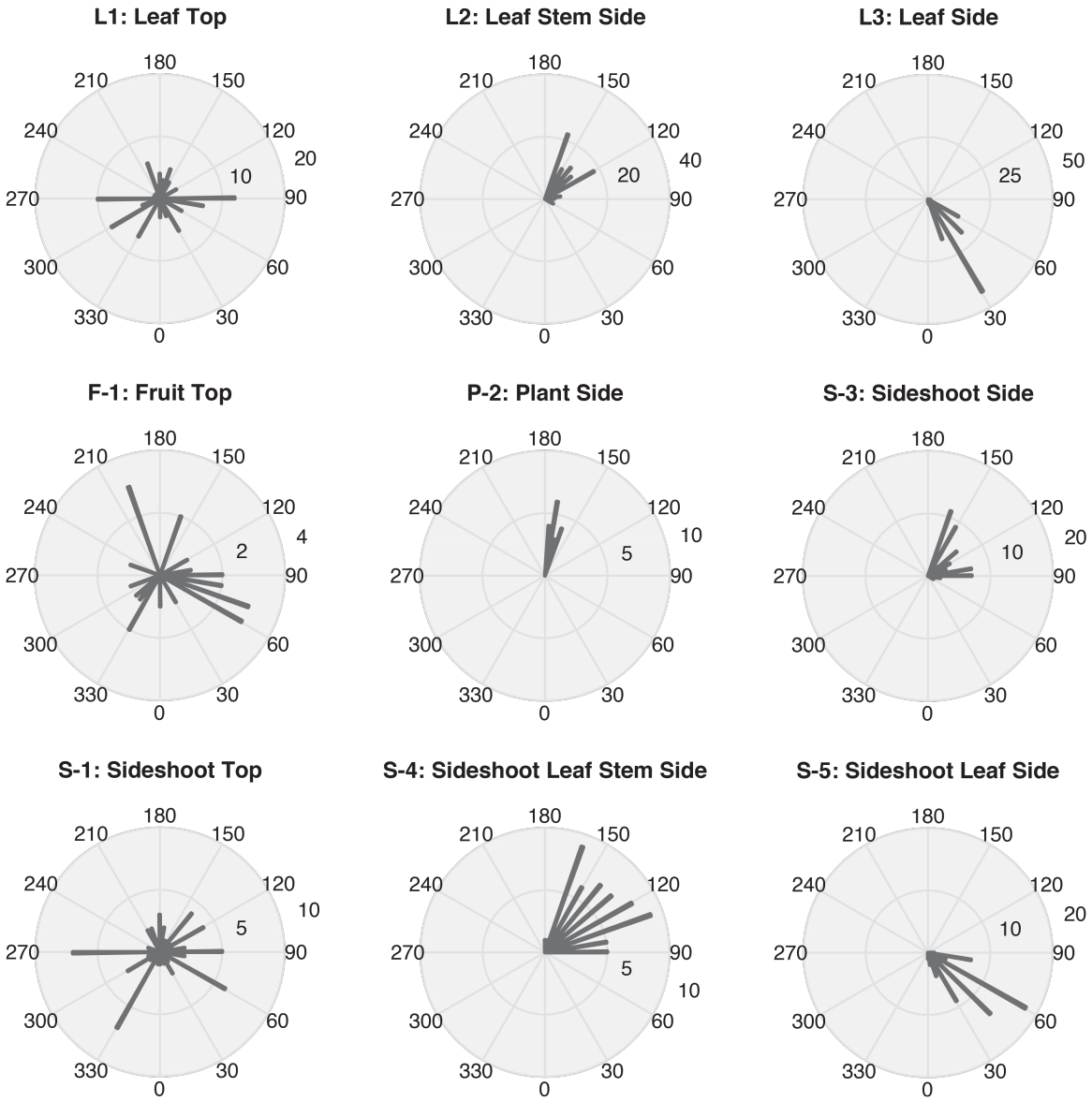


Figure 3.6: Angle distributions (number of occurences per 0-360°) for *Capsicum annuum* plant parameters as measured of 15 plants according the schematics in Figure 3.4 and 3.5. P-2 measured the angle of the total plant in the plane perpendicular to the plant rows.

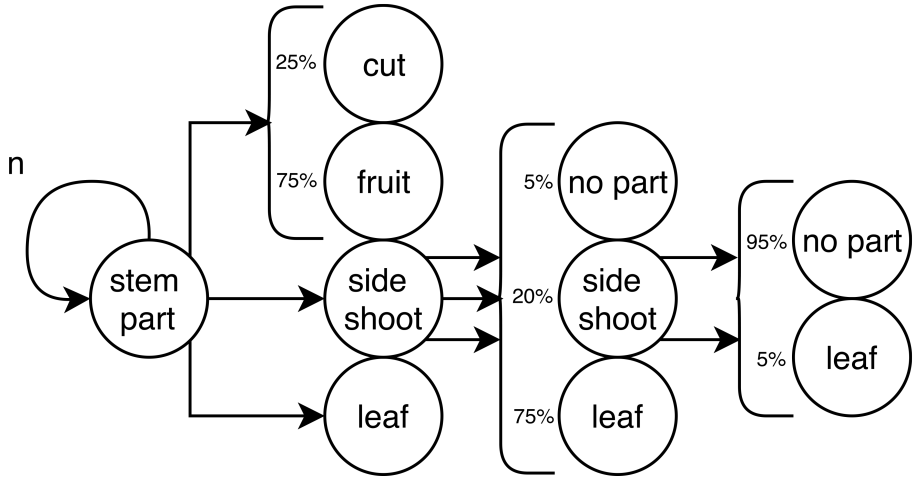


Figure 3.7: Procedural structure for plant instance generation as implemented in PlantFactory. Each node represents a plant part for which another plant part is generated for each arrow attached. Brackets imply a random choice was made between plant parts included with indicated probabilities. The stem part grew another stem part n times.

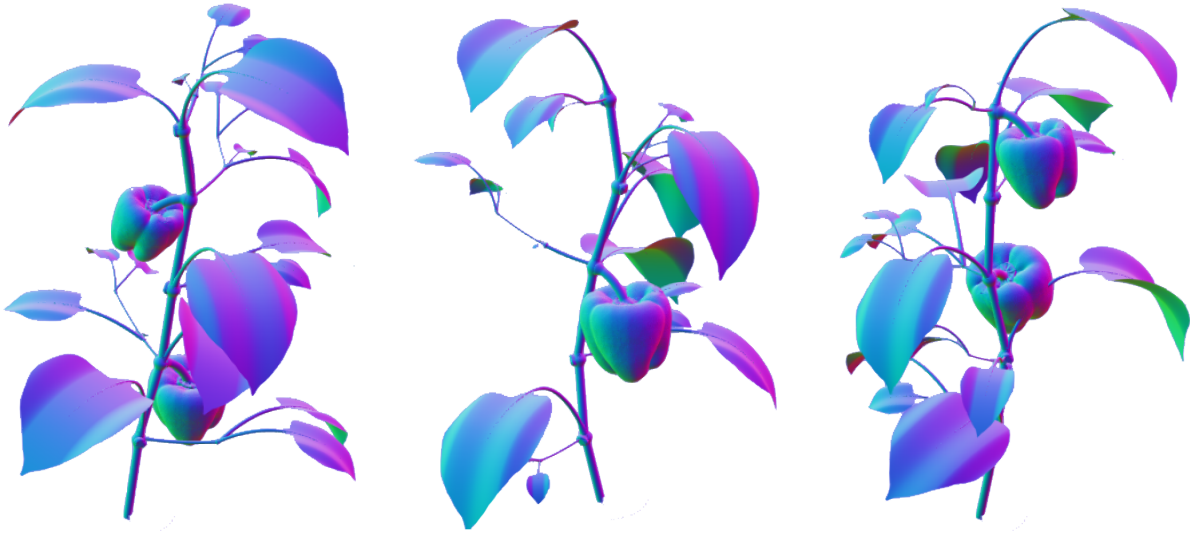


Figure 3.8: Perspective views of three meshes of randomly generated *Capsicum annuum* instances with 5 stem parts. Color encodes surface normals.

3.2.4 Textured modelling from polygon meshes

The randomly generated plant meshes were imported in the open-source software Blender 2.77a (Blender Foundation, 2016; Kent, 2015), which for our purpose supports the composition, simulation and rendering of 3D scenes. To the polygon meshes we added color, texture, local mesh displacement (bump mapping), glossiness and specularity properties.

For the leaves and the cuttings on the stem where fruit were previously removed, the photocopies were used. For the other plant parts, a color overlay was applied, based on average colors from patches of corresponding plant parts in the empirical data set. In order to simulate fruit maturity levels, a color gradient from unripe green to ripe yellow was projected on a noise texture. The gradient and noise parameters were manually determined with reference to additional unripe to ripe fruit images taken in the greenhouse. To simulate local leaf deformations (wobbles), a noise texture was used for bump displacement mapping of the leaf meshes. The parameters of this displacement were visually determined in comparison with the empirical image dataset. The polygons of the stem parts were displaced with a flattened 3D scan of the stem, processed with an edge filter to simulate their vertical grooves. For each set of plant parts, light reflection was manually modelled by adding glossiness and specularity modifiers. To add a background, a partial cloudy sky was generated. The sun was modelled as a light emitting sphere and was placed in the background with a lens flare effect. To complete the scene, the vertical wire used in horticultural practice was to support the plant was modelled by applying a white texture and bump map that curled around the stem.

3.2.5 Scene Modelling

Using Blender, a scene was modelled that represented a part of a Dutch commercial high-tech sweet pepper greenhouse (Bac et al., 2016) by reproducing the plant growing architecture as a double row of plants. In Figure 3.9 a frontal and top perspective view of the scene is shown. For each scene, 7 randomised plants were generated, imported and positioned in a row with a 20 cm spacing in between. Similar to horticultural practice, a second row of 6 random plants was added 20 cm behind the first, shifted 10 cm in parallel. Due to memory constraints, up to 13 plants could maximally be instanced in each scene. To virtually collect data, a simulated camera and illumination was added with similar optical properties as the hardware described in Section 3.2.1. Blender allows to set the focal length and sensor size according to the hardware manufacturers specifications. The

illumination intensity and distribution was empirically matched with the reference color images.

The simulated image acquisition hardware followed an arc path upwards at a fixed distance of 40 cm from the center of the 4th plant in the row of 7 plants. Along this path, 250 frame triggers were equally spaced. The camera was placed under an angle of 20 degrees looking upwards, as also used when obtaining the empirical dataset in the greenhouse.

To create multiple scenes per set of 7 plants, each plant was translated 1 position in the row until all 7 plants had once occupied the center of the row at the location where the simulated data collection occurred. The plants in the second back row also translated along. Furthermore, 6 additional unique scenes were generated with 7 new plants. Hence in total, 42 scenes were created (6 scenes, 7 positions per scene).

3.2.6 Rendering

For each scene, render computations (color, class label and depth label images) were run on the Odyssey supercomputer cluster supported by the Research Computing Group of the FAS Division of Science at Harvard University. Each frame was assigned to a single computing node with 16 cores, rendering one frame in 10 minutes on average.

To obtain the per-pixel ground truth label images in which each plant part was represented by a unique color encoding, the scene was duplicated and the color mapping of each plant part was replaced by primary or secondary colors. The background of this duplicate scene was set to black. To avoid interaction of colors in the scene, which would result in more colors than labels, the virtual camera was set to only register a single direct ray of light without bounces. Unfortunately, rendered edges between plant parts still interpolated colors. Therefore the synthetic image labels were post-processed in the commercial image processing software package Halcon 12 (MVTech, 2016) by removing any interpolated color pixels and replacing them with the most frequent neighbour color

in a 3x3 patch If this convolution failed in case the majority of the neighbouring pixels also had been interpolated, the window was enlarged until all pixels were equal to one of the class labels. Furthermore, the colors were replaced by grayscale values in a single channel to finally reduce the label image size by 98%.

Ground truth depth images were rendered separately per frame by using the mist environment variable in Blender. For each pixel in the image, the light ray distance between the object and the camera's projection centre was obtained. Hence encoded distances

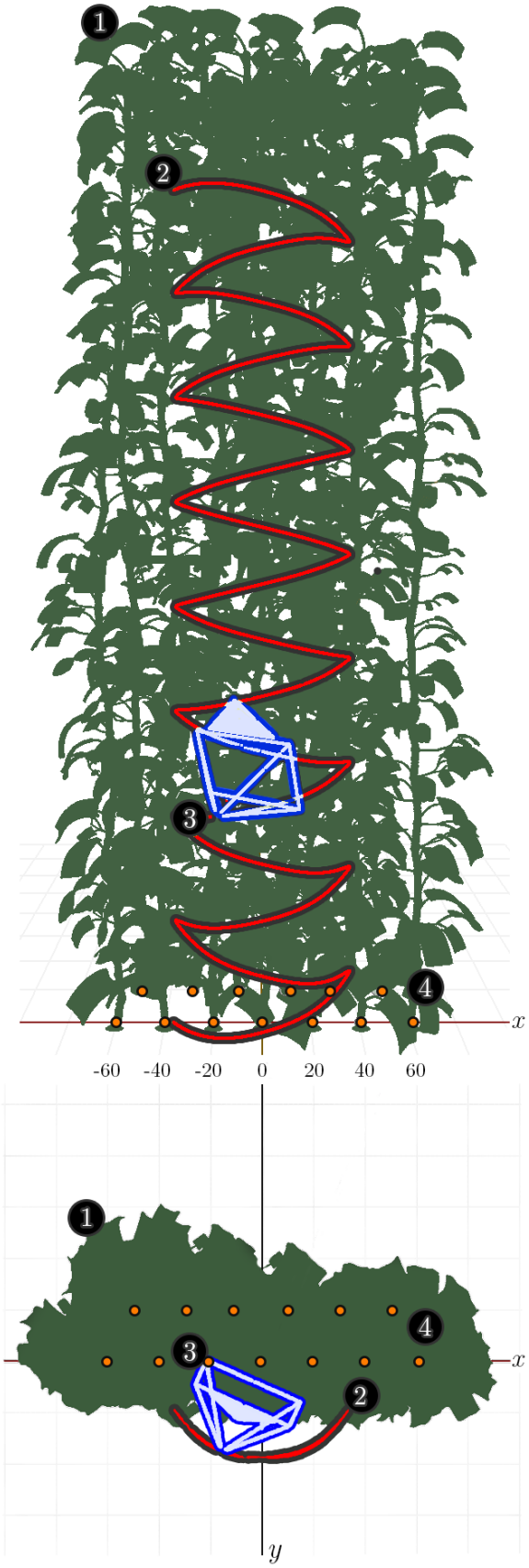


Figure 3.9: Front and top perspective views of a sweet pepper crop scene in Blender, without textures. The camera plus illumination (blue, 3) and their path (red, 2) were placed in front of 2 rows of plants (green, 1). The position of each plant on the floor plane is indicated with a dot (orange, 4). Grid spacing is 20 cm. Note that leaves are rectangular; though during render time the shape of the leaf was refined by applying an opacity map.

were not equal to real world XYZ-coordinates, which could not be obtained due to the absence of corresponding camera poses. The distance was encoded in image grayscale values, ranging from 0 to 255. In this image, the distance in centimetres for each pixel z_p could be recovered by the function $z_p = \left(\frac{255-I_p}{\left(\frac{2*150}{255}\right)}\right)$, where the intensity of a pixel I_p decreases from the maximum intensity 255 with a factor based on the range of the mist in the render, which was set at 150 centimeters. The depth images were also rendered in a colorscale with high contrast for intuitive viewing. The resolution in depth of this ground truth is coarse, though an exact representation could be obtained if needed by exporting the scene's Z-buffer in Blender to the OpenEXR (Kainz et al., 2004) linear format.

3.2.7 Annotation

The empirical image set obtained in the greenhouse contains 750 images at a range of 5 distances. From the 150 images taken at 40 cm, 50 images were annotated using Photoshop CC (Adobe, 2016) by manually outlining and coloring plant part classes. The suction cup of the robot's end-effector occluded the image and was labeled as background. Note that unlike its synthetic counter-part, ground truth in dark areas of the images were hard to manually discern and annotate. Hence only parts were annotated that could be clearly recognised. Average manual annotation time was 30 minutes per image.

3.2.8 Semantic Segmentation

We gathered evidence for our hypothesis that synthetic bootstrapping and fine-tuning with a small empirical dataset can be effective by running 5 experiments with a semantic segmentation deep learning network, using the DeepLab framework (Papandreou et al., 2015) based on Caffe (Jia et al., 2014).

Specifically we used the Deeplab VGG-16 Vanilla model (Papandreou et al., 2015) with a receptive field of 128 pixels and a stride of 8 pixels. The hyperparameters of the network were manually optimised as suggested by (Bengio, 2012) and resulted in using Adaptive Moment Estimation (ADAM) (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ and a base learning rate of 0.00005 for 30,000 iterations with a batch size of 10.

For each experiment we changed the dataset composition (synthetic or empirical images) for learning, fine-tuning or testing. The following compositions were investigated. Brackets indicate image indexes used.

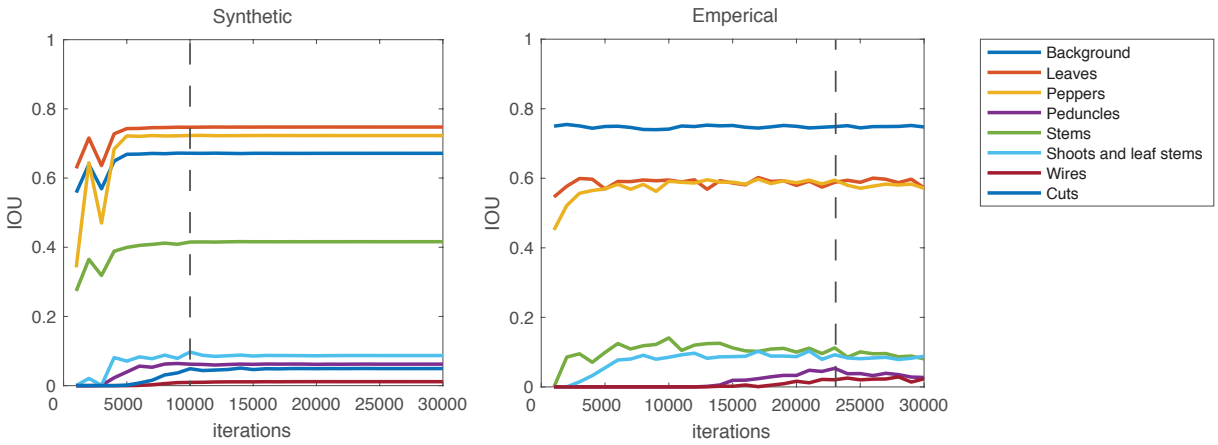


Figure 3.10: Average IOU over the validation set per class per iteration in validation set of synthetic bootstrapping (left) and empirical fine-tuning (right). Dashed vertical lines indicate at which iteration the model was fixated before training stabilized or overfitted.

A *Train: synthetic (1-8,750). Test: synthetic (8,851-8,900).*
This experiment was run to obtain a performance reference point of the model when having access to a large and detailed annotated dataset for this domain.

B *Train: synthetic (1-8,750). Test: empirical (41-50).*
To determine to what extent a synthetically trained model can generalise to a similar set in the same domain without fine-tuning.

C *Train: empirical (1-30). Test: empirical (41-50).*
As a reference to see if the model can learn using a small dataset, using empirical data.

D *Train: PASCAL VOC. Fine-tune: empirical (1-30). Test: empirical (41-50).*
To compare the effect of bootstrapping with a non-related dataset.

E *Train: synthetic (1-8,750). Fine-tune: empirical (30). Test: empirical (41-50).*
To assess the effect of bootstrapping with a related dataset.

For each experiment, overfitting was prevented by selecting the optimal model by periodically checking the model's performance on an separate validation set. For the synthetic set, these were unique images (8,751-8,800) from the 6th scene. For the empirical set, these were images of unique plants (31-40).

Performance evaluation

To calculate the performance of our method and to enable equal comparison of future methods, we used the Jaccard Index similarity coefficient as an evaluation measure. This index is also known as the intersection-over-union (IOU) (He & Garcia, 2009) and is widely used for semantic segmentation evaluation (Everingham et al., 2010). The measure is defined in Equation 6.2, where the mean IOU per class equals the intersection of the semantic segmentation and the ground truth divided by their union. To derive the measure, a pixel-level confusion matrix C is calculated first for each image I in dataset D :

$$C_{ij} = \sum_{I \in D} \left| \left\{ p \in I \mid S^I_{gt}(p) = i \wedge S^I_{ps}(p) = j \right\} \right|, \tag{3.1}$$

where $S^I_{gt}(p)$ is the ground truth label of pixel p in image I and $S^I_{ps}(p)$ is the predicted label. This implies that C_{ij} equal the number of predicted pixels i with label j. The Jaccard Index can then be derived as an average for each class L by:

$$Jaccard\,Index \; = \frac{1}{L} \sum_{i=1}^{L} \frac{C_{ii}}{G_i + P_i - C_{ii}} \;, \text{ where} \tag{3.2}$$

$$G_i = \sum_{j=1}^{L} C_{ij} \quad \text{and} \quad P_j = \sum_{i} C_{ij} \tag{3.3}$$

Hence G_i denotes the total number of pixels labeled with class i in the ground truth and P_j the total number of pixels with prediction j in the image.

3.3 Results

A synthetic image dataset of 10,500 images was generated using 6 unique scenes and an empirical dataset of 750 images was obtained. A pixel-level ground truth segmentation of 8 classes was created automatically for all images in the synthetic dataset and manually for 50 images in the empirical dataset. This section first provides example images of both sets after which the sets will be compared on differences in color, class and spatial distribution to verify to what extent the requirement of similarity was met for our first hypothesis. To answer the second hypothesis that such datasets for agriculture are a valid and valuable tool for computer vision learning methods, results of the 5 experiments will be presented.

3.3.1 Datasets Description

In Figure 3.11, examples of real and synthetic images are shown with their corresponding ground truths. The datasets and the source material can be found at:
`http://dx.doi.org/10.4121/uuid:884958f5-b868-46e1-b3d8-a0b5d91b02c0`

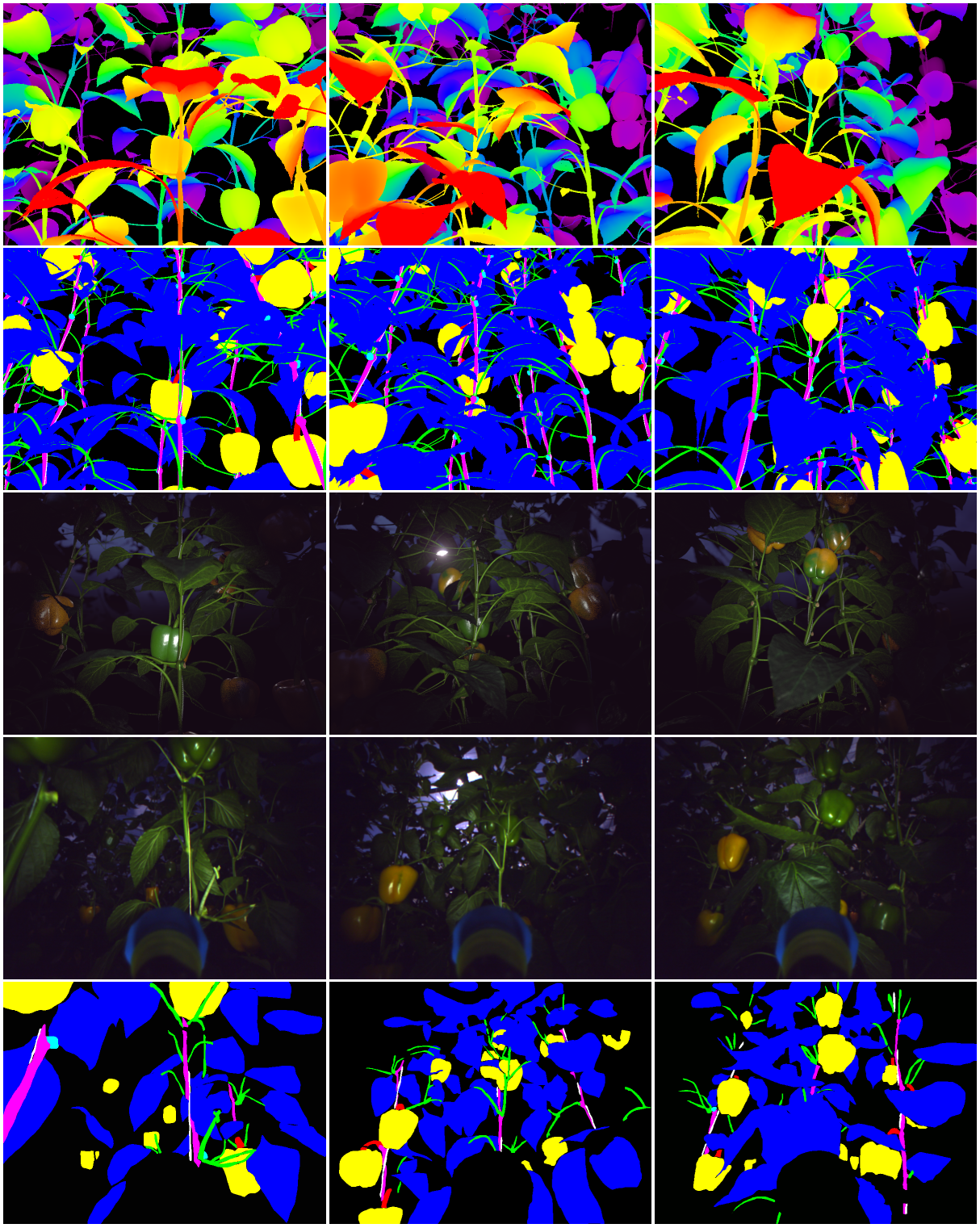
Datasets comparison

Results of the comparisons between the synthetic and empirical datasets are presented in this section. First, pixel frequencies of classes and their spatial distributions between both sets were compared because such distributions reflect if the structure of the object we intended to model was similar. In Figure 3.12 the pixel class frequencies are shown for both datasets. To investigate their spatial distribution, the normalised per class pixel label distributions are shown in Figure 3.13.

Property distributions within classes themselves was another comparative perspective, for example color distributions. Some computer vision and learning methods are sensitive to object color, affecting the generalisation of the method to new images with different color distributions. In Figure 3.14 the color spectrum for each plant part in both sets are shown.

The spectra were obtained by transforming the color images to hue, saturation and value (HSV) colorspace. The hue channel in this image represented for each pixel which color on the visible spectrum was present, irregardless of illumination and saturation intensity. Due to the heterogeneous illumination distribution in the images, the dark edges of the

Figure 3.11: Three examples of the synthetic and empirical color images and their corresponding ground truth labels. The first three rows contain column pairs of i) synthetic ground truth depth labels, ii) class labels and iii) color images. The last two rows contain column pairs of i) real images and ii) ground truth class labels. Note the depth label has an arbitrary colorscale for intuitive viewing. Class labels: ● background, ● leaves, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts. See Appendix A for enlarged pairs of synthetic and real color images



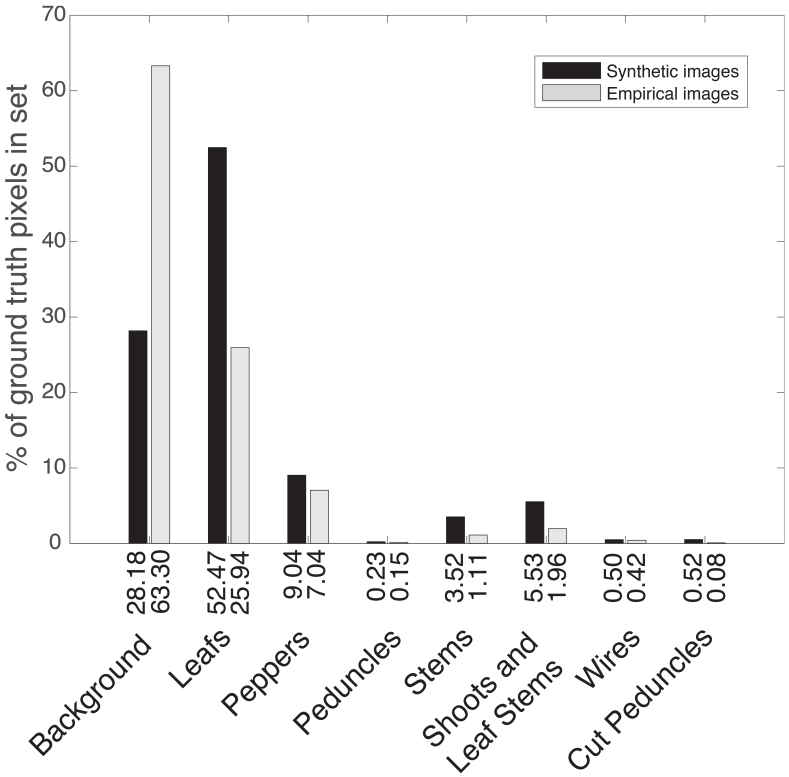


Figure 3.12: Percentage of ground truth pixels per class, compared between real and synthetic datasets.

image were overrepresented with colors in the end of the spectrum. For this reason, we focussed the color analysis on a 300x300 pixel patch in the well illuminated middle of the image.

Intensity was another dimension of interest. With an average intensity of 37 and standard deviation of 12, comparison of differences between plant parts was coarse. Instead we investigated the average spatial intensity distribution over all images in each set as shown in Figure 3.15. This enabled us to verify the similarity of the simulated illumination heterogeneity with the empirical set.

3.3.2 Semantic segmentation results

IOU results for experiment A through E are shown in Figure 3.16, separated by class. Segmentation results for the best performing network on synthetic data (A) and empirical data (E) are shown in Figure 3.17.

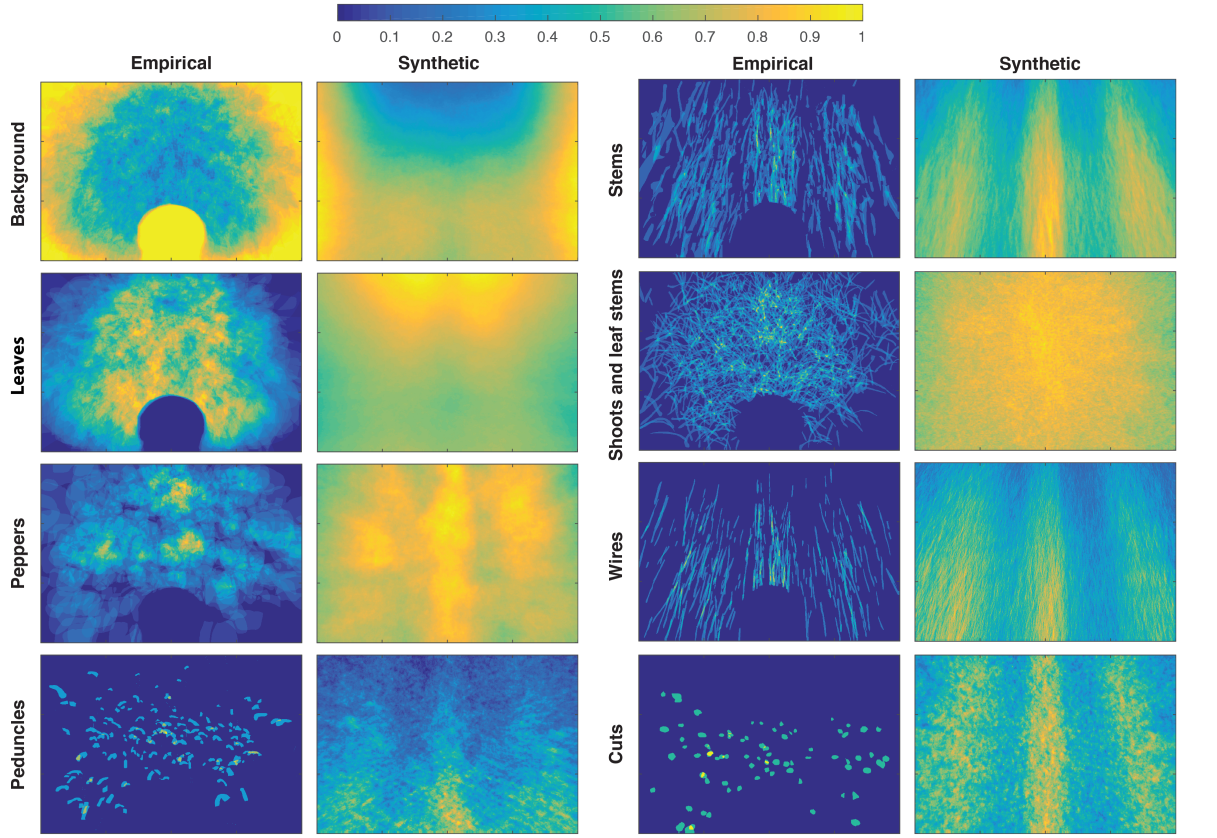


Figure 3.13: Per class normalized pixel label distributions for the emperical and synthetic dataset. Each image was obtained by summation of class masks of all images in the set, divided by the maximum resulting pixel value. Note that the suction cup of the end-effector (in the middle at the bottom) was labeled as background in the empirical dataset.

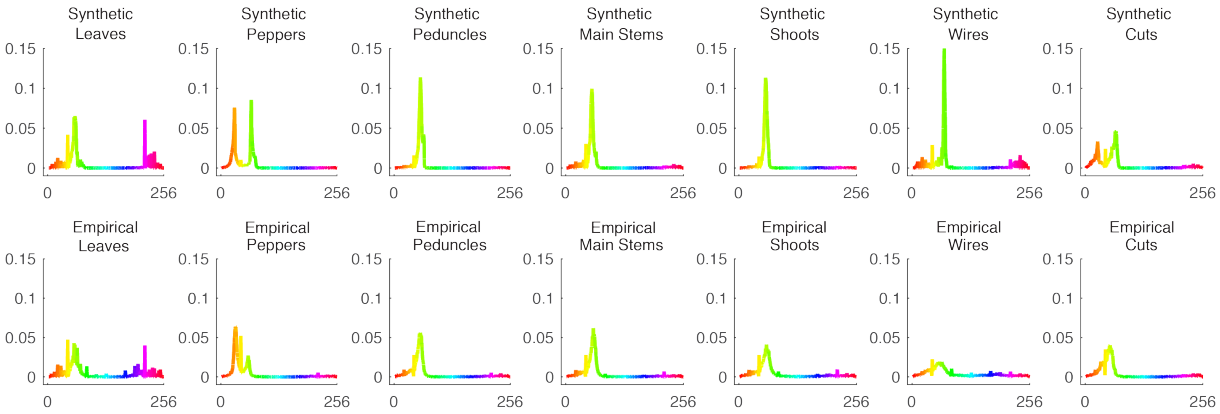


Figure 3.14: Color distribution per plant part of synthetic and empirical images. The vertical axis shows the percentage of plant part color averaged over the image set. The horizontal axis shows the hue value in HSV colorspace.

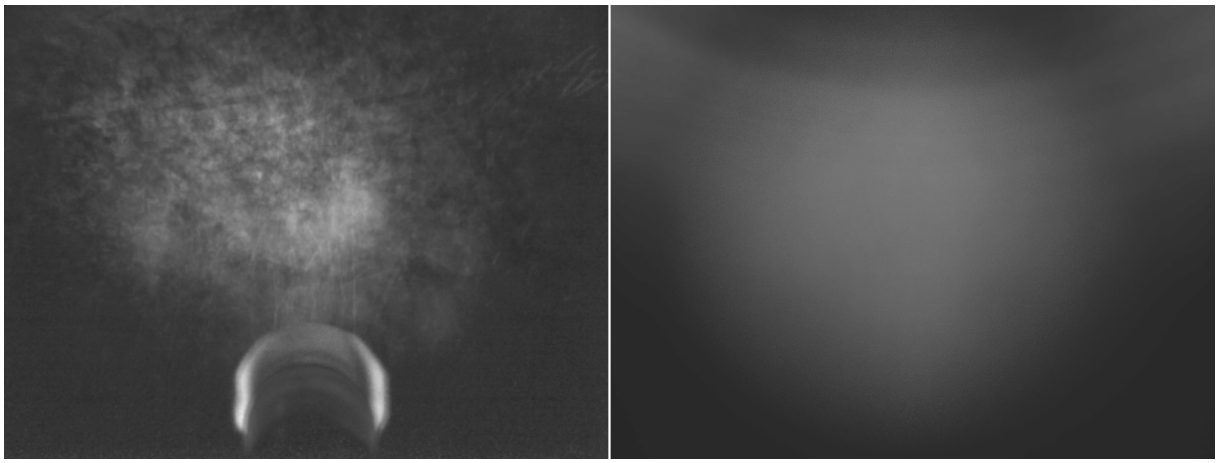


Figure 3.15: Average illumination intensity distribution over all images in the emprical (left) and the synthetic (right) sets, with an average pixel intensity of 37 and 38 respectively. In the images shown here, the intensity of both images was doubled to increase contrast for the reader.

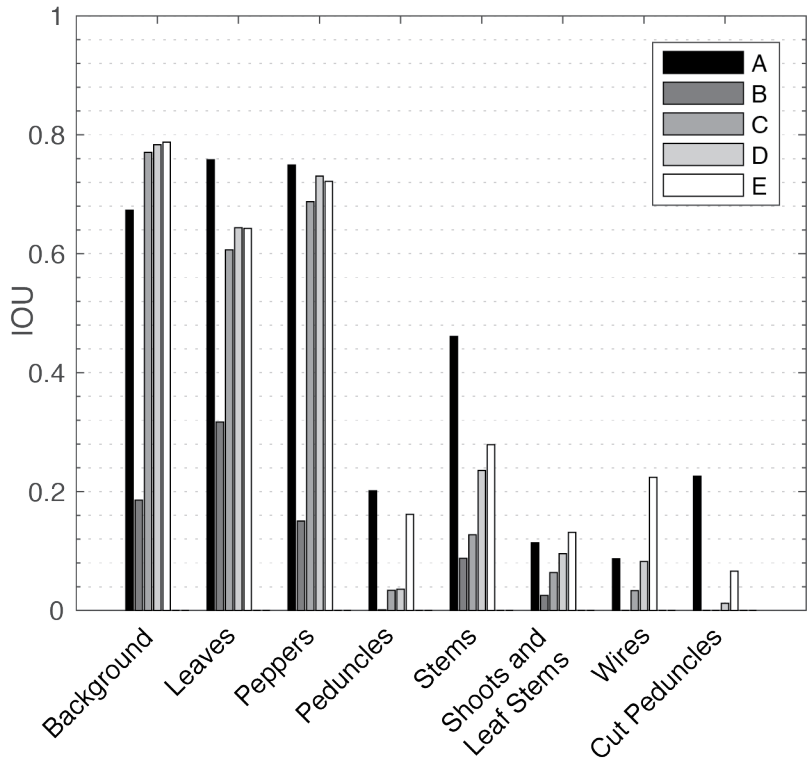


Figure 3.16: Average IOU per class over the test set for experiments A through E.

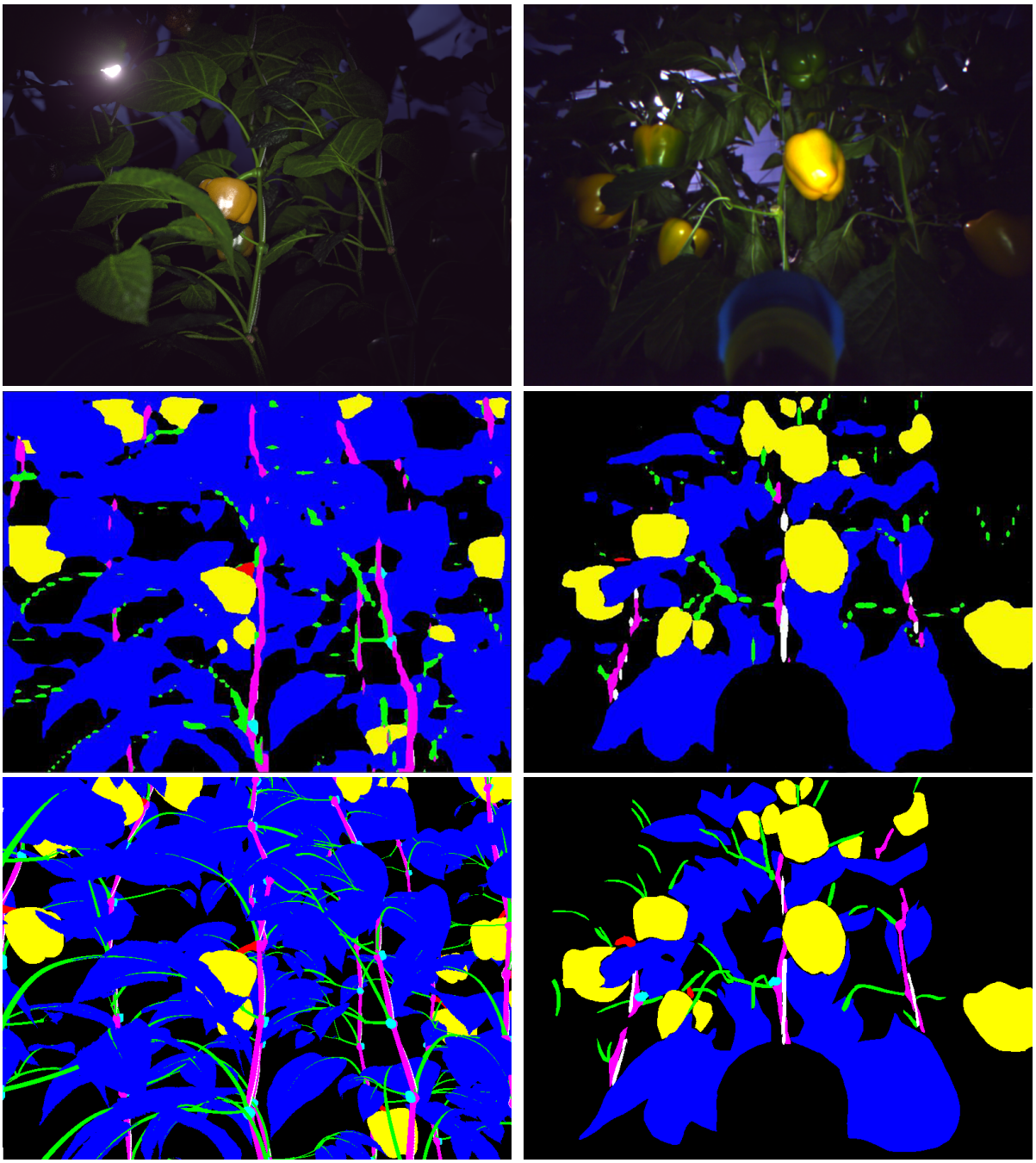


Figure 3.17: Segmentation results for synthetic test set from experiment A (left column) and empirical test set from experiment E (right column). Color images (top), classification segmentation (middle) and ground truth (bottom) are shown in each row. Class labels: ● background, ● leaves, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts.

3.4 Discussion

In this chapter we aimed to help the computer vision performance in agriculture towards a state-of-the-art level required for the next generation robotics, e.g. for harvesting, disease detection and phenotyping. Our conjecture of the cause of the low performance in this domain was the unavailability of detailed large annotated label data sets, necessary for most novel machine learning approaches. Recently, generating and training on synthetic image datasets has proven to be a popular and effective solution in other domains.

To extend such an approach to the domain of agriculture, we have described a novel methodology to generate synthetic images of plants. Although the modelling can be time consuming itself, it facilitates the generation of large-scale and more detailed datasets under a broader set of conditions, e.g. different illumination conditions, perspectives or sensors. This would otherwise not be feasible to obtain due to the required large annotation effort. Our approach is generic and applicable to any crop with a consistent modular plant architecture after obtaining an accurate and exhaustive definition of the corresponding plant parameters (Vos et al., 2010, 2007).

Our dataset is an important contribution for the availability of a variety of datasets in the computer vision community so that methods can verify their robustness and generalisation. Currently the focus in the research community is on tuning and validation using type restricted datasets, e.g. human or urban scenes. Our datasets provide a use-case for detailed hierarchical part recognition in bio-related fields, as needed in the context of robotics.

In this chapter we presented a modelling example of a single point in time for a single variety of *Capsicum annuum*. However, our methodology for generating plant models is extendable to include plant parameters for plants under multiple stages of growth. The pipeline allows to interpolate between seasonal plant parameters to generate plants in different growth stages. However, this would require more empirical measurements and there is a trade-off between modelling accuracy over the season and the accuracy that is desired from the application. For our purpose of using machine learning, we assumed a single point in time for modelling would already be sufficient to improve synthetic based learning.

As we did not have depth data in our empirical dataset, it was not possible to test any hypothesis regarding machine learning that made use of the synthetic depth data. We added the methodology for generating depth data for future research and as an example for the community how to obtain such data.

In the following subsections we discuss our hypotheses i) that we can create synthetic images similar to empirical images by discussing to what extend our requirement of similarity was met and ii) that a synthetically bootstrapped model can be used for improved learning when only fine-tuned on a small annotated empirical dataset.

3.4.1 Synthetic and Empirical Set Similiarity

At the beginning of this chapter we posed that the synthetic images should be similar to the empirical images. We can both qualitatively and qualitatively observe the differences between sets. The former is subjective and it must be noted that human perceptual evaluation of images often employs sensory completion to make up for differences or absences (McNamara, 2001). Nonetheless it is valuable to compare the sets in this regard because it provides clues for objective comparison and possible improvements.

Qualitative Comparison

When visually compared we noted slight differences in color. This can be explained by the manual color tuning process in which it was hard to find colors close to the empirical situation, given that only the result of the illumination interaction effects of the materials, the light source and the environment could be observed. Future research should include methods for automated color optimisation. Also obtaining a calibrated color ground truth is recommended in combination with proper camera color calibration.

Another difference was the greater perceived variance of shapes and poses in the empirical set. This was the result of excluding part shape variance, e.g. not taking into account an exhaustive set of leaf curls and poses or local side-shoot deformations. Hence, the plant parameters were not adequately capturing all the variation. In forthcoming research we suggest to include also intra plant part pose variations and deformations.

Quantitative Comparison

Superordinate image similarity measures were previously not well defined in literature. However, studies of individual measures like color histogram comparison (Swain & Ballard, 1991) or shape measures (Mehtre et al., 1997) have been performed. In this chapter, the similarity requirement was also not quantified in a single measure. Instead we looked at: i) label set distributions, ii) label image position distributions, iii) part color spectra and iv) illumination intensity distributions to gain a more quantitative insight into the similarity between sets.

i Within a dataset, label frequencies are often highly unbalanced (Caesar et al., 2015), resulting in neglected classes in some type of computer vision approaches. To counter that effect, object occurrence statistics can be used for normalisation (Chawla, 2010). In Figure 3.12) we observe an unbalance within and between the sets. The latter can partially be explained by the methodological difference in obtaining the ground truth in both cases. For the synthetic dataset, the labels of all pixels in the color image were computable irregardless of illumination. For the empirical dataset, a subjective decision on the label in the dark edge areas was often not possible.

ii In Figure 3.13 we compared the plant part spatial distribution in the images between sets. Overall, the sparsity in the empirical set is notable and was due to the small size of the annotated set. In the synthetic set, we note 3 vertical hotspots of stem+wire classes whereas in the empirical set the spatial variance of these classes was higher. This can be explained by a more regular plant distance in the synthetic set, due to the fact that our model did not take intra-plant propeties into account.

The distributions of the peppers between sets corresponds, e.g. a hot spot in the center of both distributions and less to the left and right explained by the particular occlusions resulting from the chosen scanning path. Closely correlated with peppers are the peduncles, however the peduncle distributions were hard to compare due to their sparsity.

Lastly, there was a difference between leaf class distributions. Whilst empirically more centered, in the synthetic set there was a higher occurrence at the top of the image. This can be explained by improper modelling the shapes and poses of the leaves, resulting in a discrepancy of silhouettes when viewed from a 20 degree upward angle.

iii Evaluating Figure 3.14, we observe plant part color similarities but also differences in our sets.

Leaves: The proportion of the left spectrum of the leaf class was similar, though the synthetic leaves should contain relatively less greens. The purple spectrum on the right of this class can be explained by the outward position of some leaves, leading to underexposure and therefore taking on the purple global background illumination color.

Peppers: The pepper colors seem far off at first, though the green peak height might be caused by the improper modelling of the percentage of green (50%) and yellow peppers (50%). In the empirical set, more ripe fruit were present because a part of the crop was selected in the greenhouse with abundance of yellow peppers to increase the spatial density of robot harvest trials. Irregardless of this ripeness imbalance, the color of the ripe peppers in the synthetic set needs to be adjusted towards the yellow end of the spectrum.

Peduncles, stems and shoots: All were modelled too green and lack yellows.

Wires: The synthetic wire color distribution shows a peak in green. By inspecting the data this was likely the result by the rendering a color interpolation at the edges of the wire class and background stem class. Furthermore, a relative large part of the wire pixels were edge pixels and these pixels were included in the ground truth of the wire class. A thicker synthetic wire might increase color similarity.

Cuts: Although the cuts were textured with a photograph, the absence of color calibration most likely resulted in the difference we observe in this class.

iv When looking at the average illumination intensity distribution over all images in each set (Figure 3.15) we observe a comparable heterogenous illumination distribution with a strong vignetting effect. This was caused by i) an interaction of illumination hardware that focussed a centered beam on the scene and ii) the lens type which had a default vignet. The average intensity of these images was similar.

For aspects discussed under both i and ii, these differences might also be caused due to the empirical set measurement only consisting from frontal, -45 and +45 degree views, whereas the synthetic set included also intermediate waypoints that furthermore moved vertically per next frame, as depicted in Figure 3.9.

Although the differences discussed in i-iv could be valuable for improving the similarity between image sets, their combined impact can only be evaluated in the perspective of a specific task performance, e.g. machine learning. In the context of computer vision, we can state that image sets are sufficiently similar when they in any manner can be used for

improved recognition. Therefore we evaluated the similarity also in the context of the task of segmentation in Section 3.4.2, where we quantitatively compare the IOU performance of the 5 experiments.

3.4.2 Bootstrapping and Segmentation

Although differences between the image sets exist, experiments A through E indicated that bootstrapping with related synthetic data improved the learning performance compared to solely training on a small empirical dataset or bootstrapping with non-related data. From each experiment, we concluded the following:

A The model trained and tested using the synthetic dataset provided a baseline performance on the task when having access to a large amount of detailed annotated data.

B Without any fine-tuning, the synthetically bootstrapped model did not generalise well to empirical data.

C Using only a small empirical dataset for training, the model learned to differentiate plant parts to a certain extent. However, this primarily holds for the classes background, leaves and peppers, with an average IOU of 0.68. We observe that the model learns the most frequent classes that were also most discriminative in color, e.g. black, dark green and yellow correspondingly. The other classes that were infrequent and overlapped in color with the frequent classes were segmented poorly, with an average IOU of 0.05.

D When bootstrapping with an non-related dataset (PASCAL VOC) and fine-tuning with empirical data, performance was increased over the previous experiments (B,C) testing on empirical images.

E When bootstrapping with our synthetic related dataset and fine-tuning with empirical data, the best performance was achieved testing on empirical images.

From the results we reckon that the increase in performance using synthetic data bootstrapping compared to the other approaches might be caused by the increased training sample number with high similarity that was made available to the CNN.

In the experiments we observed a correlation between class frequency and class performance, suggesting the model had a bias for class availability. Future efforts should be focussed on coping with this bias, for example using normalisation during the computation of the loss.

Our experiments and their conclusions are indicative because we could not prove there does not exist a different convolutional model architecture, hyper-parameter combination or initialisation per experiment that would have a better performance. However, the results present a starting point to determine how synthetic data can be used to improve segmentation performance.

Although plant part segmentation in reconstructed 3D models has previously been achieved in smaller plants (Golbach et al., 2016; Paproki et al., 2011), segmenting multiple plant parts from single 2D images previously remained unsolved. For plant robotics and phenotyping, the requirement of plant part localisation is currently a bottleneck (Minervini et al., 2015). Our results show a promising method for meeting this requirement when observing the final segmentations qualitatively.

Our work contributes to image segmentation challenges in the plant domain. In agricultural applications, our approach of segmenting individual plant parts in high detail will enable a large range of possibilities. For example, from leaf volume estimations in vineyards to a all kinds of phenotyping applications to determine plant parameters from images.

3.5 Conclusion

A new methodology for generating synthetic data sets for agricultural computer vision was presented. Based on empirical data, a sweet pepper plant model was created, randomised plant instances were generated and rendered to mimic realistic greenhouse conditions. Our hypothesis that with this approach we can create a synthetic image dataset similar to empirical images holds perceptually and qualitatively, though quantitatively there were differences in class and color distributions. However, we also stated that the requirement of similarity depends on the task, e.g. pre-training models for image segmentation. Our hypothesis that bootstrapping a convolutional neural network that fine-tunes on a small empirical dataset outperforms other methods of training has been confirmed by our experiments. Segmentation results show a promising next step for semantic part localisation in agriculture. Future efforts should be aimed in further optimising the network architectures, focussing on the performance of the infrequent classes. The datasets and their source material are publicly released and can be found at:
`http://dx.doi.org/10.4121/uuid:884958f5-b868-46e1-b3d8-a0b5d91b02c0`

3.6 Acknowledgements

This research was partially funded by the European Commission in the Horizon2020 Programme (SWEEPER GA no. 644313). The authors would like to thank prof.dr. R. D. Howe and dr. D. Perrin for their input of this research and making computing resources available.

References

Adobe (2016). Photoshop. url: `http://www.adobe.com/products/photoshop.html`.

Bac, C., Hemming, J., & van Henten, E. (2013). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture*, *96*, 148 – 162. doi: http://dx.doi.org/10.1016/j.compag.2013.05.004.

Bac, C., Hemming, J., & van Henten, E. (2014a). Stem localization of sweet-pepper plants using the support wire as a visual cue. *Computers and Electronics in Agriculture*, *105*, 111 – 120. doi: http://dx.doi.org/10.1016/j.compag.2014.04.011.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014b). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Bac, C. W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & van Henten, E. J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, *146*, 85 – 97. doi: http://dx.doi.org/10.1016/j.biosystemseng.2015.07.004. Special Issue: Advances in Robotic Agriculture for Crops.

Ballina-Gomez, H., Latournerie-Moreno, L., Ruiz-Sanchez, E., Perez-Gutierrez, A., & Rosado-Lugo, G. (2013). Morphological characterization of Capsicum annuum L. accessions from southern Mexico and their response to the Bemisia tabaci-Begomovirus complex. *Chilean journal of agricultural research*, *73*, 329 – 338.

Barth, R., Hemming, J., & van Henten, E. J. (2016). Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, *146*, 71 – 84. doi: http://dx.doi.org/10.1016/j.biosystemseng.2015.12.001. Special Issue: Advances in Robotic Agriculture for Crops.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR*, *abs/1206.5533*.

Benoit, L., Rousseau, D., Étienne Belin, Demilly, D., & Chapeau-Blondeau, F. (2014). Simulation of image acquisition in machine vision dedicated to seedling elongation to validate image processing root segmentation algorithms. *Computers and Electronics in Agriculture*, *104*, 84 – 92. doi: https://doi.org/10.1016/j.compag.2014.04.001.

Blender Foundation (2016). Blender. url: `https://www.blender.org`.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, *34*, 483–519. doi: 10.1007/s10115-012-0487-8.

Caesar, H., Uijlings, J. R. R., & Ferrari, V. (2015). Joint calibration for semantic segmentation. *CoRR*, *abs/1507.01581*.

Chawla, N. V. (2010). Data mining for imbalanced datasets: An overview. In O. Maimon, & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 875–886). Boston, MA: Springer US. doi: 10.1007/978-0-387-09823-4_45.

Cicco, M. D., Potena, C., Grisetti, G., & Pretto, A. (2016). Automatic model based dataset generation for fast and accurate crop and weeds detection. *CoRR*, *abs/1612.03019*.

Cordier, N., Delingette, H., Le, M., & Ayache, N. (2016). Extended modality propagation: Image synthesis of pathological cases. *IEEE Transactions on Medical Imaging*, *PP*, 1–1. doi: 10.1109/TMI.2016.2589760.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., aurelio Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., & Ng, A. Y. (2012). Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1223–1231). Curran Associates, Inc.

Dittrich, F., Woern, H., Sharma, V., & Yayilgan, S. (2014). Pixelwise object class segmentation based on synthetic data using an optimized training strategy. In *Networks Soft Computing (ICNSC), 2014 First International Conference on* (pp. 388–394). doi: 10.1109/CNSC.2014.6906671.

EON Software (2016). Plant factory. url: `http://www.plantfactory-tech.com/`.

Everingham, M., Eslami, S. M., Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vision*, *111*, 98–136. doi: 10.1007/s11263-014-0733-5.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Fergus, R., Fei-Fei, L., Perona, P., & Zisserman, A. (2005). Learning object categories from google's image search. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (pp. 1816–1823 Vol. 2). volume 2. doi: 10.1109/ICCV.2005.142.

Godin, C. (2000). Representing and encoding plant architecture: A review. *Ann. For. Sci.*, *57*, 413–438. doi: 10.1051/forest:2000132.

Golbach, F., Kootstra, G., Damjanovic, S., Otten, G., & van de Zedde, R. (2016). Validation of plant part measurements using a 3d reconstruction method suitable for high-throughput seedling phenotyping. *Machine Vision and Applications*, *27*, 663–680. doi: 10.1007/s00138-015-0727-5.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, *116*, 8 – 19.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 1263–1284. doi: 10.1109/TKDE.2008.239.

van der Heijden, G., Song, Y., Horgan, G., Polder, G., Dieleman, A., Bink, M., Palloix, A., van Eeuwijk, F., & Glasbey, C. (2012). Spicy: towards automated phenotyping of large pepper plants in the greenhouse. *Functional Plant Biology*, *39*, 870–877.

Hemming, J., Ruizendaal, J., Hofstee, J. W., & van Henten, E. J. (2014). Fruit detectability analysis for different camera positions in sweet-pepper. *Sensors (Basel)*, *14*, 6032–6044. doi: 10.3390/s140406032. 24681670[pmid].

(IPGRI), I. P. G. R. I. (1995). *Descriptors for Capsicum (Capsicum spp.)*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, .

Kainz, F., Bogart, R., & Hess, D. (2004). The openexr image file format. *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics, R. Fernando, Ed. Pearson Higher Education*, .

Kent, B. R. (2015). *3D Scientific Visualization with Blender*. 2053-2571. Morgan and Claypool Publishers. doi: 10.1088/978-1-6270-5612-0.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Kondaveeti, H. K. (2016). Synthetic isar images of aircrafts. doi: 10.5281/zenodo.48002.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. Insight.

McNamara, A. (2001). Visual perception in realistic image synthesis. *Computer Graphics Forum*, *20*, 211–224. doi: 10.1111/1467-8659.00550.

Mehtre, B. M., Kankanhalli, M. S., & Lee, W. F. (1997). Shape measures for content based image retrieval: A comparison. *Information Processing & Management*, *33*, 319 – 337. doi: http://dx.doi.org/10.1016/S0306-4573(96)00069-6.

Minervini, M., Scharr, H., & Tsaftaris, S. A. (2015). Image analysis: The new bottleneck in plant phenotyping [applications corner]. *IEEE Signal Processing Magazine*, *32*, 126–131. doi: 10.1109/MSP.2015.2405111.

MVTech (2016). Halcon. url: `http://www.halcon.com/`.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*, 1–21. doi: 10.1186/s40537-014-0007-7.

Nasir, A., Rahman, M., & Mamat, A. (2012). A study of image processing in agriculture application under high performance computing environment. *International Journal of Computer Science and Telecommunications*, *3*.

Papandreou, G., Chen, L.-C., Murphy, K., & Yuille, A. L. (2015). Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*.

Paproki, A., Fripp, J., Salvado, O., Sirault, X., Berry, S., & Furbank, R. (2011). Automated 3d segmentation and analysis of cotton plants. In *2011 International Conference on Digital Image Computing: Techniques and Applications* (pp. 555–560). doi: 10.1109/DICTA.2011.99.

Polder, G., van der Heijden, G. W., van Doorn, J., & Baltissen, T. A. (2014). Automatic detection of tulip breaking virus (tbv) in tulip fields using machine vision. *Biosystems Engineering*, *117*, 35 – 42. doi: http://dx.doi.org/10.1016/j.biosystemseng.2013.05.010. Image Analysis in Agriculture.

Pradal, C., Artzet, S., Chopard, J., Dupuis, D., Fournier, C., Mielewczik, M., Nègre, V., Neveu, P., Parigot, D., Valduriez, P., & Cohen-Boulakia, S. (2017). Infraphenogrid: A scientific workflow infrastructure for plant phenomics on the grid. *Future Generation Computer Systems*, *67*, 341 – 353. doi: https://doi.org/10.1016/j.future.2016.06.002.

Pradal, C., Fournier, C., Valduriez, P., & Cohen-Boulakia, S. (2015). OpenAlea: Scientific Workflows Combining Data Analysis and Simulation. In *SSDBM 2015: 27th International Conference on Scientific and Statistical Database Management*. San Diego, United States. doi: 10.1145/2791347.2791365.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3234–3243). doi: 10.1109/CVPR.2016.352.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*, 157–173. doi: 10.1007/s11263-007-0090-8.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, *16*, 1222. doi: 10.3390/s16081222.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., & Blake, A. (2013). Efficient human pose estimation from single depth images. In A. Criminisi, & J. Shotton (Eds.), *Decision Forests for Computer Vision and Medical Image Analysis* (pp. 175–192). London: Springer London. doi: 10.1007/978-1-4471-4929-3_13.

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*, 11–32. doi: 10.1007/BF00130487.

Trask, A., Gilmore, D., & Russell, M. (2015). Modeling order in neural word embeddings at scale. *CoRR*, *abs/1506.02338*.

Vos, J., Evers, J. B., Buck-Sorlin, G. H., Andrieu, B., Chelle, M., & de Visser, P. H. B. (2010). Functional–structural plant modelling: a new versatile tool in crop science. *Journal of Experimental Botany*, *61*, 2101–2115. doi: 10.1093/jxb/erp345. `arXiv:http://jxb.oxfordjournals.org/content/61/8/2101.full.pdf+html`.

Vos, J., Marcelis, L. F. M., Visser, P., Struik, P. C., & Evers, J. (2007). *Functional-Structural Plant Modelling in Crop Production*. Springer Netherlands.



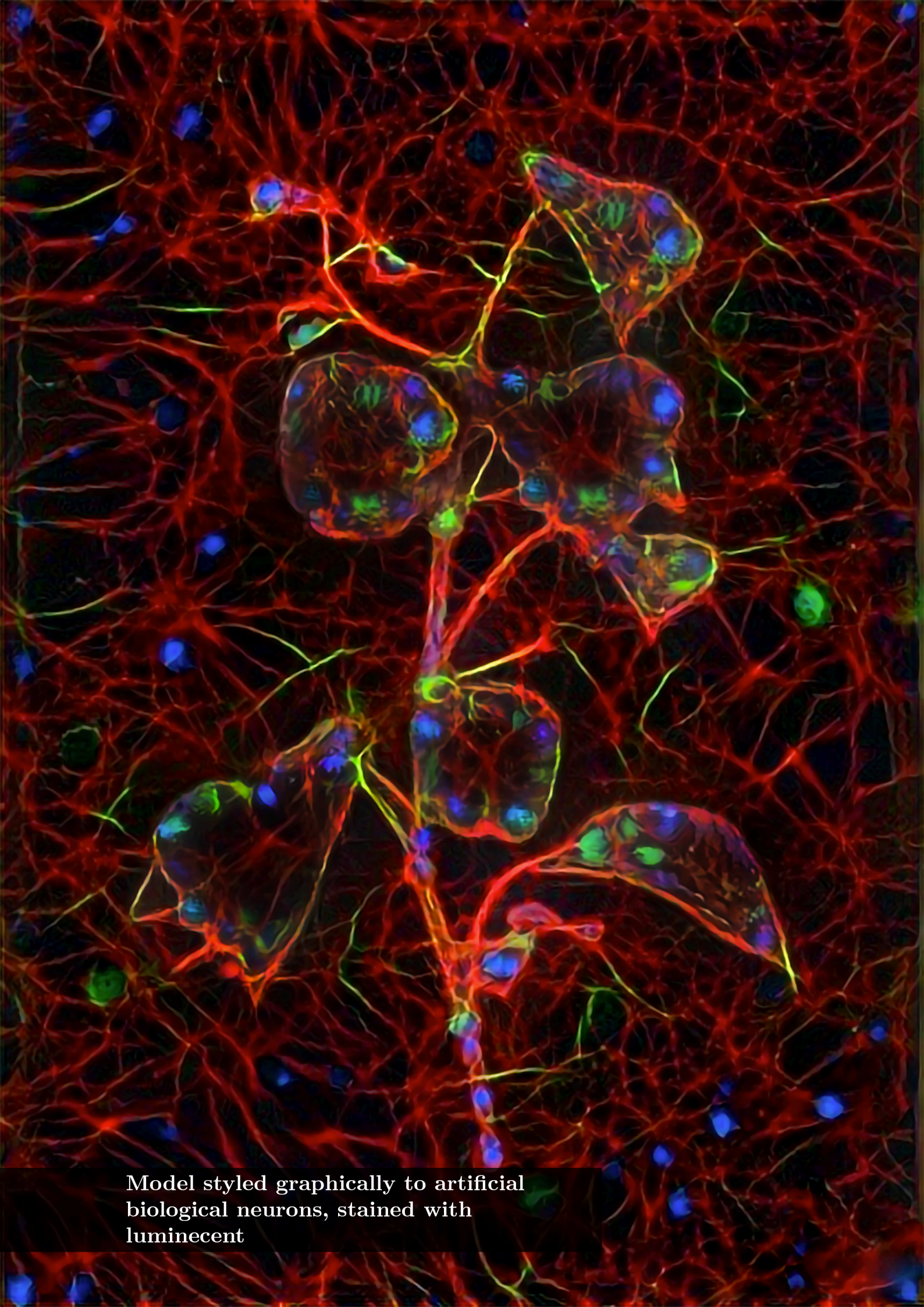
Figure 3.18: Example of synthetic color image (top) and empirical color image (bottom).



Figure 3.19: Example of synthetic color image (top) and empirical color image (bottom).



Figure 3.20: Example of synthetic color image (top) and empirical color image (bottom).



Model styled graphically to artificial biological neurons, stained with luminecent

Chapter 4

Synthetic Bootstrapping of Convolutional Neural Networks for Semantic Plant Part Segmentation.

This chapter is based on:

Barth, R., IJsselmuiden, J., Hemming, J., and van Henten, E.J. van (2017). Synthetic bootstrap- ping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture*.

Abstract

A current bottleneck of state-of-the-art machine learning methods for image segmentation in agriculture, e.g. convolutional neural networks (CNN), is the requirement of large manually annotated datasets on a per-pixel level. In this chapter, we investigated how related synthetic images can be used to bootstrap CNNs for successful learning as compared to other learning strategies. We hypothesise that a small manually annotated empirical dataset is sufficient for fine-tuning a synthetically bootstrapped CNN. Furthermore we investigated i) multiple deep learning architectures, ii) the correlation between synthetic and empirical dataset size on part segmentation performance, iii) the effect of post-processing using conditional random fields (CRF) and iv) the generalisation performance on other related datasets. For this we have performed 7 experiments using the *Capsicum annuum* (bell or sweet pepper) dataset containing 50 empirical and 10,500 synthetic images with 7 pixel-level annotated part classes. Results confirmed our hypothesis that only 30 empirical images were required to obtain the highest performance on all 7 classes (mean IOU=0.40) when a CNN was bootstrapped on related synthetic data. Furthermore we found optimal empirical performance when a VGG-16 network was modified to include *à trous* spatial pyramid pooling. Adding CRF only improved performance on the synthetic data. Training binary classifiers did not improve results. We have found a positive correlation between dataset size and performance. For the synthetic dataset, learning stabilises around 3,000 images. Generalisation to other related datasets proved possible.

4.1 Introduction

4.1.1 Research Aim

In this chapter we investigated a methodology to reduce the dependancy on manually annotated datasets for plant part segmentation in agriculture when applying state-of-the-art deep learning methods, e.g. convolutional neural networks (CNN) for semantic segmentation. CNNs were bootstrapped by a synthetic dataset and fine-tuned on a small manually annotated dataset. Additionally, our aim was to further specify CNN and data requirements for this task and therefore we investigated i) the correlation between synthetic dataset size and performance, ii) the minimum required amount of fine-tuning data, iii) explicit improvements for this task by training part classes separately in binary classifiers, iv) the effect of post-processing using conditional random fields (CRFs) and vii) the generalisation power to related datasets differing in acquisition distance and hardware.

Currently state-of-the-art computer vision methods for semantic segmentation are dominated by supervised machine learning such as CNNs (Everingham et al., 2015; Zhao et al., 2016; Wu et al., 2016). With the advent of these methods comes the requirement of large and detailed annotated datasets (Najafabadi et al., 2015). Although this depends on the model's number of free parameters and the problem complexity, it has already been shown that dataset size and classification performance are positively correlated (Banko & Brill, 2001; Brants et al., 2007). Specifically for deep learning methods, this correlation holds given sufficient model size, training iterations and regularisation (Erhan et al., 2010; Srivastava et al., 2014).

Unfortunately, the lack of annotations frequently imposes a new bottleneck for learning. Annotating per pixel class labels is labour intensive, which can become infeasible for large sets with a multitude of classes. Particularly computer vision in domains like agriculture, with a high amount of occlusions and high object and environmental complexity (Gongal et al., 2015), obtaining detailed annotated datasets that capture all image variance often proved to be insurmountable. One solution is to perform large-scale crowd-sourcing (Kavasidis et al., 2014; Everingham et al., 2010a; Russell et al., 2008; Everingham et al., 2010a), though this can remain costly. Another popular and successful solution is to bootstrap (or pre-train) machine learning models with synthetic data (Ros et al., 2016; Shotton et al., 2013; Cordier et al., 2016).

To investigate this solution and its boundary conditions in the domain agricultural computer vision, we report on deep learning experiments that used the *Capsicum annuum* (bell or sweet pepper) dataset (Barth et al., 2018) containing 8 classes (7 plant parts plus background class) annotated both in empirical images (50) and corresponding synthetic images (10,500). The dataset provides a demarcated scope for semantic part segmentation, comparable to datasets like The Penn-Fudan face part dataset (7 classes) (Wang et al., 2007), Labeled Faces in the Wild (3 classes) (Learned-Miller et al., 2016), Caltech-UCSD Birds-200-2011 bird part dataset (3 classes) (Wah et al., 2011) and the Pascal-Person-Part dataset (6 classes) (Chen et al., 2016a, 2014).

The need for object and part recognition on a per-pixel level in this domain arises from requirements in harvest robotics (Bac et al., 2013), phenotyping (van der Heijden et al., 2012) and disease detection (Polder et al., 2014), which require precise object classification and localisation. For example in harvest robotics, obstacle maps for motion planning need to have a resolution up to the plant part level (Bac et al., 2016, 2014a).

4.1.2 Hypotheses

Our main hypothesis was that only a small manually annotated empirical fine-tuning dataset would remain required for optimal and successful empirical learning after bootstrapping a convolutional neural network (CNN) with related synthetic data, as compared to other training methods.

To gain further insights in the dataset size requirements for learning, our additional hypothesis was that segmentation performance increases with larger synthetic or empirical dataset size, though will be limited at a certain order of magnitude.

During the experiments we developed further additional hypotheses and tests for causes of certain segmentation results, e.g. the training of binary classifiers to circumvent skewed learning and the addition of CRFs to improve local class segmentation boundaries.

Finally, we hypothesise that a empirically fine-tuned model can robustly generalise to related datasets, due to the distributed and hierarchical representation learning of CNNs. We define related datasets as images of the same crop, though under different conditions such as illumination, acquisition hardware or imaging distance.

Although previously we have provided brief evidence for our main hypothesis as a perspective towards future research (Barth et al., 2018), in this chapter we will expand on that work. We affirm our earlier experiments with a more advanced CNN architecture and place the results in a broader context in this chapter.

4.1.3 Requirements

Regarding our main hypothesis, we require a quantification of the small empirical dataset size. Although this number is arbitrary, we aim for a supervised machine learning methodology that needs no more of a manual annotation effort than 2 days. Given an average previously reported annotation time of 30 minutes per image in the used dataset (Barth et al., 2018), this translates to an upper bound of 30 images.

For optimal learning, we require that no other learning scheme that includes combinations of synthetic, empirical, other related and/or unrelated data for bootstrapping and/or fine-tuning has a higher performance on the empirical test set.

We define successful learning as the recognition of all classes, preferably with a low performance variance amongst classes. Success itself is quantified using the intersection-over-union measure, stated in Equation 6.2. However, because the extent of success is highly task dependent, we do not define hard requirement values. We can however indicate that for tasks such as detection, a relatively low (IOU ≥ 0.5) is sufficient because it is not the precise overlap that counts, but partial recognition suffices. However, for tasks such as phenotyping the measure is required to be high (IOU ≥ 0.9), since exact dimensions and morphology are of interest.

4.1.4 Contributions

Our work provides the field of computer vision in agriculture a pioneering methodology for state-of-the-art segmentation, whilst simultaneously reducing the constraints on labour intensive manual annotations. Results are a key part in the next leap of robotization in agriculture to keep up with the increasing demand of productivity and quality whilst decreasing the pressure on resources required (Bac et al., 2014b).

4.1.5 Research Context

The use of synthetic image data is emerging as a powerful tool in the computer vision community to generate training data for bootstrapping machine learning models. Such models can either be co-trained or fine-tuned with empirical data (Dittrich et al., 2014; Kondaveeti, 2016), on which the models are also deployed. Examples that show improved object recognition performance can be found in multiple domains, e.g. 3D human pose estimation from depth images (Shotton et al., 2013) and multi-modal magnetic resonance imaging for pathological cases (Cordier et al., 2016). Other notable examples for urban scene classification showed that synthetic images alone were sufficient as training data for a model applied to real scenes (Hattori et al., 2015), though accuracy was increased when combined with real training data (Ros et al., 2016).

To a certain extent, synthetic training and empirical fine-tuning can be seen as a form of soft transfer learning (Caruana, 1995; Bengio, 2011; Bengio et al., 2011), where a model can be successfully applied to a different task and domain. However, since in our case not only the task is equal but also the data is highly similar, we therefore adhere to the term bootstrapping.

The challenge of semantic segmentation in computer vision can be described as to dividing an image into non-overlapping meaningful regions, ultimately determining the object (part) class on a per-pixel level. Historically, semantic segmentation has been performed mainly supervised, although weakly or non-supervised approaches have also been successful for certain problems (Wehrens, 2010; Zhu et al., 2016). Compared to other methods that only generate a single high-level label description per image, semantic segmentation has the benefit of localisation in the image plane. This advantage is useful for applications such as robotics where object manipulation and navigation is dependent on accurate positional information, e.g. autonomously driving cars (Shapiro, 2016; Badrinarayanan et al., 2017), warehouse order picking robots (Zeng et al., 2016) or agricultural robots (Bac et al., 2013, 2016).

Specifically for agricultural robotics, per-pixel level annotations are either required or will improve performance as opposed to using coarse localisation methods (e.g. bounding box detection). We see the main applications in i) crop handling, ii) phenotyping and iii) disease detection.

Regarding crop handling, harvesting robotics requires precise end-effector placement at the target fruit (Bac et al., 2016; Li et al., 2016) or near the fruit such as a peduncle (Sa et al., 2017; Henten et al., 2009) or to avoid obstacles during that motion (Bac et al., 2013). Per-pixel class segmentations can provide the information to allow this precise end-effector placement when registered depth can be inferred by using 3D sensors, e.g. stereo imaging (Bac et al., 2014a) or time-of-flight cameras (van der Heijden et al., 2012). Other crop tasks, such as open field weeding, also requires to differentiate on a per-pixel level where the crop and weeds are (Milioto et al., 2017b,a), to allow for precise spraying (de Soto et al., 2016) or dutch hoeing (Hemming & Rath, 2001). Also for the task of leaf picking, per-pixel labels might be beneficial regarding precision over current bounding box approaches (Ahlin et al., 2016).

Regarding automated phenotyping, where the task is to correlate plant parameters with their underlying genetics to guide plant breeding, the localisation of plant parts in the image is a hard requirement (Araus & Cairns, 2014). Plant parameters such as leaf size (van der Heijden et al., 2012), stalk thickness (Vijayarangan et al., 2017) or spikelet counts (Pound et al., 2017) are to be estimated with high precision. Per-pixel segmentation is a key development for this domain. However, also for this task a registration with depth information is required to infer real world dimensions from the image coordinates that the segmentation provides.

For plant disease detection, the task is not only to determine which plant is healthy or diseased but moreover where on the plant the infection is localised (Phadikar & Sil, 2008) to allow local automated treatment (Oberti et al., 2013) or to map the phase or size of the infection in the crop to guide crop management (Lu et al., 2017).

Previously successful semantic segmentation methods were based on manually crafted features as input for shallow learning models such as support vector machines or random forests (Johnson et al., 2013; Fulkerson et al., 2009). For other computer vision tasks, similar methods were recently superseded by convolutional neural networks (Everingham et al., 2015). However, initially such methods could not perform semantic segmentation due to the convergent nature of the networks' architectures. Convolutional neural networks start with global information that is compressed in an increasingly spatially independent hierarchy of features, forming a distributed representation of the input whilst losing locality information. However, for semantic segmentation preserving the locality information is key.

At first, solutions for the locality problem appeared by learning additional objectives like coarse bounding boxes that represented the location of the object in the image (Uijlings et al., 2013; Pont-Tuset et al., 2015). Although often still preferred for speed or annotation costs, the downside of bounding box methods is the lack of segmentation detail. Later approaches tried merging classifications from multiple levels in the network's hierarchy combined with super-pixel pre-segmentation (Farabet et al., 2013). Recently, a novel architecture was presented using fully convolutional neural networks (Papandreou et al., 2015; Chen et al., 2015), differing from other networks by replacing the fully connected layers with convolutional ones and adding dense predictions using the *à trous* algorithm (meaning with holes, also reported as atrous) (Mallat, 1999). Furthermore, not uncommon with previous approaches, an integrated layer was added for applying a CRF as post-processing to refine the lost locality of the segmentation.

To further expand the challenge of object localisation, efforts were made to localise parts within those objects (Felzenszwalb & Huttenlocher, 2000, 2005), later also contrived for semantic segmentation (Wang & Yuille, 2015; Tsogkas et al., 2015). Although this refinement can be considered as merely increasing the number of classes, this would neglect the strong spatial correlations between object parts. Some methods applied compositional models and high-level information to include these relationships and improve on object part segmentations. With convolutional neural networks however, the distributed hierarchical representation of objects and their features facilitates learning these correlations.

This chapter will first describe the research materials in Section 4.2. The general methodology across experiments is then described in Section 4.3. In the sections that follow, each experiment is reported separately with their own introduction, method, results and discussion section. A general discussion is provided in Section 4.11 and we conclude the chapter in Section 4.12.

4.2 Materials

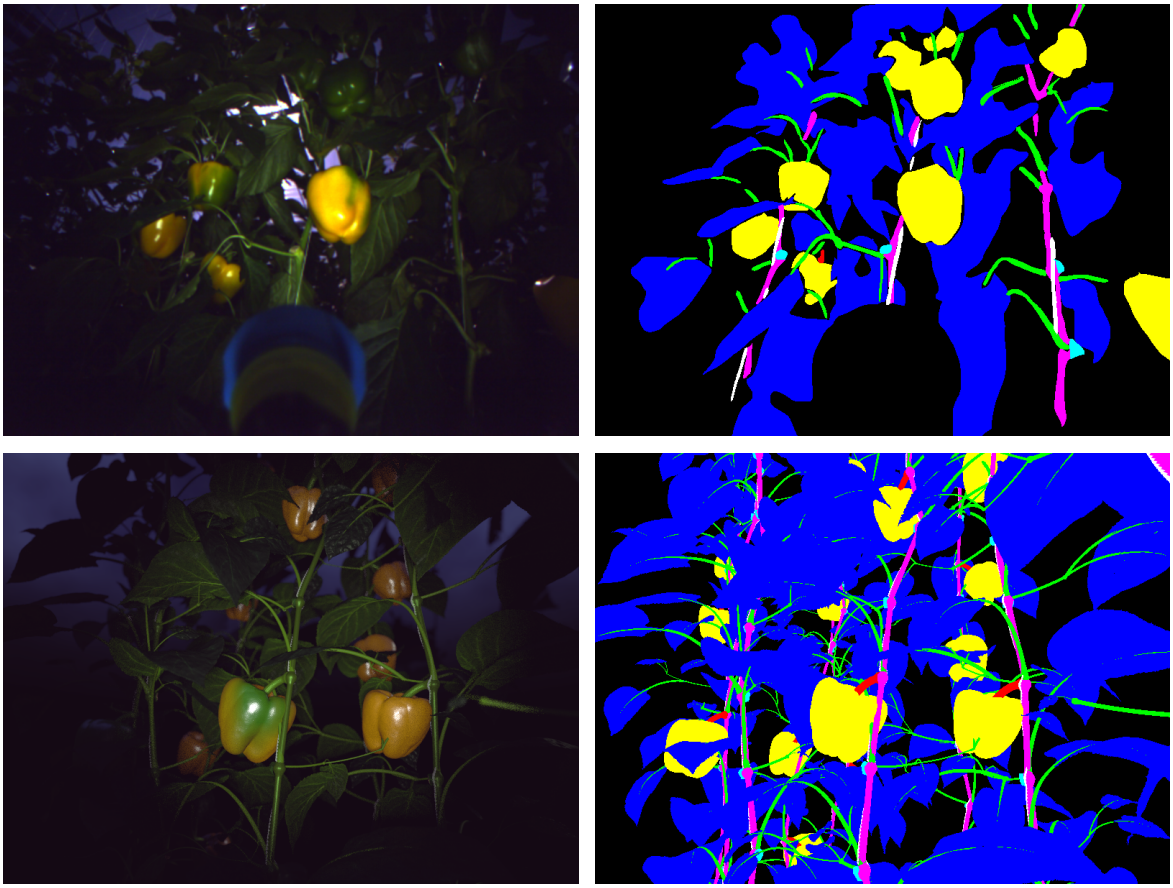


Figure 4.1: Examples of empirical (top) and synthetic (bottom) color images (left) and their corresponding ground truth labels (right). Class labels: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts where pepper where harvested.

4.2.1 Dataset Description

The *Capsicum annuum*, sweet- or bell pepper image dataset (Barth et al., 2018) consists of 50 empirical images of a crop in a commercial high-tech greenhouse and 10,500 corresponding synthetic images, modelled to approximate the empirical set visually and geometrically. The synthetic images were generated to reflect the empirical situation by rendering random 3D meshes of plants. These meshes were randomly generated using 21 empirically measured plant parameters. To create realistic images, the greenhouse growing architecture was modelled as well as similar camera and illumination settings for rendering.

In both image sets, 8 classes were annotated on a per-pixel level, either manually for the empirical dataset or automatically for the synthetic dataset. In Figure 5.4 examples of images of the dataset are shown. The dataset was publicly released at:
`http://dx.doi.org/10.4121/uuid:884958f5-b868-46e1-b3d8-a0b5d91b02c0`

4.2.2 Convolutional Neural Network Architectures

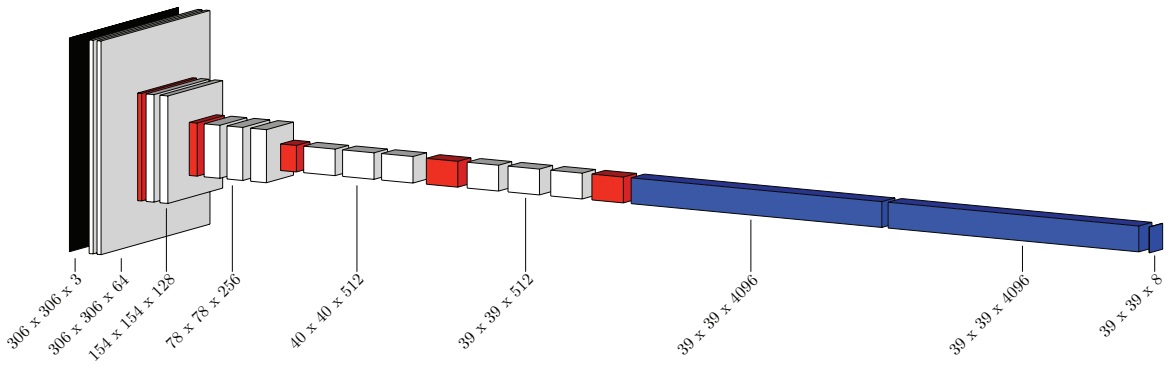


Figure 4.2: Convolutional neural network architecture of Deeplab Vanilla model, based on the VGG-16 composition. All models used in this chapter are modified versions of this architecture. Color encodes layer type: ● cropped input image, ○ convolutional layer + rectified linear unit, ● pool layer and ● fully connected layers transformed to fully convolutional ones. Dimensions indicate feature map width x height x number of feature maps.

For our experiments we used the publicly available fully convolutional neural network (CNN) architectures of DeepLab (Papandreou et al., 2015; Chen et al., 2015) implemented on top of Caffe (Jia et al., 2014). Although other deep learning implementations were being researched for semantic segmentation (Shelhamer et al., 2016; Mostajabi et al., 2014; Long et al., 2015), those were not yet available for verification at the time of our research. DeepLab models can either be trained with weakly semi-supervised learning (e.g. bounding boxes) or with strong supervision (e.g. per-pixel labels). For our approach, the detailed annotated dataset allowed training strong supervision models, resulting in more localised labeling as compared to bounding boxes.

The underlying architecture for the CNN models was based on VGG-16 (Simonyan & Zisserman, 2014) as depicted in Figure 4.2. VGG-16 was originally intended for global object detection. To adjust it for semantic segmentation, 2 changes were made to the architecture (Papandreou et al., 2015). First the fully connected multi-layered perceptron at the end of the network was replaced by fully convolutional layers (Long et al., 2015); hence the network was applied in a convolutional manner on the input image at its original resolution which resulted in per-pixel labels. However, given the used stride of the convolutions this resulted in down-sampled prediction. Therefore the second adjustment was made by implementing the *à trous* algorithm; by upscaling the filters with filler zeros, predictions at the original resolution could be made (Chen et al., 2016a). Although a better performing RESNET-101 implementation for DeepLab also exists (Chen et al., 2016a), we were not able to train such network due to GPU memory constraints.

To the base model DeepLab-Vanilla (Papandreou et al., 2015), the following additional adjustments to the architecture were previously made to explore the effect on segmentation performance on the PASCAL VOC 2012 dataset (Everingham et al., 2010b).

1 - **Increasing the field-of-view (DeepLab-LargeFOV).** Adjusted input stride to 12 and using a kernel size of 3x3 at the first fully convolutional layer, the receptive field was doubled in comparison to DeepLab-Vanilla whilst having a third of the number of free parameters (Chen et al., 2015).

2 - **Addition of multi-scale predictions (DeepLab-MSc)**. Improved segmentation accuracy at the boundaries of objects, the final feature map layer receives 5 additional feature maps convoluted from intermediate layers in the network (Chen et al., 2015).

3 - **Combination of the former two adjustments (DeepLab-MSc-LargeFOV)** (Chen et al., 2015).

4 - **Addition of an attention model on multiple scales (DeepLab-Attention).** The input image was resized to several scales and used for training both a network and an attention model. The attention model weighed each image scale and each feature thereof for the final segmentation (Chen et al., 2016b).

5 - **Addition of *à trous* spatial pyramid pooling (ASPP) (Deeplab-v2).** ASPP included image context at multiple scales by convolutional feature layers with different fields-of-view (Chen et al., 2016a; He et al., 2014).

For each model, pixels were independently classified without considering label agreement across the image. Applying a fully connected CRF (Krähenbühl & Koltun, 2011) can include long-range object dependencies and reduces label noise while refining part edge details. The CRF takes the CNN pixel prediction as the unary potentials and maximises label consistency by encouraging the assignment of locally similar labels that have similar properties. The DeepLab models can be extended with such a CRF as an extra integrated layer, although its parameters cannot be trained by back-propagation and should be found separately.

4.2.3 Hardware

Experiments were run on a NVIDIA DevBox system with 4 TITAN X Maxwell 12GB GPUs, Intel Core i7-5930K and 128GB DDR4 RAM running Ubuntu 14.04. As a dependancy for the DeepLab V2 Caffe version, the archived version of CUDA 7.5 was installed. Training a single model took 24 hours on average (using a batch size of 10 cropped images of 300x300 pixels per GPU and 30,000 iterations). Testing on a single 800x600 pixel image took around 200 ms.

4.3 Methods

We performed Experiments I through VII, each consisting of sub-experiments by varying dataset composition and/or model architecture. In Figure 4.3 an overview of the main Experiments is presented. Not all permutations of models, hyper-parameters, dataset types and sizes were explored due to the infeasible combinatorial computational cost. For this reason, the best performing model architecture was selected first by evaluating Experiment I, which was then further used for Experiments II-VII.

For each experiment, the same range of images was selected for the train, validation or test phase to make sure unique synthetic plant models or empirical plants were separated in the different phases and equal between experiments. The synthetic dataset consisted of 6 scenes of 1,750 images (10,500 total), with each scene containing unique plants (Barth et al., 2018). To ensure separation between unique images, the first 5 scenes with images 1-8,750 were used as synthetic training images whereas the remaining scene were used for validation images (8,851-8,900) and test images (8,751-8,800). Similarly, for the empirical dataset, images 1-30 of unique plants were used for training and the remainder images 31-40 for validation and 41-50 for testing.

The hyperparameters of the network were manually optimised using the validation dataset and a combination of models and dataset configurations as suggested by (Goodfellow et al., 2016; Bengio, 2012). Specifically, we searched for a learning rate that reduced the loss on the validation set gradually over the iterations towards zero. For the solvers Stochastic Gradient Decent, AdaDelta, Adaptive Gradient Descent, ADAM, Nesterov, and RMSprop, learning rates of 0.1, 0.01, 0.001 and 0.0001 were explored. We noted that none of the models overfitted on the validation set, though IOU performance differed between solvers and learning rates.

The hyperparameter search resulted in the choice of Adaptive Moment Estimation (ADAM) (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ and a base learning rate of 0.001 for 30,000 iterations with a batch size of 4. These chosen hyper-parameters were found to be consistently optimal for multiple experiments with different datasets and therefore we fixed them across all experiments. To each model, an adjustment was made in the layer weight initialisation procedure. We updated the models to using MSRA weight fillers (He et al., 2015; Mishkin & Matas, 2015). Furthermore, the dropout rate (Srivastava et al., 2014) was adjusted to 0.50 to improve generalisation.

As a performance measure the intersection-over-union (IOU) was used, as described in (He & Garcia, 2009; Everingham et al., 2010a; Barth et al., 2018) which is also known as the Jaccard Index similarity coefficient. The IOU can be determined per class or as an average over all classes. An IOU represents the intersection of the classified area and the ground truth area, divided by their union. The measure as an average over all classes is defined in Equation 6.2, where for each class their IOU equals the intersection of the semantic segmentation and the true labels divided by their union. To derive the measure, a pixel-level confusion matrix C is calculated first for each image I in dataset D, where $S^I_{gt}(p)$ is the ground truth label of pixel p in image I and $S^I_{ps}(p)$ is the predicted label. This implies that C_{ij} equals the count of pixels with ground truth label i and prediction j. The mean IOU over each class can be derived as an average over each class L.

Figure 4.3: Overview of performed main experiments I through VII, with sub-experiments A through G. Model architectures (rectangles) were trained, validated, tuned or tested with dataset types empirical (diamond), synthetic (circle) or different but related datasets (hexagon) with the number of image samples displayed within. A subscript indicates that only a single specific class was used.

$$IOU = \frac{1}{L}\sum_{i=1}^{L}\frac{C_{ii}}{G_i + P_i - C_{ii}} \text{ , where} \tag{4.1}$$

$$C_{ij} = \sum_{I \in D}\left|\left\{p \in I \mid S^I_{gt}(p) = i \wedge S^I_{ps}(p) = j\right\}\right|, and \tag{4.2}$$

$$G_i = \sum_{j=1}^{L} C_{ij} \text{ and } P_j = \sum_{i=1}^{L} C_{ij} \tag{4.3}$$

Hence G_i denotes the total number of pixels labeled with class i in the ground truth and P_j the total number of pixels with prediction j in the image.

Apart from quantitative evaluation, qualitative evaluation of the segmented images was performed to assess differences in segmentation style (e.g. course versus fine). Albeit two different models can achieve equal IOUs, the underlying distributions can be distinctive. Furthermore, the emergent classification property of the spatial distribution of the true and false positives can be determined.

Aside from previously mentioned regularisation methods, overfitting for each experiment was prevented by selecting the optimal model by periodically checking the performance on the validation set. This method of early stopping requires to select a point where performance either stabilised or decreased. This was done manually by evaluating the IOU per class over the iterations. In this research, the early stopping point found for each model was at 30.000 iterations.

4.4 Experiment I

CNN model architecture is a key factor to classification performance (Bengio, 2009). Proven base architectures are often modified with new insights to validate the enhancements on a range of benchmark datasets. To investigate how a set of model architecture modifications relate to part segmentation performance for our use-case, we compared 6 deep learning architectures in Experiments I-A through I-F, ordered by increasing expected performance according to previously obtained results by their respective authors. From these experiments we selected the best performing model for further Experiments II-VII. Assuming that the largest dataset for training results in optimal performance on the test set (Soekhoe et al., 2016), (a hypothesis we aimed to verify for our case with Experiment III) the full synthetic training dataset of 8,750 images was used to train each model in Experiment I. We did not use empirical data for Experiment I since we assumed performance and behaviour of the different architectures would be ranked similarly given comparable domain images.

4.4.1 Methods

The following experiments were run using synthetic images 1-8,750 for training, validation images 8,851-8,900 and testing images 8,751-8,800; I-A: DeepLab-Vanilla, I-B: DeepLab-MSc, I-C: DeepLab-LargeFOV, I-D: DeepLab-MSc-LargeFOV, I-E: DeepLab-Attention and I-F: DeepLab-V2.

The performance was compared quantitatively by evaluating mean IOUs over the test set images and over each class. Furthermore, the final segmentations were assessed qualitatively. The primary requirement for a model to be selected for future experiments was the ability to recognise all classes. The secondary requirement was high IOU performance relative to the other models.

4.4.2 Results

In Figure 4.4 the performances of model I-A through I-F are shown. For qualitative investigation, an example segmentation for each model with corresponding color and ground truth image is shown in Figure 4.5. Furthermore, the underlying per class probability heat maps are presented to provide insight into the raw output of the CNN.

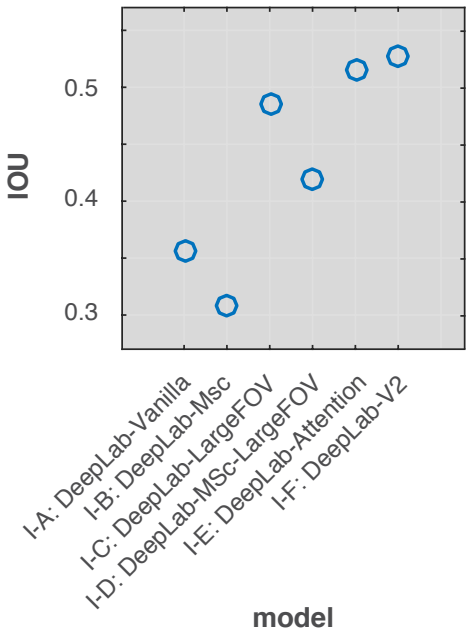


Figure 4.4: Results of Experiment I, displaying mean test set IOU over each class for each model architecture I-A through I-F.

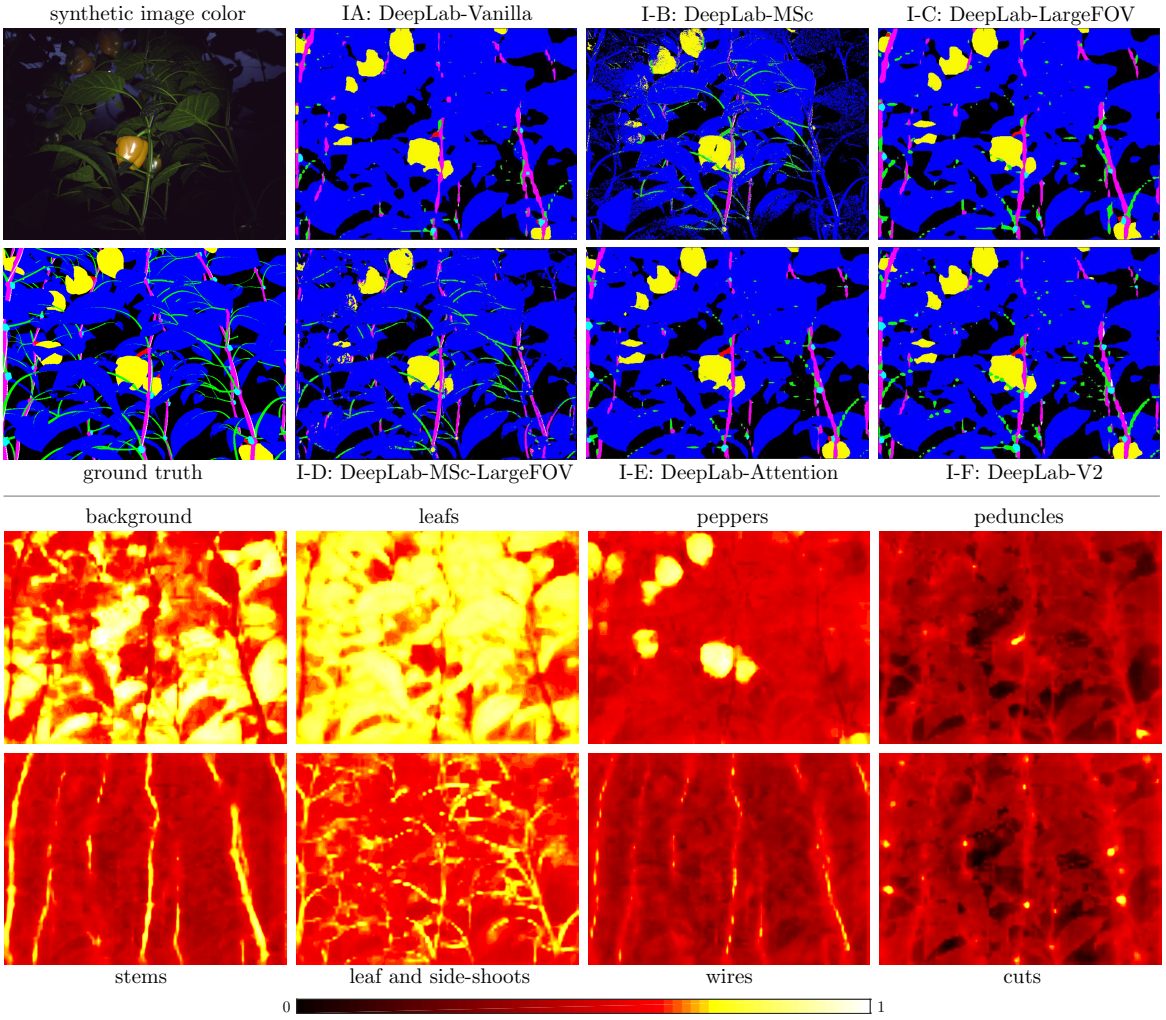


Figure 4.5: Results of Experiment I. The top two rows show segmentation results of Experiments I-A through I-F. Class labels: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts. The bottom two rows show per class probability maps of DeepLab-V2 for the synthetic color image from which the final segmentation was derived by selecting per pixel the class with the highest value of all maps.

4.4.3 Discussion

In Figure 4.4 we observed that all of the implemented modifications to the base model (DeepLab-Vanilla) improved the mean IOU results, except for adding 5 feature maps convoluted from intermediate layers in the final feature map (MSc) (Chen et al., 2015). Evaluating Figure 4.5, the models fail to learn the uncommon classes or darker areas in the image, e.g. wires and cuts. The commonality in the datasets was determined as the percentage of ground truth pixels (in decreasing order): background, leafs, peppers, shoots and leaf stems, main stems, wires, peduncles and cut peduncles. This is further supported with evidence provided in 6.9), where the IOU per class on the validation set can be observed over the training time for each model.

Figure 4.5 shows the qualitative performance differences between models. DeepLab-MSc had the desired property of sharp segmentation boundaries, though failed to cope with pixels lacking distinct color in the outer area of the image. Furthermore, combining MSc with LargeFOV resulted in the neglect of uncommon classes, as can be observed in 6.9 I-D. The other models appeared coarse around the plant parts, providing more true positive detections.

In Figure 4.5 the per class probability maps by DeepLab-V2 for a synthetic example image is shown. These distributions gave insight in the underlying learned class probabilities. It shows that the stem and wire classes were highly overlapping and the final segmentation of the wires was often overruled by the stem class. Furthermore, although leaf stems and side-shoot segmentations were sparsely present in the final segmentation, the model appeared to learn the individual probability distributions quite well. Plausibly, learning binary classifiers for these classes should improve IOU performance, which we investigated in Experiment V.

Overall, the most recent proposed modification DeepLab-V2; the addition of *à trous* spatial pyramid pooling (Chen et al., 2016a), had the highest IOU. Moreover, it is being able to learn all plant parts (6.9)) hence meeting one of our requirements. We therefore selected the DeepLab-V2 model for Experiments II-VII.

Future research is suggested of adding the beneficial DeepLab-MSc sharp edge properties to DeepLab-V2, without suppressing the uncommon classes.

4.5 Experiment II

We hypothesise that synthetic bootstrapping and fine-tuning with a small empirical dataset can improve performance over other learning strategies. Previously we explored this briefly using the DeepLab-Vanilla model (Barth et al., 2018). In this chapter we try to further validate those results and expand on that work. We ran the following 5 experiments, using the DeepLab-V2 model.

4.5.1 Methods

The motivation for each experiment is given below and the used image indices are shown between brackets. To evaluate the performance, the mean IOU over the test set images and over all classes for each experiment was obtained.

II-A DeepLab-V2. *Train: synthetic (1-8,750). Test: synthetic (8,751-8,800).*
This experiment was run to obtain a baseline performance of the model when having access to a large and detailed annotated dataset for this domain. Assuming performance increases with dataset size until the model's complexity is saturated (Zeiler & Fergus, 2014), this experiment provides insight into the theoretical upper bound of the performance of all experiments in II. This assumption is further tested in Experiment III.

II-B DeepLab-V2. *Train: synthetic (1-8,750). Test: empirical (41-50).*
Determines to what extent a synthetically trained model can generalise to a similar set in the same domain (e.g. empirical images) without fine-tuning. If performance would approximate the performance obtained in I-F, this is evidence against the aspect of our main hypothesis that fine-tuning improves performance.

II-C DeepLab-V2. *Train: empirical (1-30). Test: empirical (41-50).*
Investigates if the model can learn with only a small empirical dataset. If performance would approximate results of Experiment I-F, this is evidence against the aspect of our main hypothesis that a *large* dataset for bootstrapping would be required for improved performance.

II-D DeepLab-V2. *Train: PASCAL VOC. Fine-tune: empirical (1-30). Test: empirical (41-50).*

Compares the effect of bootstrapping with a non-related dataset. If the performance approximate the performance obtained in I-F, this is evidence against the aspect of our main hypothesis that a *related* dataset of the same domain is needed for improved performance.

II-E DeepLab-V2. *Train: synthetic (1-8,750). Fine-tune: empirical (1-30). Test: empirical (41-50).*

Assesses the performance of bootstrapping with a related dataset and fine-tuning with a small empirical set. Given our main hypothesis, this experiment is expected to achieve best performance on empirical data.

4.5.2 Results

The IOU results for each experiment are shown in Figure 4.6 and were split into per class IOUs in Figure 4.8. Segmentation results for the best performing model on synthetic data II-A and empirical data II-E are shown in Figure 4.7.

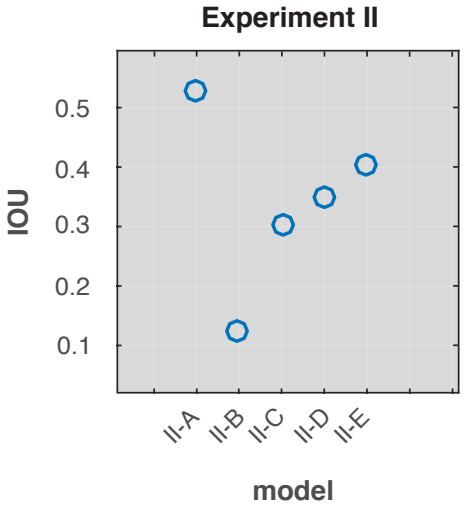


Figure 4.6: Results of Experiment II-A through II-E. The mean IOU over the test set images and over all classes for each experiment is displayed

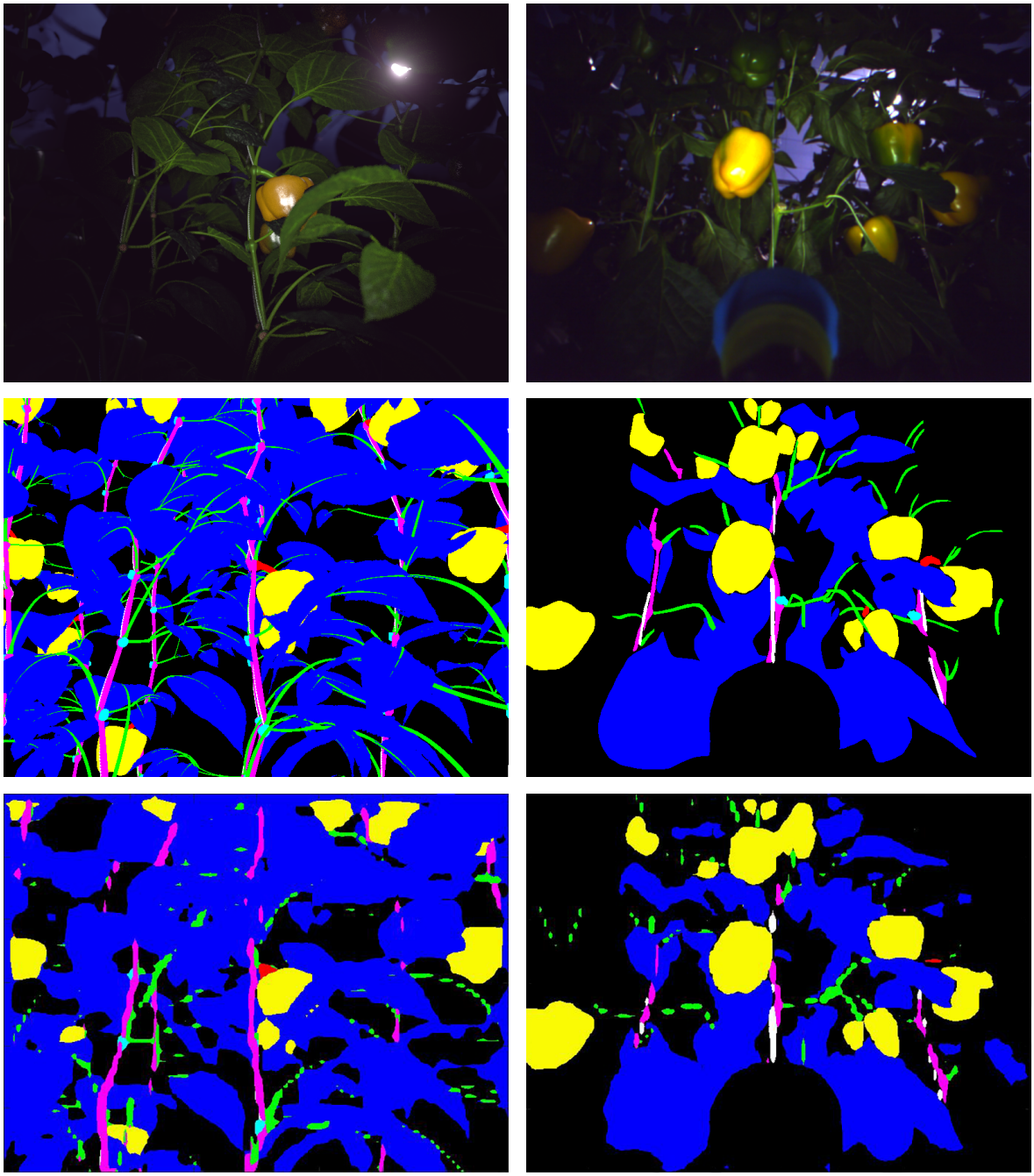


Figure 4.7: Example segmentation results for synthetic test set from experiment II-A (left column) and empirical test set from experiment II-E (right column). Color images (top row), ground truth (middle row) and classification segmentation (bottom row) are shown. Class labels: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts.

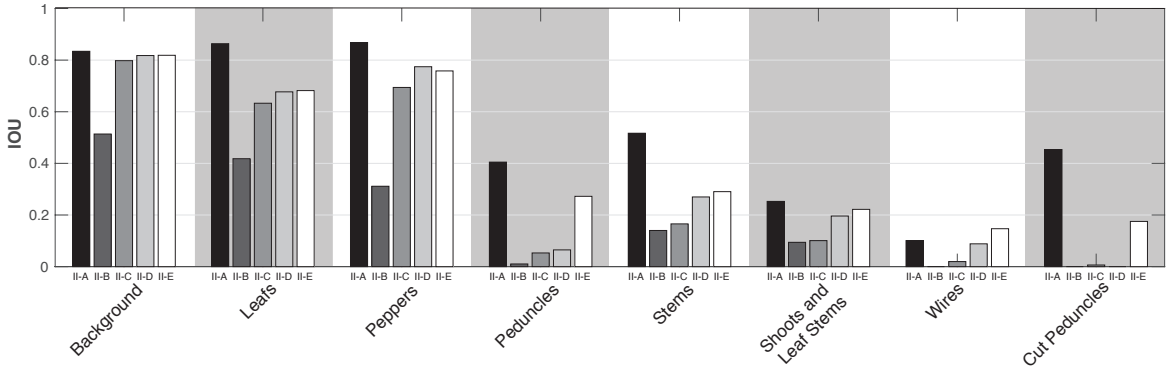


Figure 4.8: Results of Experiment II. For each class, the mean test set IOU per class is displayed for each model II-A through II-E.

4.5.3 Discussion

From the quantitative results in Figure 4.6 we derived the following.

II-A This model indicated a benchmark or baseline of optimal performance when the model had access to a large dataset with perfect ground truth. We assumed a positive correlation between dataset size and classification performance (Banko & Brill, 2001; Brants et al., 2007), to be further validated in Experiment III.

II-B Without fine-tuning, the synthetically bootstrapped model could not generalise properly to empirical data.

II-C When training a model using only a small empirical dataset, the performance of the most common classes approached the baseline performance of II-A, as can be seen in Figure 4.8 and 6.9. However, the model failed to discriminate the uncommon classes. It appeared the model only learned the most color discriminative classes.

II-D A model bootstrapped with a non-related dataset (PASCAL-VOC) that was fine-tuned with empirical data, resulted in increased performance on empirical data compared to the former experiments II-B and II-C where no fine-tuning was used. This implies fine-tuning on a bootstrapped network was beneficial. Any CNN network requires training time and a large dataset to converge to an effective feature distribution. Bootstrapping provides a stable starting point, from which the fine-tuning can quickly converge to a new optimum of the new dataset.

II-E The best IOU performance on empirical data and inclusion of all classes (see 6.9) was achieved when bootstrapping on related synthetic data, confirming our hypothesis that a synthetic bootstrapping using synthetic data and fine-tuning with empirical data results in optimal learning.

When evaluating the results qualitatively in Figure 4.7 we observe very high quality results in plant part recognition in both dataset types. Furthermore, although the IOU for some classes seems relatively low (IOU<0.3) and segmentations were therefore not completely overlapping with the ground truth, we do observe good recall for each part nonetheless.

Note that for the empirical segmentation, parts in the image were detected that were not annotated manually in the ground truth due to dark regions, but were present in the image. Hence these were evaluated as false positives, although they would be true positives if human annotation was perfect. Hence this annotation bias resulted to a lower reported mean IOU.

For both segmentations it holds that elongated parts were not connected. This was likely due to the *à trous* algorithm that upscales a sparse low resolution input feature map (Chen et al., 2016a). Post processing with CRF might solve this issue, as investigated with Experiment VI in Section 4.9.

4.6 Experiment III

This experiment investigated the hypothesis of the positive correlation between segmentation performance and dataset size for our use-case, given a model with sufficient learnable parameters (Soekhoe et al., 2016; Zeiler & Fergus, 2014).

4.6.1 Methods

The DeepLab-V2 model was trained with logarithmically increasing synthetic dataset size, with image ranges III-A: 1-$10^{1.0}$, III-B: 1-$10^{1.5}$, III-C: 1-$10^{2.0}$, III-D: 1-$10^{2.5}$, III-E: 1-$10^{3.0}$, III-F: 1-$10^{3.5}$ and III-G 1-$10^{3.942}$ ($\approx$ 8,750).

4.6.2 Results

In Figure 4.9 the results for Experiment III are shown. For the underlying IOU distribution per class and over time, refer to Appendix A.

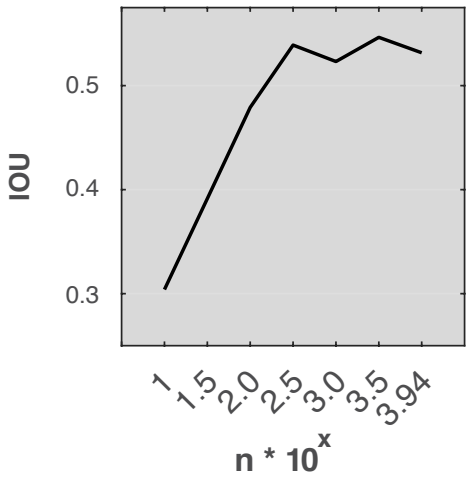


Figure 4.9: Results of Experiment III: for an increasing synthetic dataset training size n, the mean IOU over the test set images and over all classes is displayed.

4.6.3 Discussion

Observing Figure 4.9, the segmentation performance increased with dataset size though seemed to settle around 3,500 training images. As can be seen in 6.9, the performance increase was mainly due to the rising correct classification of uncommon classes.

These results confirms our hypothesis that dataset size positively correlates with performance, in line with previous research of others (Zeiler & Fergus, 2014; Soekhoe et al., 2016). Additionally, it provided us the upper bound requirement for synthetic dataset size, as more synthetic training images do not further increase performance. However, the cause of the performance stabilisation might be twofold. First, our synthetic plant models may only capture a realistic but limited variation within the implemented plant parameter bounds. Hence, although each plant and scene was randomised (Barth et al., 2018), there should be an upper limit where generating new images merely adds redundant information. Therefore, if the synthetic models would improve on the variance similarity with the empirical situation, i.e. incorporating the wider range of plant parameters to increase diversity in the model, the performance might keep increasing further with dataset size.

Second, it might also be the case that the learning ability of the CNN model was saturated, meaning all weights could already be exploited and therefore the network was not able to absorb more information. A possible solution would be to raise the network's complexity, e.g. by increasing the number of feature maps and layers, whilst looking out for the overfitting pitfall using proper regularisation (Goodfellow et al., 2016).

4.7 Experiment IV

Until now the experiments investigated i) which model would likely to be most suitable for our domain, ii) whether it was possible to empirically fine-tune a synthetically bootstrapped model and iii) how much synthetic data was required to bootstrap a model. As posed in Section 4.1.2, our main hypothesis is that only a small manually annotated dataset would be required for (ii). Experiment IV aims to further specify the dataset size requirements as evidence for this hypothesis by evaluating how much empirical images are required to fine-tune a synthetically bootstrapped model.

4.7.1 Methods

We fine-tuned the DeepLab-V2 model that was synthetically bootstrapped with images 1-8,750 with an increasing empirical dataset ranges of: IV-A: 1-5, IV-B: 1-10, IV-C: 1-15, IV-D: 1-20, IV-E: 1-25 and IV-F: 1-30. The performance was compared quantitatively by evaluating the mean IOU over the test set images and over all classes for each experiment.

4.7.2 Results

In Figure 4.10, the IOU results are shown for an increasing empirical fine-tuning dataset size.

4.7.3 Discussion

In Figure 4.10 we observed that already with 5 empirical fine-tune images a reasonable performance can be achieved when a model is bootstrapped synthetically with related images (IOU=0.306). This provides insight into the lower bound of manual annotated data required for fine-tuning, although this remains highly dependent on the IOU needed of a specific task. Results confirm our hypothesis only a small annotated dataset is sufficient for successful learning and meets our requirement of up to 30 annotated images.

From the figures in 6.9 we derived that fine-tuning settles in a minimum rapidly and furthermore overfitting is likely to occur when the training is not stopped prematurely.

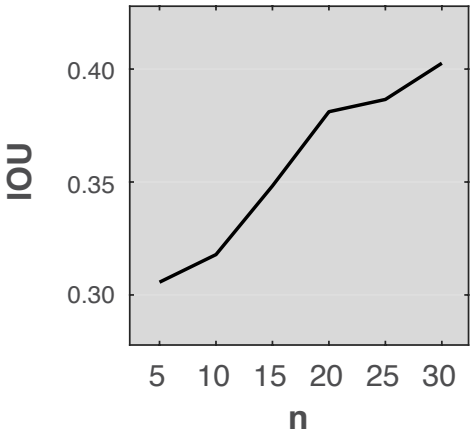


Figure 4.10: Results of Experiment IV: for an increasing empirical fine-tuning dataset size n , the mean IOU over the test set images and over all classes is displayed.

Again the hypothesis was confirmed that an increase of dataset size improves performance. In our case a relative increase of 32% was achieved using 30 images as opposed to 5. Additionally, performance increase did not yet seem to settle, indicating that the small dataset did not yet cover all empirical variance and more empirical data is likely to further increase performance.

4.8 Experiment V

During the experiments we noted that common classes were better classified than uncommon classes.

A possible explanation could be in the nature of CNN weight learning. The weights of a convolutional neural network classifier move over the error landscape on the direction of the average gradient of the mini-batch. In the DeepLab architectures, a single training example consists of a cropped image of 300x300 pixels, assuming a batch size of 1. However, this image does not count as a single training example; instead the error of each pixel is used to determine the gradient. Given that the weights update in the average direction of the error, the gradient might therefore be biased towards common classes.

A possible solution is to normalise the per class error during the loss function computation. However, initial experiments where the loss was normalised by the number of each label present (as opposed to summing the loss) did not yield a significant difference in performance for any class.

We hypothesised further that the performance of individual part classes could be boosted by an aggregation of dedicated binary CNN models, one for each class. By applying this learning strategy, the error landscape is assumed to be simplified and therefore easier to learn with a bias to a single class.

4.8.1 Methods

Experiment V trained binary DeepLab-V2 models per plant part. To compare performance with the non-binary V2 model, the binary segmentations were aggregated by overlaying the output in descending order of IOU performance. The mean IOU over the test set images and over all classes of the aggregation was then compared to that of Experiment III-G, because those results were obtained using DeepLab-V2 without binary training.

4.8.2 Results

In Figure 4.11 results are shown for the regular and binary models. In 6.9 the per class IOU performance can be observed.

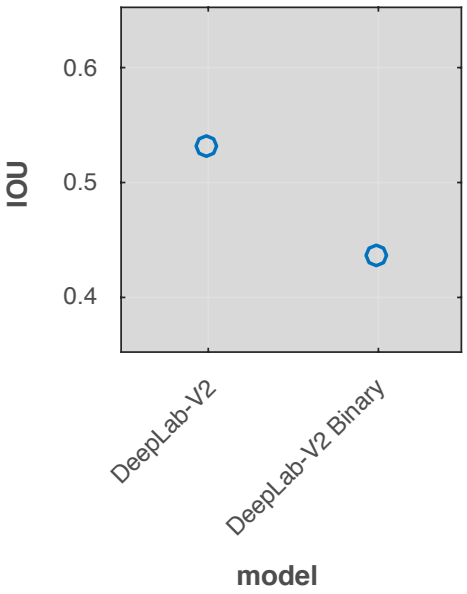


Figure 4.11: Results of Experiment V. The mean IOU over the test set images and over all classes plotted for the DeepLab-V2 and aggregated binary DeepLab-V2 models.

4.8.3 Discussion

In Figure 4.11 the results show that performance decreased relatively with 17% when training binary, as opposed to our hypothesis that this would improve results. Specifically the 'Cuts' class underperformed significantly, as can be seen in 6.9.

Although this experiment used a different training scheme, the underlying availability bias of each class remained equal. To cope with this difference, we suggest to balance the training data by cropping or masking the input proportionally to the class distributions presented in (Barth et al., 2018).

As each binary classifier was initialised with equal color normalisation parameters, based on the average color distribution over all classes, the data for each binary classifier was not zero-centered and normalised. However, attempts to normalise the data accordingly did not yield significant results.

4.9 Experiment VI

DeepLab models that were previously enhanced by adding fully conditional random fields (Krähenbühl & Koltun, 2011), showed improved segmentation performance for the PASCAL-VOC dataset (Chen et al., 2015, 2016a, 2015). In the previous Experiments I-V without CRF, the final classification for each pixel was determined by taking the maximum from all softmax class prediction layers. However, this approach disregards local and global label agreement, as similar labels tend to be clustered together and some labels co-occur more frequently than others. Applying a CRF can introduce local and global label agreement, usually resulting in a refinement of segmentation accuracy around the label edges.

4.9.1 Methods

Experiment VI optimised a CRF for the output of a synthetically trained model (1-8,750) that was applied on synthetic data (8,751-8,800) (VI-A) and that was fine-tuned (1-30) and applied to empirical images (41-50) (VI-B). The optimisation of the CRF comprised of a selection of hyper-parameters, as described in (Krähenbühl & Koltun, 2011). To obtain the CRF parameters, we performed a coarse to fine grid search over a subset of possible parameters in the range of [0 10 20 40] on the validation sets, similar to (Chen et al., 2015). The following values provided maximum IOU on the validation set of the synthetic data: w_1=0.4, w_2=1.6, σ_α=0.1, σ_β=0.04 and σ_γ=9 and on the empirical data: w_1=0.2, w_2=0.5, σ_α=0.15, σ_β=0.04 and σ_γ=5.

The performance was compared quantitatively by evaluating the mean IOU over the test set images and over all classes before and after applying the CRF. Furthermore, the final segmentations were assessed qualitatively to evaluate how the CRF influenced the segmentation.

4.9.2 Results

Post-processing the class probability maps output (see Figure 4.5) of the DeepLab-V2 model using CRF resulted in an average IOU increase of 1.51% on the synthetic set but yielded marginal performance increase of 0.01% on the empirical set. To provide insight in qualitative improvement on the synthetic set, Figure 4.12 displays an exemplary segmentation result with and without CRF as compared to the ground truth.

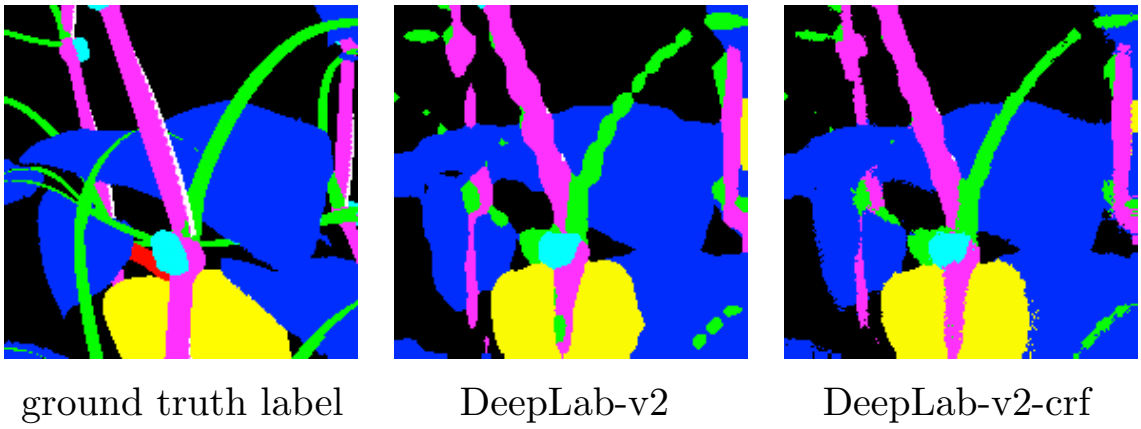


Figure 4.12: Qualitative result of Experiment VI on a cropped image. The segmentation of the DeepLab-V2 model (middle) as compared to the ground truth (left) and with CRF post-processing (right).

4.9.3 Discussion

Post-processing with CRF improved segmentation performance on the synthetic dataset both qualitatively as quantitatively with a relative +1.51%. The improvements were comparable to previously obtained results in other datasets (+3%) (Chen et al., 2015). Qualitatively we observed in Figure 4.12 that disconnected circular class regions were connected and smoothed with sharp edges.

Unfortunately these results were not duplicated for the empirical dataset. Hence our hypothesis that the addition of CRFs improves local class segmentation boundaries only partially holds. Extended parameter search for the CRF did not provide better results. A possible explanation could be the small empirical test set size (10) as compared to the synthetic test set data size (50).

4.10 Experiment VII

To test the DeepLab-V2 bootstrapped and fine-tuned model on generalisation power and robustness, we applied it to different but related datasets. Results provide insight in how applicable a model will be to new conditions. In turn this gives an estimate of the required additional annotation efforts when image acquisition conditions or scenes change.

4.10.1 Methods

In Experiment VII-A we deployed the model to datasets with equal empirical conditions but with different acquisition distances of 30, 20 and 10 cm from the crop. In Experiment VII-B we tested on a previous sweet-pepper image database obtained with different acquisition hardware. This dataset differed in artificial illumination, camera exposure settings, color calibration and crop season.

Due to the absence of a ground truth for these datasets, performance was compared qualitatively by evaluating the final segmentations. Specifically, we looked at the recognition of all classes and qualitatively at true and false positives.

4.10.2 Results

In Figure 4.13 exemplary segmentation results of this experiment are shown.

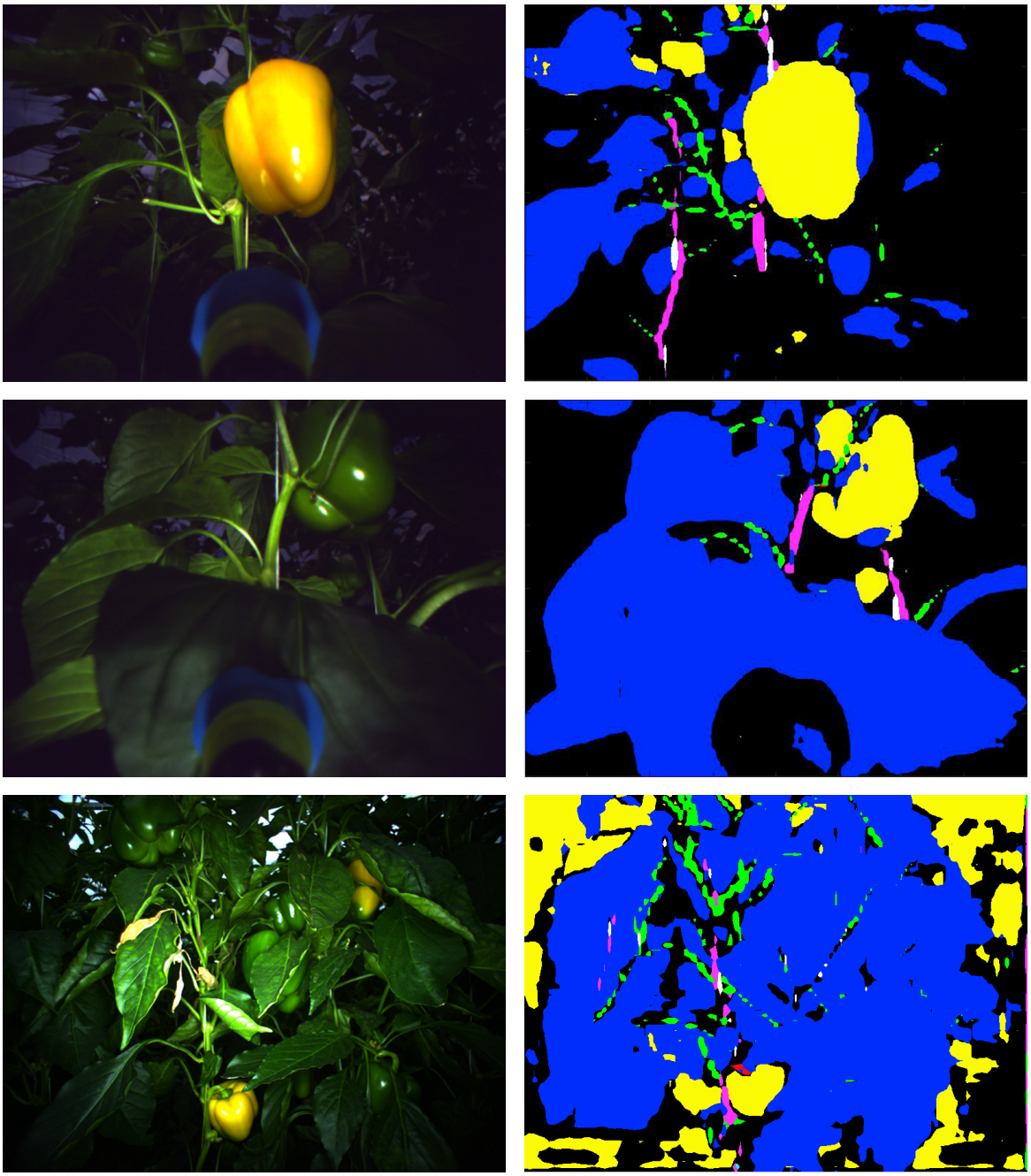


Figure 4.13: Example segmentation results (right column) of the synthetically trained, empirically fine-tuned DeepLab-V2 model to images (left column) taken from 15 cm (row 1), 10 cm (row 2) and using different illumination hardware, exposure settings, color calibration and season (row 3). Class labels: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts.

4.10.3 Discussion

Evaluating the exemplary results in Figure 4.13, we observed that the generalisation capability of an empirically fine-tuned CNN to other datasets was quite successful.

Images with similar hardware on closer distances were segmented similarly as the training set distance, although the number of false positive segmentations seemed to increase. Furthermore, not all classes seemed to be recognised (e.g. cuts).

Images from the different hardware, illumination conditions and color calibration and season, were still segmented fairly well, though mostly in the centered region of the image. Around the edges the probability for false positives seemed to increase, most probably due to relatively darker edges as compared with the empirical dataset on which it was trained.

The results suggest that empirical dataset fine-tune images do not necessarily have to be equal to the target image situation during deployment, as certain generalisation of the model can be expected. However, some performance degradation can be seen. Hence small additional manual annotation efforts are likely to be required. We partially confirmed our hypothesis that empirically fine-tuned CNNs can be applied robustly to related datasets.

	Leafs	Peppers	Peduncles	Stems
CART (Bac et al., 2013)	73.6	54.5	49.5	40.0
DeepLab-V2	78.5	34.5	78.6	21.6

Table 4.1: True Positive rates of CART on part detection in hyperspectral sweet-pepper images and our CNN model.

4.11 General Discussion

We compared our work to previously reported True Positive (TP) rates for plant part image segmentation in sweet-pepper (Bac et al., 2013) that used classification and regression trees (CART) on multi-spectral data. Performance differs as shown in Table 4.1. However, the reported measure in itself did not take into account other measures such as False Positives (FP). Furthermore, because TP rates can be maximised at the expense of increased FP rates, the overall performances of the methods were not directly comparable. However, as opposed to their conclusion (Bac et al., 2013), we hypothesise our segmentation results would be usable as reliable input for an obstacle map, because we obtained an empirical stem part IOU$\approx$0.5. Furthermore, we observed qualitatively a low amount of false positive detections that might obfuscate a obstacle map.

In previous research by others, pursuing a comparable goal of segmenting plant species using synthetic image data (Cicco et al., 2016), similar results were obtained. Whilst training only on synthetic data and testing on empirical data, one of the models IOU performance for leaf class was 60.2, whereas when fine-tuned with empirical data the performance increased with 23% to 74.1. In our case, using the same training and testing methodology, the leaf class performance increased with 75% from an IOU of 0.4 to 0.7 (see Appendix A). However, we must note that the results were not directly comparable due to the difference in number of classes in each approach and the differing amount of empirical training data (30 vs 900). The authors do conclude similarly that a synthetic dataset can improve segmentation performance over the use of solely empirical images.

Although we aimed for a comprehensive set of experiments to obtain observations for our hypothesis and search for dataset requirements, we understand that our exploration was not exhaustive. However, we think our results show a clear direction of how important factors such as synthetic bootstrapping, architecture and dataset size and type influence part segmentation performance. Results showed state-of-the-art performance, given a minimal amount of manually annotated empirical fine-tuning data.

The results of Experiment III raised an important question we could not capture in our experiments. It remains unclear to what extent the complexity of the plant model and synthetic images influences the performance of bootstrapping. To answer this, the dataset complexity should be varied. However, this resided outside the scope of this chapter and we suggest this as future research.

Related to this direction, would be informative to investigate the use of generative adversarial networks (GAN) to improve synthetic images towards the feature distribution of the empirical data (Shrivastava et al., 2016; Goodfellow et al., 2014).

Currently the DeepLab models do not differentiate between instances of parts. Future research on instance aware segmentation (Dai et al., 2016) could further improve the usability of the segmentations by discriminating between individual parts, for example by the MASK R-CNN architecture (He et al., 2017).

The difference between IOU performance of VOC and related dataset bootstrapping (II-D and II-E), was shown to be 15% relative and 0.05% absolute. It might be argued that related synthetic bootstrapping might not be worthwhile over bootstrapping with unrelated commonly available datasets. However, when we evaluated Figure 4.8 and 6.9, it shows that unrelated bootstrapping fails to recognise uncommon classes such as peduncles and cut peduncles. Our approach with related synthetic bootstrapping shows best performance and did recognise all classes.

Although there is a tradeoff measured in time investment between creating a synthetic dataset (Barth et al., 2018) and manually annotating additional empirical data, our results show that when empirical dataset size is a constraint and uncommon parts are required to be recalled, synthetic bootstrapping can provide a solution. Moreover, when a synthetic model is created formerly, it can be quickly used to generate synthetic data under a broad set of new application conditions whilst minimising the manual annotation requirement.

4.12 Conclusion

In this chapter we showed a methodology to reduce the current bottleneck of the reliance on manually annotated images that state-of-the-art machine learning requires. We provided evidence for our hypothesis that only a small manually annotated empirical fine-tuning dataset is still needed for optimal and successful empirical learning after bootstrapping a convolutional neural network (CNN) with related synthetic data. Our results show only 30 empirical training images were sufficient to obtain a mean IOU performance over all classes of 0.40. Furthermore, our method approached the synthetic baseline performance with a mean IOU over all classes of 0.53. Regarding our requirements, our method was i) unique in ensuring the recognition of all classes, ii) was optimal compared to other learning strategies and iii) was evaluated qualitatively successful and had desired quantitative results for tasks such as part detection.

Experiments confirm our hypothesis that performance is positively correlated with dataset size both for the synthetic and empirical datasets, although there is an upper limit of synthetic data where performance stabilises. We suggested further research to further improve performance by increasing model complexity or synthetic data variance.

Of the VGG-16 model architectures that were investigated, the addition of *à trous* spatial pyramid pooling proved to be most effective. The post-processing by conditional random fields yielded a small performance boost in the synthetically trained networks, though failed to improve in the same amount on the empirical data. Training binary classifiers to improve uncommon class performance did not yield improved results. The generalisation capability to images under different conditions was demonstrated as feasible, though not equal in performance as when fine-tuned.

Our work provides the field of computer vision in agriculture a pioneering methodology for state-of-the-art segmentation performance, whilst simultaneously reducing the reliance on labour intensive manual annotations. Results are a key part in the next leap of robotization in agriculture to keep up with the increasing demand of productivity and quality whilst decreasing the pressure on resources required.

4.13 Acknowledgements

The authors would like to thank prof. dr. R. Howe and dr. D. Perrin for their input of this research and making computing resources available. This research was partially funded by the European Commission in the Horizon2020 Programme (SWEEPER GA no. 644313).

References

Ahlin, K., Joffe, B., Hu, A.-P., McMurray, G., & Sadegh, N. (2016). Autonomous leaf picking using deep learning and visual-servoing. *IFAC-PapersOnLine*, *49*, 177 – 183. doi: https://doi.org/10.1016/j.ifacol.2016.10.033. 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2016.

Araus, J. L., & Cairns, J. E. (2014). Field high-throughput phenotyping: the new crop breeding frontier. *Trends in Plant Science*, *19*, 52 – 61. doi: https://doi.org/10.1016/j.tplants.2013.09.008.

Bac, C., Hemming, J., & van Henten, E. (2013). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture*, *96*, 148 – 162. doi: http://dx.doi.org/10.1016/j.compag.2013.05.004.

Bac, C., Hemming, J., & van Henten, E. (2014a). Stem localization of sweet-pepper plants using the support wire as a visual cue. *Computers and Electronics in Agriculture*, *105*, 111 – 120. doi: http://dx.doi.org/10.1016/j.compag.2014.04.011.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014b). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Bac, C. W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & van Henten, E. J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, *146*, 85 – 97. doi: http://dx.doi.org/10.1016/j.biosystemseng.2015.07.004. Special Issue: Advances in Robotic Agriculture for Crops.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PP*, 1–1. doi: 10.1109/TPAMI.2016.2644615.

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* ACL '01 (pp. 26–33). Stroudsburg, PA, USA: Association for Computational Linguistics. doi: 10.3115/1073012.1073017.

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. V. (2018). Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Com-*

puters and Electronics in Agriculture, *144*, 284 – 296. doi: https://doi.org/10.1016/j.compag.2017.12.001.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, *2*, 1–127. doi: 10.1561/2200000006.

Bengio, Y. (2011). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27* UTLW'11 (pp. 17–37). JMLR.org.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR*, *abs/1206.5533*.

Bengio, Y. et al. (2011). Deep learners benefit more from out-of-distribution examples. In G. J. Gordon, & D. B. Dunson (Eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)* (pp. 164–172). Journal of Machine Learning Research - Workshop and Conference Proceedings volume 15.

Brants, T., Popat, A. C., Xu, P., Och, F. J., & Dean, J. (2007). Large language models in machine translation. In *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 858–867).

Caruana, R. (1995). Learning many related tasks at the same time with backpropagation. In *In Advances in Neural Information Processing Systems 7* (pp. 657–664). Morgan Kaufmann.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & L. Yuille, A. (2016a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, . *PP*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*.

Chen, L. C., Yang, Y., Wang, J., Xu, W., & Yuille, A. L. (2016b). Attention to scale: Scale-aware semantic image segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3640–3649). doi: 10.1109/CVPR.2016.396.

Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., & Yuille, A. (2014). Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cicco, M. D., Potena, C., Grisetti, G., & Pretto, A. (2016). Automatic model based dataset generation for fast and accurate crop and weeds detection. *CoRR*, *abs/1612.03019*.

Cordier, N., Delingette, H., Le, M., & Ayache, N. (2016). Extended modality propagation: Image synthesis of pathological cases. *IEEE Transactions on Medical Imaging*, *PP*, 1–1. doi: 10.1109/TMI.2016.2589760.

Dai, J., He, K., & Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dittrich, F., Woern, H., Sharma, V., & Yayilgan, S. (2014). Pixelwise object class segmentation based on synthetic data using an optimized training strategy. In *Networks Soft Computing (ICNSC), 2014 First International Conference on* (pp. 388–394). doi: 10.1109/CNSC.2014.6906671.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, *11*, 625–660.

Everingham, M., Eslami, S. M., Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vision*, *111*, 98–136. doi: 10.1007/s11263-014-0733-5.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010a). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010b). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*, 1915–1929. doi: 10.1109/TPAMI.2012.231.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2000). Efficient matching of pictorial structures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)* (pp. 66–73 vol.2). volume 2. doi: 10.1109/CVPR.2000.854739.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*, 55–79. doi: 10.1023/B:VISI.0000042934.15159.49.

Fulkerson, B., Vedaldi, A., & Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, *116*, 8 – 19.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 2672–2680). Curran Associates, Inc.

Hattori, H., Boddeti, V. N., Kitani, K., & Kanade, T. (2015). Learning scene-specific pedestrian detectors without real data. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3819–3827). doi: 10.1109/CVPR.2015.7299006.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 1263–1284. doi: 10.1109/TKDE.2008.239.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. *CoRR*, *abs/1703.06870*.

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III* (pp. 346–361). Cham: Springer International Publishing. doi: 10.1007/978-3-319-10578-9_23.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, *abs/1502.01852*.

van der Heijden, G., Song, Y., Horgan, G., Polder, G., Dieleman, A., Bink, M., Palloix, A., van Eeuwijk, F., & Glasbey, C. (2012). Spicy: towards automated phenotyping of large pepper plants in the greenhouse. *Functional Plant Biology*, *39*, 870–877.

Hemming, J., & Rath, T. (2001). Pa—precision agriculture: Computer-vision-based weed identification under field conditions using controlled lighting. *Journal of Agricultural Engineering Research*, *78*, 233 – 243. doi: https://doi.org/10.1006/jaer.2000.0639.

Henten, E. V., Slot, D. V., Hol, C., & Willigenburg, L. V. (2009). Optimal manipulator design for a cucumber harvesting robot. *Computers and Electronics in Agriculture*, *65*, 247 – 257. doi: https://doi.org/10.1016/j.compag.2008.11.004.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, .

Johnson, M., Shotton, J., & Cipolla, R. (2013). Semantic texton forests for image categorization and segmentation. In A. Criminisi, & J. Shotton (Eds.), *Decision Forests for Computer Vision and Medical Image Analysis* (pp. 211–227). London: Springer London. doi: 10.1007/978-1-4471-4929-3_15.

Kavasidis, I., Palazzo, S., Salvo, R. D., Giordano, D., & Spampinato, C. (2014). An innovative web-based collaborative platform for video annotation. *Multimedia Tools and Applications*, *70*, 413–432. doi: 10.1007/s11042-013-1419-7.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Kondaveeti, H. K. (2016). Synthetic isar images of aircrafts. url: `http://dx.doi.org/10.5281/zenodo.48002`. doi: 10.5281/zenodo.48002.

Krähenbühl, P., & Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*.

Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., & Hua, G. (2016). Labeled faces in the wild: A survey. In M. Kawulok, M. E. Celebi, & B. Smolka (Eds.), *Advances in Face Detection and Facial Image Analysis* (pp. 189–248). Cham: Springer International Publishing. doi: 10.1007/978-3-319-25958-1_8.

Li, Y., Cao, Z., Lu, H., Xiao, Y., Zhu, Y., & Cremers, A. B. (2016). In-field cotton detection via region-based semantic image segmentation. *Computers and Electronics in Agriculture*, *127*, 475 – 486. doi: https://doi.org/10.1016/j.compag.2016.07.006.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lu, J., Hu, J., Zhao, G., Mei, F., & Zhang, C. (2017). An in-field automatic wheat disease diagnosis system. *Computers and Electronics in Agriculture*, *142*, 369 – 379. doi: https://doi.org/10.1016/j.compag.2017.09.012.

Mallat, S. (1999). A wavelet tour of signal processing. San Diego: Academic Press. (2nd ed.).

Milioto, A., Lottes, P., & Stachniss, C. (2017a). Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *IV-2/W3*, 41–48. doi: 10.5194/isprs-annals-IV-2-W3-41-2017.

Milioto, A., Lottes, P., & Stachniss, C. (2017b). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns, .

Mishkin, D., & Matas, J. (2015). All you need is a good init. *CoRR*, *abs/1511.06422*.

Mostajabi, M., Yadollahpour, P., & Shakhnarovich, G. (2014). Feedforward semantic segmentation with zoom-out features. *CoRR*, *abs/1412.0774*.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*, 1. doi: 10.1186/s40537-014-0007-7.

Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Hočevar, M., Baur, J., Pfaff, J., & Ulbrich, H. (2013). Selective spraying of grapevine's diseases by a modular agricultural robot, . *44*, 149 – 153.

Papandreou, G., Chen, L.-C., Murphy, K., & Yuille, A. L. (2015). Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*.

Phadikar, S., & Sil, J. (2008). Rice disease identification using pattern recognition techniques. In *2008 11th International Conference on Computer and Information Technology* (pp. 420–423). doi: 10.1109/ICCITECHN.2008.4803079.

Polder, G., van der Heijden, G. W., van Doorn, J., & Baltissen, T. A. (2014). Automatic detection of tulip breaking virus (tbv) in tulip fields using machine vision. *Biosystems Engineering*, *117*, 35 – 42. doi: http://dx.doi.org/10.1016/j.biosystemseng.2013.05.010. Image Analysis in Agriculture.

Pont-Tuset, J., Arbeláez, P., Barron, J., Marques, F., & Malik, J. (2015). Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*.

Pound, M. P., Atkinson, J. A., Wells, D. M., Pridmore, T. P., & French, A. P. (2017). Deep learning for multi-task plant phenotyping. *bioRxiv*, . doi: 10.1101/204552. arXiv:https://www.biorxiv.org/content/early/2017/10/17/204552.full.pdf.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. (2016). The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*, 157–173. doi: 10.1007/s11263-007-0090-8.

Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., & Perez, T. (2017). Peduncle detection of sweet pepper for autonomous crop harvesting combined color and 3-d information. *IEEE Robotics and Automation Letters*, *2*, 765–772. doi: 10.1109/LRA.2017.2651952.

Shapiro, D. (2016). Accelerating the race to autonomous cars. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '16 (pp. 415–415). New York, NY, USA: ACM. doi: 10.1145/2939672.2945360.

Shelhamer, E., Long, J., & Darrell, T. (2016). Fully convolutional networks for semantic segmentation. *CoRR*, *abs/1605.06211*.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., & Blake, A. (2013). Efficient human pose estimation from single depth images. In A. Criminisi, & J. Shotton (Eds.), *Decision Forests for Computer Vision and Medical Image Analysis* (pp. 175–192). London: Springer London. doi: 10.1007/978-1-4471-4929-3_13.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., & Webb, R. (2016). Learning from simulated and unsupervised images through adversarial training. *CoRR*, *abs/1612.07828*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, *abs/1409.1556*.

Soekhoe, D., van der Putten, P., & Plaat, A. (2016). On the impact of data set size in transfer learning using deep neural networks. In H. Boström, A. Knobbe, C. Soares, & P. Papapetrou (Eds.), *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings* (pp. 50–60). Cham: Springer International Publishing. doi: 10.1007/978-3-319-46349-0_5.

de Soto, M. G., Emmi, L., Perez-Ruiz, M., Aguera, J., & de Santos, P. G. (2016). Autonomous systems for precise spraying – evaluation of a robotised patch sprayer. *Biosystems Engineering*, *146*, 165 – 182. doi: https://doi.org/10.1016/j.biosystemseng.2015.12.018. Special Issue: Advances in Robotic Agriculture for Crops.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Tsogkas, S., Kokkinos, I., Papandreou, G., & Vedaldi, A. (2015). Semantic part segmentation with deep learning. *CoRR*, *abs/1505.02438*.

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., & Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, *104*, 154–171.

Vijayarangan, S., Sodhi, P., Kini, P., Bourne, J., Du, S., Sun, H., Poczos, B., Apostolopoulos, D. D., & Wettergreen, D. (2017). High-throughput robotic phenotyping of energy sorghum crops. In *Field and Service Robotics*. Springer-Verlag.

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). *The Caltech-UCSD Birds-200-2011 Dataset*. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, J., & Yuille, A. L. (2015). Semantic part segmentation using compositional model combining shape and appearance. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, L., Shi, J., Song, G., & Shen, I.-f. (2007). Object detection combining recognition and segmentation. In Y. Yagi, S. B. Kang, I. S. Kweon, & H. Zha (Eds.), *Computer Vision – ACCV 2007: 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18-22, 2007, Proceedings, Part I* (pp. 189–199). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-76386-4_17.

Wehrens, R. (2010). Self-organising maps for image segmentation. In A. Fink, B. Lausen, W. Seidel, & A. Ultsch (Eds.), *Advances in Data Analysis, Data Handling and Business Intelligence: Proceedings of the 32nd Annual Conference of the Gesellschaft für Klassifikation e.V., Joint Conference with the British Classification Society (BCS) and the Dutch/Flemish Classification Society (VOC), Helmut-Schmidt-University, Hamburg, July 16-18, 2008* (pp. 373–383). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-01044-6_34.

Wu, Z., Shen, C., & van den Hengel, A. (2016). Wider or deeper: Revisiting the resnet model for visual recognition. *CoRR*, *abs/1611.10080*.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I* (pp. 818–833). Cham: Springer International Publishing. doi: 10.1007/978-3-319-10590-1_53.

Zeng, A., Yu, K., Song, S., Suo, D., Jr., E. W., Rodriguez, A., & Xiao, J. (2016). Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. *CoRR*, *abs/1609.09475*.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2016). Pyramid Scene Parsing Network. *ArXiv e-prints*, . arXiv:1612.01105.

Zhu, H., Meng, F., Cai, J., & Lu, S. (2016). Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, *34*, 12 – 27. doi: http://dx.doi.org/10.1016/j.jvcir.2015.10.012.

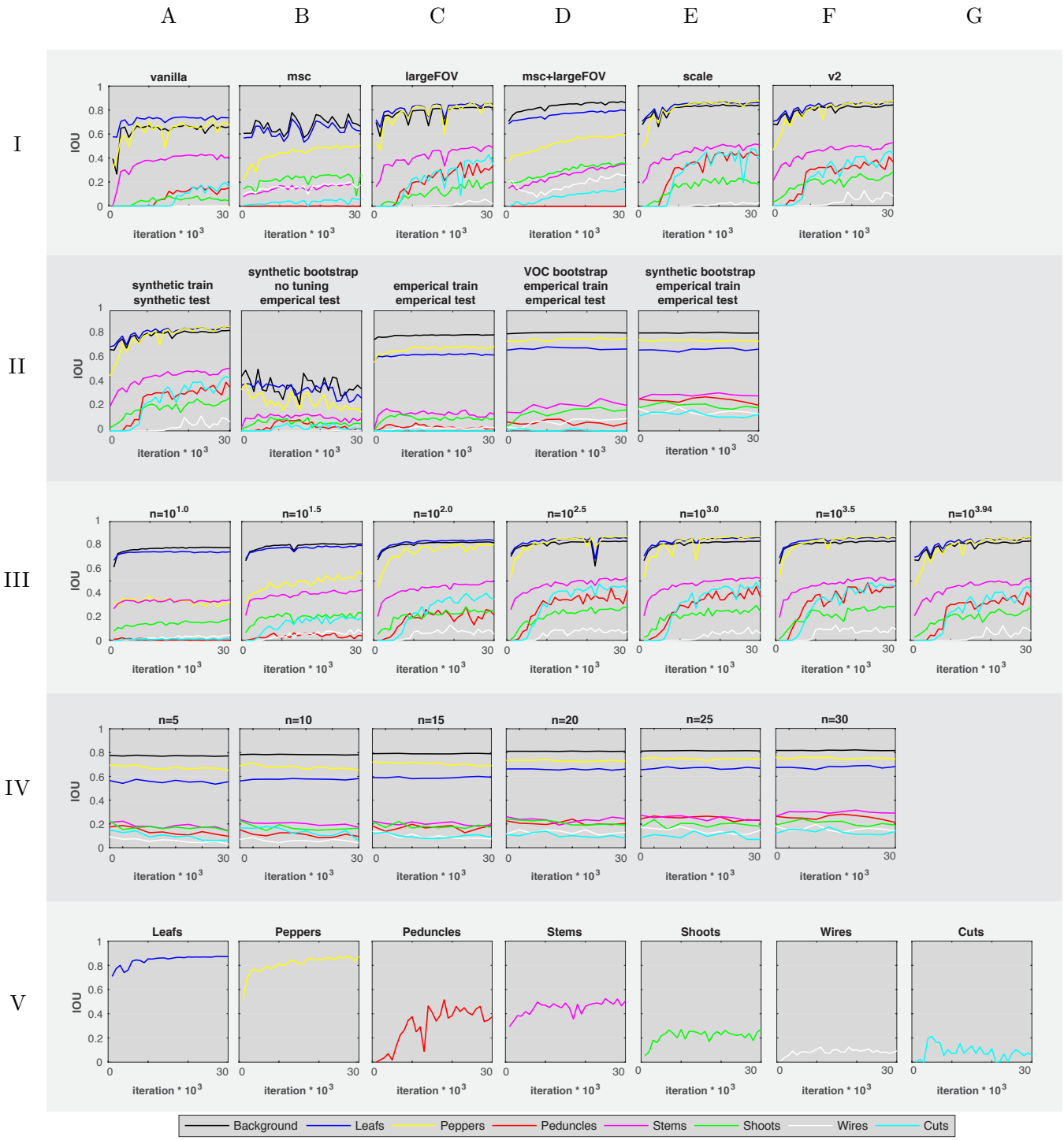
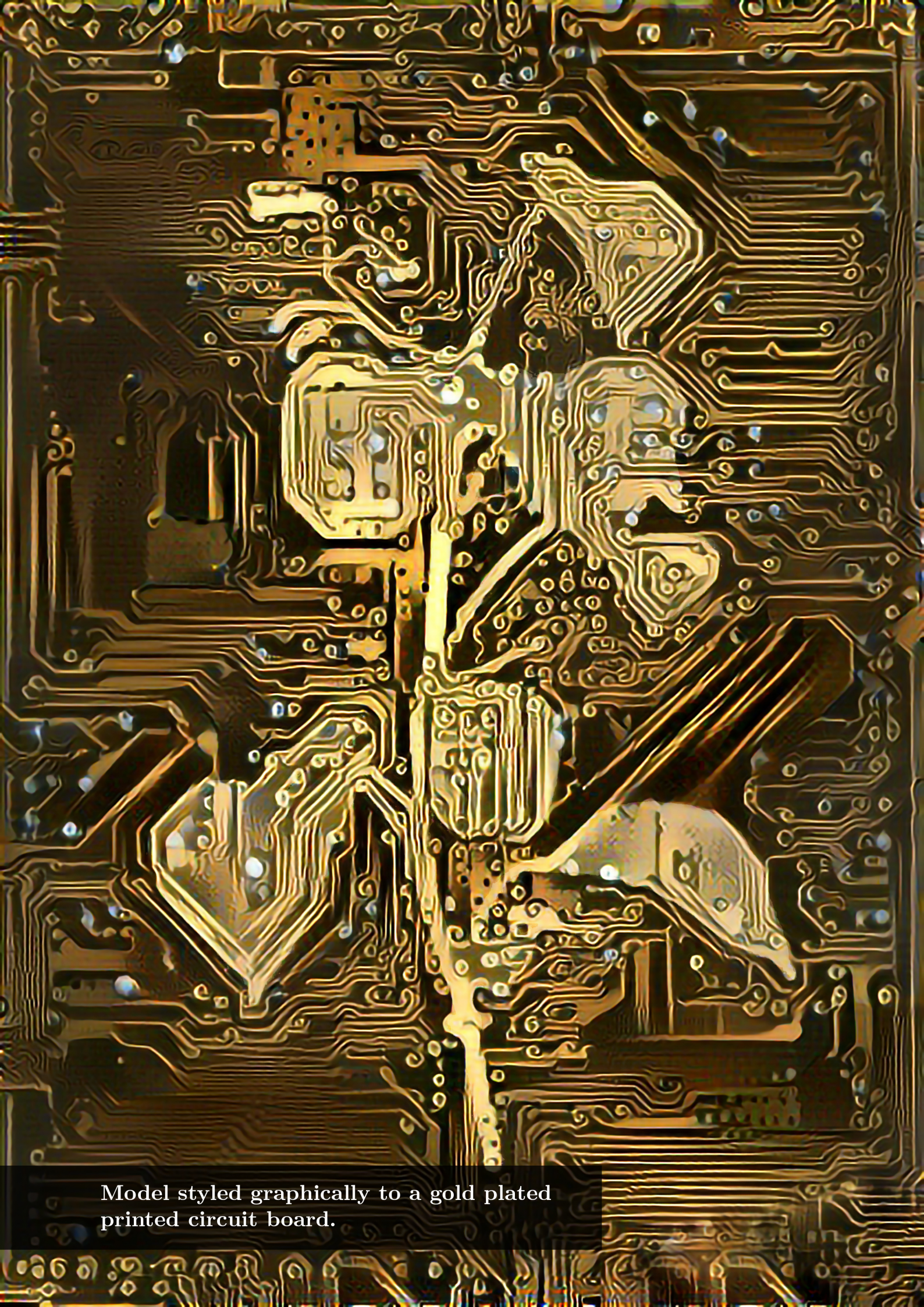


Figure 4.14: Detailed overview of IOU results of Experiments I through VII. For each model within those experiments A through F, the mean test set IOU, split by class, over the number of training iterations is displayed. Hence, figures show performance per class over time and the spread between all classes.



Model styled graphically to a gold plated printed circuit board.

Chapter 5

Improved Segmentation Performance by Optimising Realism of Synthetic Images using Cycle Generative Adversarial Networks.

This chapter is based on:

Barth, R., Hemming, J., and van Henten, E.J. van (2017). Optimising Realism of Synthetic Images using Cycle Generative Adversarial Networks for Improved Part Segmentation in Robotic Vision. *Submitted to the International Journal of Computer Vision: Special Issue on Deep Learning for Robotics.*

Abstract

In this chapter we report on improved plant part image segmentation performance using convolutional neural networks. This was achieved by optimising the visual realism of synthetic agricultural images. In Part I, a cycle consistent generative adversarial network was applied to synthetic and empirical images with the objective to generate more realistic synthetic images by translating them to the empirical domain. We hypothesise and showed that plant part image features such as color and texture become more similar to the empirical domain post translation. Results confirm this with an improved mean color distribution correlation with the empirical data prior (0.62) and post translation (0.90). Furthermore, the mean image features of contrast, homogeneity, energy and entropy moved closer to the empirical mean post translation. In Part II, 7 experiments were performed using convolutional neural networks with different combinations of synthetic, synthetic translated to empirical and empirical images. We hypothesised that the translated images can be used for (i) improved learning of empirical images and (ii) that learning without any fine-tuning with empirical images is improved by bootstrapping with translated images over bootstrapping with synthetic images. Results confirm both hypotheses. First a maximum intersection-over-union performance was achieved of 0.52 when bootstrapping with translated images and fine-tuning with empirical images; an 8% increase compared to only using synthetic images. Second, training without any empirical fine-tuning resulted in an average IOU of 0.31; a 55% performance increase over previous methods that only used synthetic images. The work presented in this chapter can be seen as an important step towards improved sensing for agricultural robotics, less reliant on annotated images.

5.1 Introduction

A key success factor of agricultural robotics performance is a robust underlying perception methodology that can distinguish and localise object parts (Bac et al., 2013; Gongal et al., 2015; Bac et al., 2014). In order to train state-of-the-art machine learning methods that can achieve this feat, large annotated empirical image datasets remain required. Synthetic images can help bootstrapping such methods in order to reduce the required amount of annotated empirical data (Barth et al., 2017b). However, a gap in realism remains between the modelled synthetic images and the empirical ones, plausibly restraining synthetic bootstrapping performance.

The long term objective of our research is to improve plant part segmentation performance. Previous work performed synthetically bootstrapping deep convolutional neural networks (CNN) (Barth et al., 2017b). In this chapter we report on optimising the realism of rendered synthetic images modelled from empirical photographical data (Barth et al., 2017a) that was used in our previous work. We hypothesise that the dissimilarity between synthetic and empirical images can be qualitatively and quantitatively reduced using unpaired image-to-image translation by cycle-consistent adversarial networks (Cycle-GAN) (Zhu et al., 2017). Furthermore, we hypothesise that the synthetic images translated to the empirical domain can be used for improved learning, potentially further closing the performance gap that remained previously when bootstrapping with only synthetic data (Barth et al., 2017b). Additionally, we hypothesise that without any empirical fine-tuning, improved empirical learning can be achieved using only translated images as opposed to using only synthetic images.

The key contributions presented in this chapter are the (i) further minimisation of the dependency on annotated empirical data for image segmentation learning and (ii) improving the performance thereof. This can be seen as an important step towards improved sensing for agricultural robotics.

5.1.1 Theoretical background

Convolutional neural networks recently have shown state-of-the-art performance on many image segmentation tasks (Chen et al., 2017; Long et al., 2015; Chen et al., 2015a). However, CNNs require large annotated datasets on a per-pixel level in order to successfully train the large number of free parameters of the deep network. Moreover, in agriculture the high amount of image variety due to a wide range of species, illumination conditions and morphological seasonal growth differences, leads to an increased annotated dataset size dependency. Satisfying this requirement can quickly become a bottleneck for learning.

One solution is to bootstrap CNNs with synthetic images including automatically computed ground truths (Dittrich et al., 2014; Ros et al., 2016). Consequently, the bootstrapped network can be fine-tuned with a small set of empirical images, which can result in increased performance over methods without synthetic bootstrapping (Barth et al., 2017b).

Previously we have shown methods to create such a synthetic dataset by realistically rendering 3D modelled plants (Barth et al., 2017a). Despite intensive manual optimisation for geometry, color and textures, we have shown that a discrepancy remains between the synthetic and empirical images. Although this dataset can be used for successful synthetic bootstrapping and improved empirical learning, there remained a difference between the achieved performance and the theoretical optimal performance (Barth et al., 2017b).

Recently, the advent of generative adversarial networks (GAN) introduced another method of image data generation (Goodfellow et al., 2014). In GANs two deep convolutional neural networks are trained simultaneously and adversarially: a generative model G and a discriminative model D. The generative model's goal is to capture the feature distribution of a dataset by learning to generate images thereof from latent variables (e.g. random noise vectors). The discriminative model in turn evaluates to what extent the generated image is a true member of the dataset. In other words, model G is optimised to trick model D while model D is optimising to not get fooled by model G. In Figure 5.1 a schematic overview of this learning process is shown. As both models can be implemented as CNNs, the error can be back-propagated to minimise the loss of both models simultaneously. The result after training is a model G that can generate new random images highly similar to the learned dataset. This method is useful if one wants to generate more similar images from the same domain. Given that this does not provide a corresponding ground truth, this method was not pursued for this chapter.

In later approaches, GANs were conditioned with an additional input image from another domain (Isola et al., 2016), forming an image pair that had some relation with each other (e.g. a color image and its label or class mapping). The generator was tasked with image-to-image translation to create an coherent image (e.g. color) from a corresponding pair image (e.g. label map). The discriminator's goal is then to evaluate if input pairs are either real or generated. The loss can then be fed back to both the discriminator and generator to improve on their tasks. The result after training is a generator G that can translate images from one domain X (e.g. color images) to images in another Y (e.g. label maps) or more formally notated as $G : X \rightarrow Y$. In Figure 5.2 a schematic overview of the learning process is shown. Given that this does not provide additional novel training pairs, we did not pursue this method for this chapter.

A requirement for image-to-image translation using conditional GANs is a large set of which images in both domains are paired geometrically. For our objective of translating images from the synthetic domain to the empirical domain, this requirement was not met because images from both domains did not geometrically correspond one-to-one.

A recent approach aimed to dissolve this paired requirement by investigating unpaired image-to-image translation (Zhu et al., 2017). In cycle-consistent adversarial networks (Cycle-GAN), a mapping $G : X \rightarrow Y$ is learned whilst also an inverse mapping $F : Y \rightarrow X$. Both domains X and Y have corresponding discriminators D_X and D_Y. Hence, D_X ensures G to translate X similar to Y whilst D_Y safeguards an indistinguishable conversion of Y to X.

However since the domains are unpaired, the translation at this point does not guarantee that an individual image $x \in X$ is mapped to an geometrically similar image in domain Y (or vice versa $y \in Y$ to X). This is because there are boundless mappings from x that result in the same target distribution of Y. Therefore the mapping needs to be constrained in a way the original geometry is maintained.

To achieve that, a cycle consistency loss was added to further regularise the learning. Given a sample $x \in X$ and $y \in Y$, a loss was added to the optimisation such that $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Hence, the learning was therefore constrained by the intuition that if an input image is translated from one domain to the other and then back again, an image should be retrieved similar to the original input. This similarity is captured by the cycle consistency loss, which forces the generators G and F to achieve unpaired geometrically consistent image-to-image translation from one domain to the other and vice versa.

In Figure 5.3 a schematic is show of this learning process. Note that this method was pursued for this chapter, because it allows the creation of a large dataset of images in the empirical domain. Furthermore, the key utility lies in the image pair P, in which the ground truth class mapping from the synthetic images could also be used for the synthetic translated to empirical images.

The chapter is structured in 2 parts, each with their corresponding materials, methods, discussion and conclusion sections. Part I describes the image-to-image translation from the synthetic rendered domain to the empirical photographic domain. In Part II, the effect on segmentation learning was investigated using the translated images.

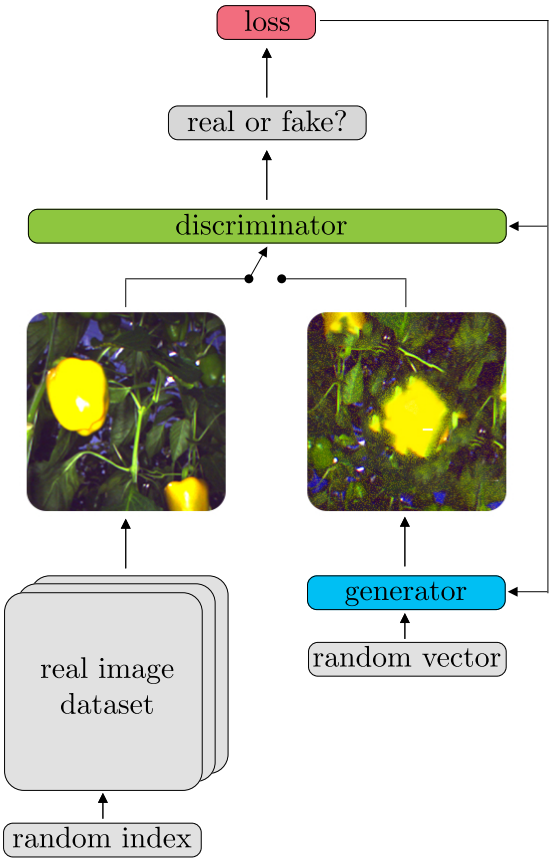


Figure 5.1: Learning schematic of a generative adversarial network. In each learning step, a discriminator receives either a random real domain image from a database or a generated image from the generator. The discriminator determines if this image is real or generated. Through a loss function, feedback to both the discriminator and the generator is given to optimise their tasks. In this example, the generator learns to synthesise empirical photographic images from random vectors. Note that this example was not pursued for this chapter.

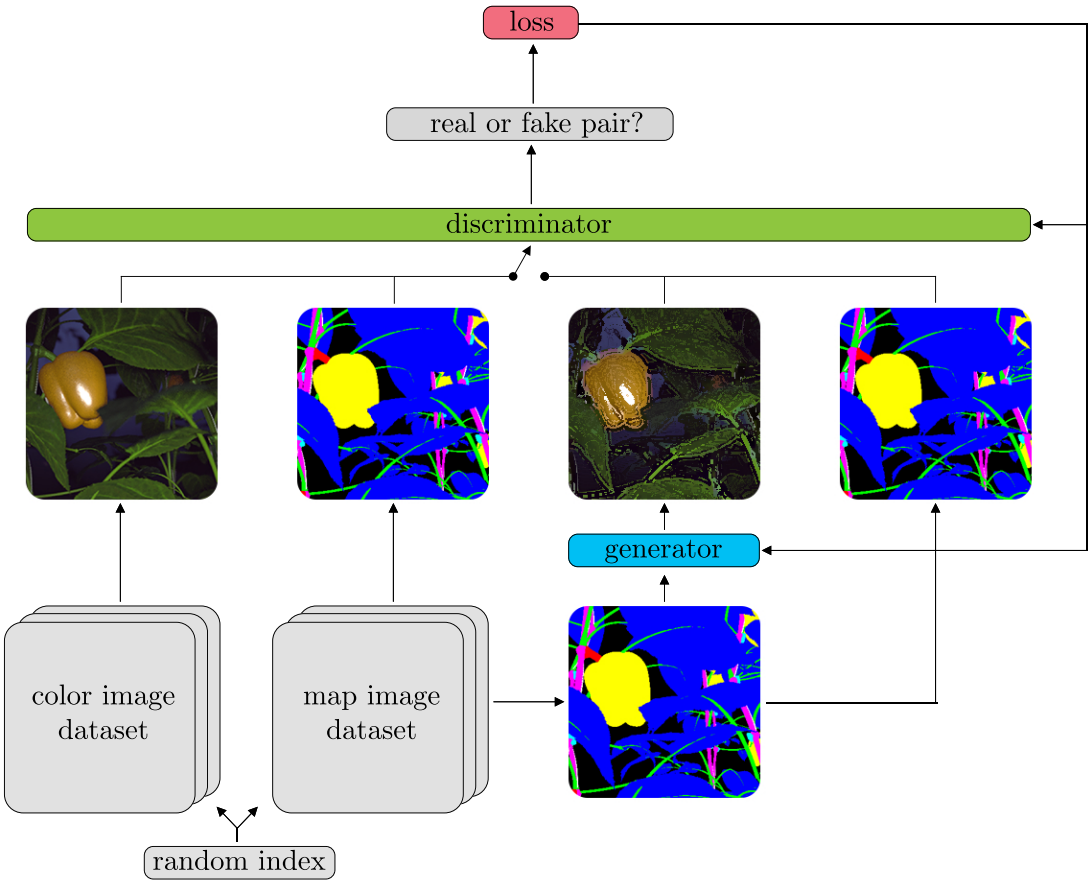


Figure 5.2: Learning schematic of a conditional generative adversarial network. In each learning step, a discriminator receives either a real pair of corresponding images in different domains (e.g. colored 3D render and a map) or a fake pair of which one domain image was synthesized by the generator (e.g. colored 3D render) from an image in the other domain (e.g. map). The discriminator determines if this image pair is real or fake. Through a loss function feedback to both the discriminator and the generator is given to optimise their tasks. In this example, the generator learns to synthesise render-like color images from class maps. Note that this example was not pursued for this chapter.

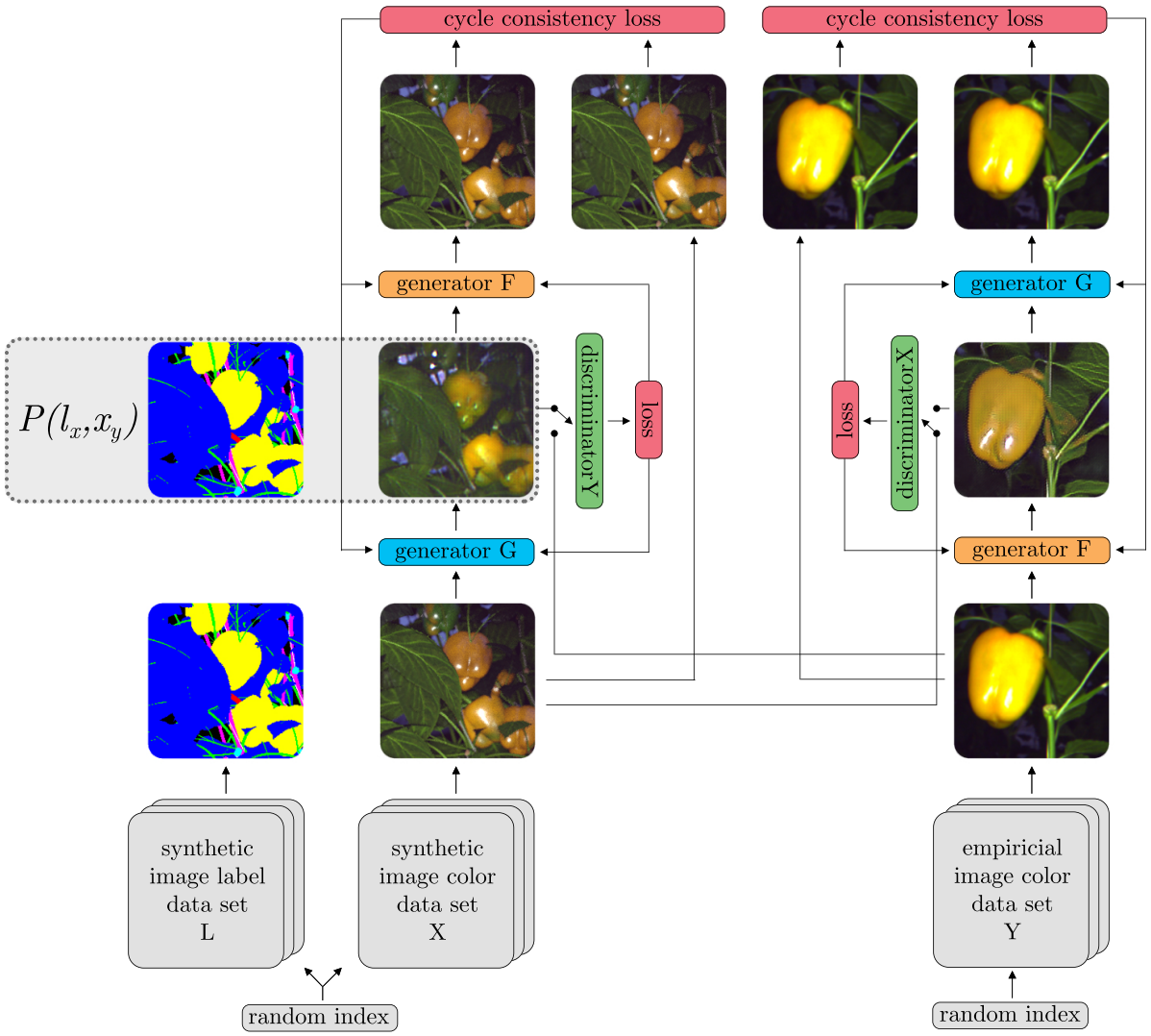


Figure 5.3: Learning schematic of a cycle generative adversarial network. In each learning step, generator G receives an image from domain X and generator F receives an image from domain Y. Each generator is trained to transform the input image to the other domain. A discriminator Y and discriminator X for each corresponding domain is trained to distinguish between generated and original domain images. From those first set of generated images, the opposing generator then synthesizes the second set of images back to its original domain (which ideally should result in the original domain image). A cycle consistency loss is then calculated by comparing the second set of images with the initial input image. The loss of both discriminators and cycle consistency is fed back to both generators for learning. In this example, each generator learns to synthesise an image to the opposing domain, whilst remaining geometrically consistent. This example was pursued in this chapter to obtain image pair P, consisting of the label l_x that corresponds to x_y; the translated image from domain X to Y.

5.2 Part I: Image-to-image translation

In this first part of the chapter we describe and evaluate the unpaired image-to-image translation on agricultural images from the synthetic to the empirical domain and vice versa. The main objective was to obtain pairs of images P consisting of a synthetic to empirical domain translated image and corresponding ground truth map (see Figure 5.3).

5.2.1 Materials

Image dataset

The unpaired image dataset of *Capsicum annuum* (sweet- or bell pepper) was used (Barth et al., 2017a) that consists of 50 empirical images of a crop in a commercial high-tech greenhouse and 10,500 corresponding synthetic images, modelled to approximate the empirical set visually. In both sets, 8 classes were annotated on a per-pixel level, either manually for the empirical dataset or automatically computed for the synthetic dataset. In Figure 5.4 examples of images in the dataset are shown. The dataset was publicly released at: `http://dx.doi.org/10.4121/uuid:884958f5-b868-46e1-b3d8-a0b5d91b02c0`

Both synthetic and empirical images were first cropped to 424x424 pixels to exclude the robot end-effector's suction cup in the image, because initial image-to-image translation experiments showed the cup was replicated undesirably in other parts of the image. This was in line with previous findings from the original authors where color and texture translation often succeeded but large geometric changes were translated with less success (Zhu et al., 2017). Secondly, the resolution was resampled bilinearly to 1000x1000 pixels as additional experiments during Part II showed upscaling improved the learning.

From the *Capsicum annuum* dataset, the synthetic images 1-1000 were used for translation training and the remainder for testing. For the empirical images, 50 annotated images of the *Capsicum annuum* dataset were used for testing, whereas for training 175 non-annotated images were used that were not part in the released dataset, but were collected during the same data acquisition.

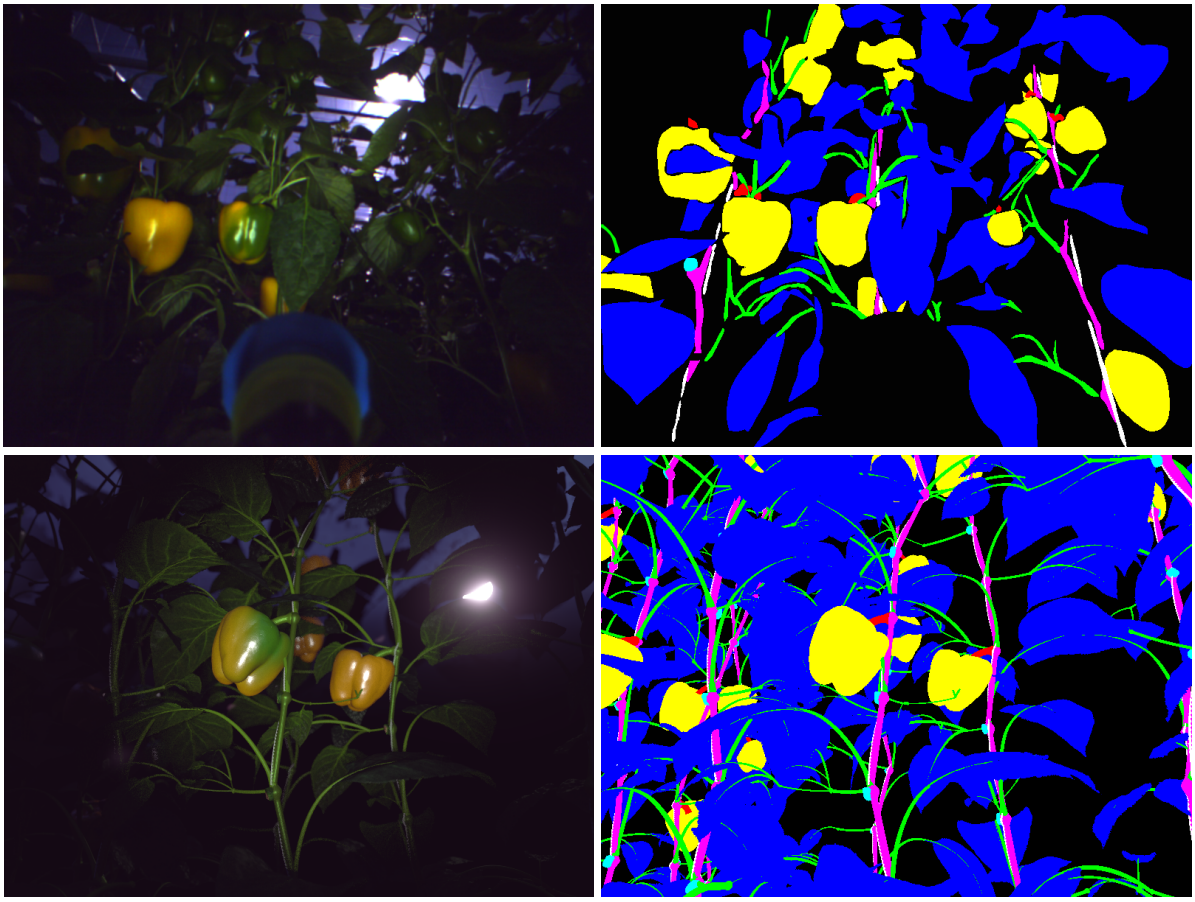


Figure 5.4: Uncropped examples of empirical (top row) and synthetic (bottom row) color images (left column) and their corresponding ground truth labels (right column). Class labels: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts where pepper where harvested.

Software

The Berkeley AI Research (BAIR) laboratory implementation of unpaired image-to-image translation using cycle-consistent adversarial networks was used (Zhu et al., 2017).

Hardware

Experiments were run on a NVIDIA DevBox system with 4 TITAN X Maxwell 12GB GPUs, Intel Core i7-5930K and 128GB DDR4 RAM running Ubuntu 14.04.

5.2.2 Methods

The adversarial learning scheme in Figure 5.3 was applied with synthetic images as domain X and empirical images as domain Y. The hyper-parameters of the Cycle-GAN were manually optimised by visually evaluating the resulting images with their target domain. The number of generative and discriminative filters were set to 50 and the learning rate was set to 0.0002 with an ADAM (Kingma & Ba, 2014) momentum term of 0.5. The basic discriminator model was used, whereas for the generator the RESNET 6 blocks model (He et al., 2015a). Weights for the cycle loss were set to 10 for each direction.

Quantitative translation evaluation

Although the success of the translation will already be quantitatively captured by the adversarial loss, this measure is biased and mathematically obfuscated. By specifically looking at key image features like color, contrast, homogeneity, energy and entropy, it could be derived if the translated images improved on those features. This would provide evidence to what extent the dissimilarity gap between the synthetic and empirical domains were closed further.

For this purpose, we first compared for each class the synthetic color distribution prior and post translation with those of the empirical distribution. The color spectrum of each class was obtained by first transforming the color images to HSI colorspace. The hue channel in the transformed image represented for each pixel which color was present, irregardless of illumination and saturation intensity. The histogram of this channel was then taken to count the relative color occurrence per class.

As we hypothesise that the color difference post translation will be reduced, the average correlation of each class between the synthetic and empirical color distributions was compared to the average correlation of each class between the synthetic translated to empirical and the empirical distributions.

Second, to obtain additional image features, first an average gray level co-occurrence matrix (GLCM) (Haralick et al., 1973)was calculated for each class for the first 10 images in the synthetic, synthetic translated to empirical and empirical sets. The GLCM summarises how often a pixel with a certain intensity value i occurred in a specific spatial relationship to a pixel with the intensity value of j. This relationship was set to horizontally neighbouring pixels.

From the GLCM, the following features were derived. Contrast; measuring the overall difference in luminance between neighbouring pixels (Equation 5.1). Homogeneity; a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal, which implies that high values of homogeneity reflect the absence of changes in the image and indicates a locally homogenous distribution in image textures (Equation 5.2). Energy; a measure of texture crudeness or disorder (Equation 5.3). Entropy; measuring the amount of information or complexity in the image (Equation 5.4).

$$contrast = \sum_{i,j} |i-j|^2 \, GLCM(i,j) \tag{5.1}$$

$$homogeneity = \sum_{i,j} \frac{GLCM(i,j)}{1+|i-j|} \tag{5.2}$$

$$energy = \sum_{i,j} GLCM(i,j)^2 \tag{5.3}$$

$$entropy = \sum_{i,j} -ln(GLCM(i,j)) \cdot GLCM(i,j) \tag{5.4}$$

5.2.3 Results

	backgr.	leafs	peppers	peduncles	stems	shoots	wires	cuts	mean
correlation(synthetic, empirical)	0.25	0.78	0.42	0.93	0.76	0.83	0.45	0.48	**0.62**
correlation(synthetic→empirical, empirical)	0.86	0.94	0.93	0.93	0.92	0.98	0.81	0.79	**0.90**

Table 5.1: Color distribution correlation per class and average between the synthetic and synthetic translated to empirical with the empirical image dataset.

In Figure 5.6 the results of the image-to-image translations are shown. The second column is of most interest to our research, as it shows the set X_y of synthetic images which were translated to the empirical domain. However, as a reference also the translation from empirical to the synthetic domain is shown in the third column.

The color distributions for each class for the synthetic, empirical and synthetic to empirical translated images are shown in Figure 5.7. The corresponding correlations between the empirical images and the synthetic or synthetic to empirical images are shown in Table 5.1.

For the image features contrast, homogeneity, energy and entropy, the results per class for the synthetic, empirical and synthetic translated to empirical are shown in Figure 5.5. The difference of 0.100 was found in contrast averaged over all classes between the synthetic and empirical set, whereas this difference was reduced to 0.015 between the translated and the empirical set. Similarly, for homogeneity this was reduced from 0.028 to 0.015. For the energy feature, this was reduced from 0.126 to 0.026. Regarding entropy, the average difference was reduced from 0.364 to 0.003.

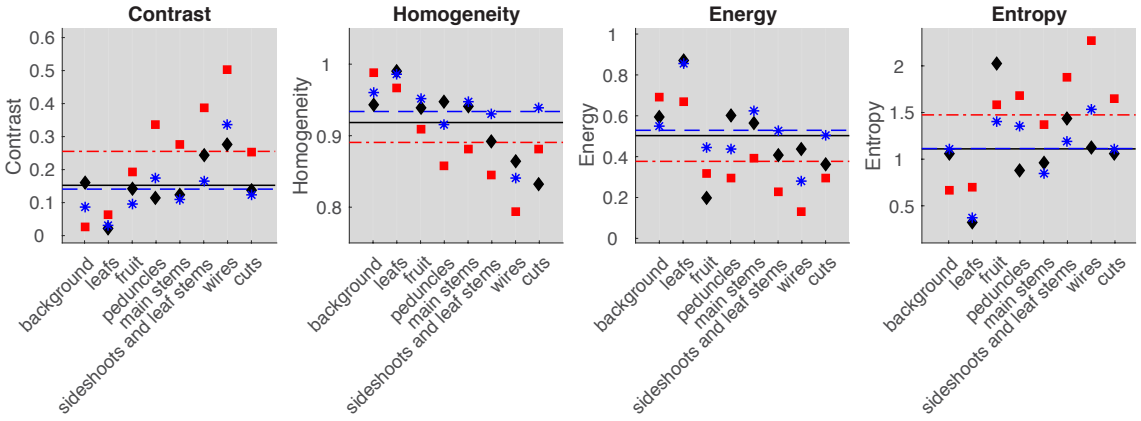


Figure 5.5: Image features values for contrast, homogeneity, erergy and entropy per class for the empirical ◆, synthetic ■ and synthetic translated to empirical *. Average over all classes is represented by a solid line for the empirical set, a dashed-dotted line for the synthetic set and a dashed line for the synthetic translated to empirical set.



Figure 5.6: Image-to-image translation examples using Cycle-GAN. Source domain images prior translation are shown in the outer columns; synthetic images (left) and empirical images (right). The second column shows set of interest X_y; the translated synthetic images to empirical ones. The third column shows empirical images translated to synthetic ones.

Figure 5.7: Color distributions discretized to 256 values in the hue channel (x-axis) per class of the synthetic, empirical and synthetic translated to empirical images. Integral per distribution amounts to 1 (y-axis).

5.2.4 Discussion and conclusion

Qualitative evaluation of the results showed a remarkable visual translation from synthetic images to highly empirical looking images and vice versa. Most notably the scattering of illumination and color of each plant part were converted realistically. It also appeared that the model learns to distinguish plant parts without any supervised information, as the (partially) ripe and unripe fruit are often translated to the other domain with altered maturity levels. A difference in camera focus seemed translated properly, indicating that local features (e.g. edge blur and texture) can be mapped accurately.

Some image artifacts do arise however, especially the translation to overexposed area's like sunshine or fruit reflections. The explanation might be that the model cannot generate this information correctly because any information beyond overexposure prior translation was already collapsed to a single maximum value (e.g. 255). Furthermore, an underlying checker-like texture seems to have been added to the translated local textures.

Larger morphological features (e.g. plant part shape and their geometry) were not translated, indicating a limitation of the Cycle-GAN approach. This suggests that the source synthetic data should be geometrically highly similar to the empirical situation for a realistic translation to succeed.

However, since geometry was not translated, this did allow for using the underlying synthetic ground truth labels to be used with the translated images for Part II. If also the geometry was translated, then the ground truth labels would not have been translated accordingly.

The method was not suited when one image set contains additional parts absent in the other set, e.g. the inclusion of a suction cup in our earlier experiments. We noticed in previous experiments that this part was undesirably replicated in other areas of the image.

Regarding our hypothesis, we confirm that image feature differences with the empirical set was reduced post translation. In Figure 5.7 the translation effect on color distribution can be seen for each plant part and background. Quantitatively, the mean correlation of the color distributions increased of the synthetic data with the empirical data prior (0.62) and post translation (0.90) (See Table 5.1). Furthermore, for the image features contrast, homogeneity, energy and entropy, the average difference with the empirical set after translation was reduced. For some individual classes this did not hold, e.g. the homogeneity of the cuts was erroneously doubled instead.

This part of the work contributed to the field of computer vision (e.g. for agricultural robotics) by providing a method for optimising realism in synthetic training data to potentially improve state-of-the-art machine learning methods that semantically segment plant parts, as evaluated in Part II of this chapter.

5.3 Part II: Improving semantic segmentation

In Part II, the effect on plant part segmentation learning by using the translated images from Part I instead of synthetic images was investigated. First, we hypothesise by bootstrapping with translated images and empirical fine-tuning, the highest empirical performance can be achieved over methods that bootstrap with limited dataset size of (30) empirical images or a large set (8750) of synthetic images. Secondly, we hypothesise that that without any empirical fine-tuning, learning can be improved with translated images as compared to using synthetic bootstrapping instead.

5.3.1 Materials

The synthetic and empirical datasets as described in Part I (see Section 5.2.1) were used as well as the obtained image pairs $P(l_x, x_y)$ (see Figure 5.2).

Software

The publicly available semantic segmentation framework DeepLab V2 was used, which implemented convolutional neural network (CNN) models (Papandreou et al., 2015; Chen et al., 2015b) on top of Caffe (Jia et al., 2014). Specifically, the VGG-16 network was used with a modification to include *à trous* spatial pyramid pooling for image context at multiple scales by convolutional feature layers with different fields-of-view (Chen et al., 2016; He et al., 2014).

Hardware

Experiments were run on the same hardware as used in Part I. As a dependancy for the DeepLab V2 Caffe version, the archived version of CUDA 7.5 was installed.

5.3.2 Methods

To compare performance differences, 7 experiments were preformed using different combinations of train, fine-tune and test sets. The motivation for each experiment is given below and the used sets and image ranges are shown between brackets.

A *Train: empirical (1-30). Test: empirical (41-50).*
As a reference to see if the model can learn using a small dataset, using only empirical data. The performance was expected to be low given the small dataset size.

B *Train: synthetic (1-8750). Test: synthetic (8851-8900).*
This experiment was run to obtain baseline performance of the model when having access to a large and detailed annotated dataset.

C *Train: synthetic (1-8750). Test: empirical (41-50).*
As a reference to see to what extent a synthetic trained network can generalise to the empirical domain, without empirical fine-tuning. Given the image dataset similarity gap, performance should be relatively low compared to a more realistic synthetic dataset (e.g. synthetic images transformed to the empirical domain in Experiment E).

D *Train: synthetic (1-8750). Fine-tune: empirical (1-30). Test: empirical (41-50).*
As a reference to see to what extent a synthetic trained network can generalise to the empirical domain, including empirical fine-tuning. In order to determine if the dataset translation from the synthetic domain to the empirical domain improves learning, this experiments acts as a reference performance of Experiment F.

E *Train: synthetic translated to empirical (1-8750). Test: synthetic translated to empirical (8851-8900).*
This experiment was run to obtain baseline performance of the model when having access to a large and detailed annotated dataset. The performance should be similar of Experiment A.

F *Train: synthetic translated to empirical (1-8750). Test: empirical (41-50).*
As a reference to see to what extent synthetic trained network with improved realism can generalise to the empirical domain, without fine-tuning with empirical images. This experiment should provide the main result for our second hypothesis.

G *Train: synthetic translated to empirical (1-8750). Fine-tune: empirical (1-30). Test: empirical (41-50).*
This experiment should provide the main result for our first hypothesis. Performance was expected to be highest amongst experiments testing on empirical data.

CNN Training

For each experiment, a convolutional neural network was trained and/or fine-tuned and tested according to the dataset scheme as described in Section 5.3.2. The hyperparameters of the network were manually optimised using separate validation datasets for combination of models and data set configurations as suggested by (Goodfellow et al., 2016; Bengio, 2012). This resulted in using Adaptive Moment Estimation (ADAM) (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ and a base learning rate of 0.001 for 30,000 iterations with a batch size of 4. These chosen hyper-parameters were found to be consistently optimal previously (Barth et al., 2017b) and therefore we fixed them across conditions. An adjustment was made in the layer weight initialisation procedure, by updating the model to using MSRA weight fillers (He et al., 2015b; Mishkin & Matas, 2015). Furthermore, the dropout rate (Srivastava et al., 2014) was adjusted to 0.50 to circumvent early overfitting and facilitate generalisation. The size of the input layer was cropped to 929x929 pixels, which was the maximum that our GPU memory would allow.

Performance Evaluation

To calculate the performance of the segmentation, we used the Jaccard Index similarity coefficient as an evaluation procedure, also known as the intersection-over-union (IOU) (He & Garcia, 2009) which is widely used for semantic segmentation evaluation (Gabriela Csurka, 2013; Everingham et al., 2010). The measure is defined in Equation 6.2, where the mean IOU over all classes equals the intersection of the segmentation and the ground truth divided by their union. A higher IOU implies more overlap, hence better performance. To derive the measure, a pixel-level confusion matrix C is calculated first for each image I in data set D :

$$C_{ij} = \sum_{I \in D} \left| \left\{ p \in I \mid S^I_{gt}(p) = i \wedge S^I_{ps}(p) = j \right\} \right|, \tag{5.5}$$

where $S^I_{gt}(p)$ is the ground truth label of pixel p in image I and $S^I_{ps}(p)$ is the predicted segmentation label. This implies that C_{ij} equal the number of predicted pixels i with label j. The average IOU over all classes L is given by:

$$IOU = \frac{1}{L} \sum_{i=1}^{L} \frac{C_{ii}}{G_i + P_i - C_{ii}} \text{ , where} \tag{5.6}$$

$$G_i = \sum_{j=1}^{L} C_{ij} \quad \text{and} \quad P_j = \sum_{i} C_{ij} \tag{5.7}$$

Hence G_i denotes the total number of pixels labeled with class i in the ground truth and P_j the total number of pixels with prediction j in the image.

5.3.3 Results

In Figure 5.9 the average IOU over all classes for Experiments A through G is shown, as well as previous results of similar experiments A-D (Barth et al., 2017b). In Figure 5.8 the performances were split over the classes. Qualitative results are presented in Figure 5.10.

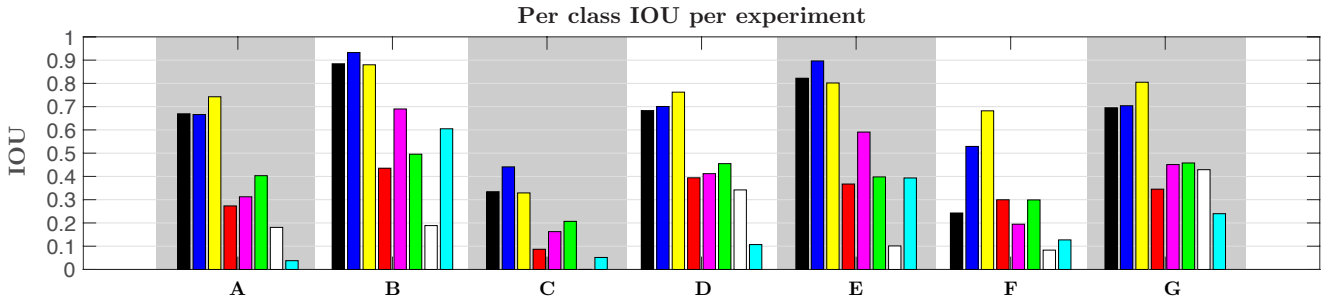


Figure 5.8: For Experiments A through G, the IOU per class is displayed, ordered as: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts where pepper where harvested.

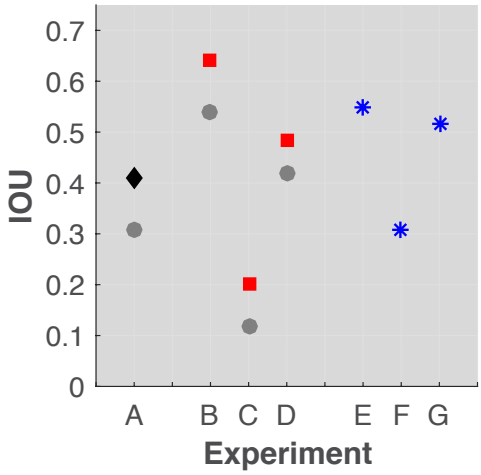
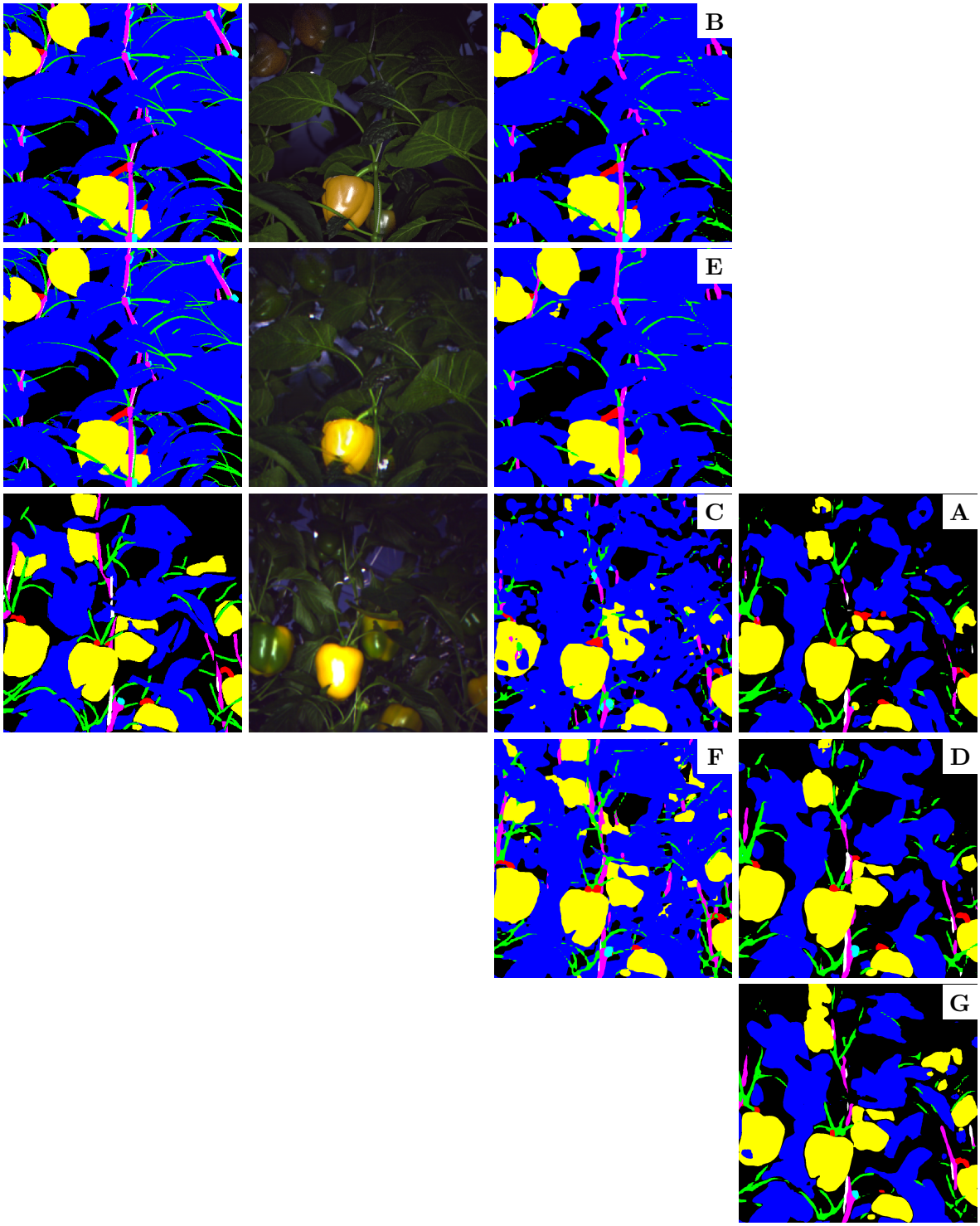


Figure 5.9: Average IOU over all classes for Experiment A with empirical training (◆), Experiments B,C and D with synthetic image bootstrapping (■) and Experiments E, F and G with synthetic translated to empirical image bootstrapping (*). Previous perfomance for similar experiments A-D are shown for reference (●).

Figure 5.10: Qualitative results from Experiments A through G. In the first column, the ground truths for synthetic (top), synthetic translated to empirical (middle) and empirical (bottom) are shown with labels: ● background, ● leafs, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ● cuts. In the second column color images are displayed. Experimental results are grouped in the third column (trained without empirical data) and fourth column (fine-tuned with empirical data).



5.3.4 Discussion and conclusion

Compared to the former attempts using the same dataset (Barth et al., 2017b), current results showed an overall improved performance of 0.08 IOU on average per Experiments A through D. The two differences implemented in our current attempt were first the cropping to 424x424 pixels to exclude the suction cup and then the upscaling to 1000x1000 pixels. The same CNN configuration was used. Additional experiments showed that the upscaling was the main cause of the performance increase. This might be explained by the CNN's larger field of view, allowing for detail only to dissolve by convolutions and pooling in deeper layers of the network.

In Experiment A, the CNN that trained and tested using only empirical images reached a performance of 0.41 average IOU. Looking at the per class performance distribution in Figure 5.8, we note that the *cut* class was barely recognised with an IOU of 0.04. Recognising all classes was previously considered as a requirement (Barth et al., 2017b). Therefore we concluded that training with empirical images alone did not suffice, although qualitative results (see Figure 5.10) looked promising and useful for some tasks like fruit detection.

The best performance of all experiments was achieved in Experiment B with an average IOU of 0.64, albeit on synthetic data only. This result was expected as the CNN had access to a large dataset with an exact ground truth. Furthermore, this performance could be used as a baseline to indicate the maximum obtainable IOU for this domain and currently used CNN architecture. The performance of the other experiments can be put into perspective of this IOU. Qualitative results still showed some gaps in thin and elongated classes like leaf stems and shoots, although improved over previous segmentations where such gaps were larger (Barth et al., 2017b).

Experiment C showed the performance on empirical images by bootstrapping with synthetic images but without fine-tuning with empirical images. With an average IOU of 0.20, the performance approximately doubled over previous results (Barth et al., 2017b). However, looking at the per class distribution classes like *peduncle* and *cut* were barely recognised (IOU<0.1) and the *wire* class was omitted all together. Qualitatively, results looked far from the ground truth. Looking at the qualitative results, we can conclude that without fine-tuning, training with synthetic data would not be sufficient for many tasks.

When a synthetically bootstrapped network from Experiment C was fine-tuned with empirical images in Experiment D, the IOU performance on empirical images was 0.48, as opposed to not bootstrapping in Experiment A this was an increase of 17%. We concluded that bootstrapping with synthetic images and fine-tuning with empirical images can be used to close the gap towards the optimal estimated possible performance of Experiment B. Furthermore, we noted that all classes were included, although the *cut* class was again barely recognised (IOU=0.11). Qualitatively, results looked close to the ground truth.

Similar to Experiment B, Experiment E trained and tested on a large dataset. Instead of synthetic images, the synthetic images translated to the empirical domain were used. The performance of 0.56 IOU was lower than Experiment B (IOU=0.64), although qualitative results looked comparable. The difference in performance might be explained by the introduced variance in image features like color and texture from the empirical domain, which could have made the classification more uncertain.

Experiment F evaluated on empirical images when trained on synthetic translated to empirical images, without fine-tuning with empirical images. Compared to Experiment C, where synthetic images were used instead of translated ones, the performance increased with 55% to an average IOU of 0.31. Although qualitatively also improvements could be observed over Experiment C, we noted from the class performance distribution there existed still a relative poor performance on classes *wires* and *cuts*.

By fine-tuning the model from F with empirical images in Experiment G, the best performance on empirical data was obtained (IOU=0.52); an increase of 27% over Experiment A and 8% over Experiment D. Qualitatively, results looked close to the ground truth and comparable to results of Experiments A and D. Looking at the class distribution, all classes were included. Most notably the *cut* class performance increased with 118% over Experiment D and with 600% over Experiment A to an IOU of 0.24.

Regarding our hypotheses, first we concluded from these results that synthetic images translated to the empirical domain can be used for improved learning (IOU=0.52) over training only with empirical (IOU=0.41) or synthetic data (IOU=0.48). With reference to our second hypothesis, we concluded that without any fine-tuning with empirical images, a CNN can be applied for improved learning of empirical data (IOU=0.31) with only synthetic translated to empirical images used for training, over methods that only used synthetic images (IOU=0.20).

5.4 General discussion and conclusion

In Part I, a cycle consistent generative adversarial network was applied to synthetic and empirical images with the objective to generate more realistic synthetic images by translating them to the empirical domain. Our analysis showed that the image feature distributions of these translated images, both in color and texture, were improved towards the empirical images. Regarding our hypothesis, it was confirmed that the image feature difference with the empirical set was reduced post translation. Qualitatively, the translated synthetic images looked highly similar to the real world situation. However, some translation artifacts remained. Furthermore the Cycle-GAN method could not improve upon geometric dissimilarities with the empirical domain. The latter proved an advantage however, as the synthetic ground truth also corresponded to the translated color images, allowing for the experiments on improved learning in the second part of our work.

In Part II, it was evaluated to what extent synthetic translated images to the empirical domain could improve on CNN learning with empirical images over other learning strategies. We confirmed our hypotheses that by using translated images and fine-tuning with empirical images, the highest performance for empirical images can be achieved (IOU=0.52) over training with only empirical (IOU=0.41) or synthetic data (IOU=0.48)

Besides improving performance on empirical images, another key contribution of our work is the further minimisation of the CNN dependency on annotated empirical data. We confirmed our hypothesis that without any empirical image fine-tuning, learning can be improved with translated images (IOU=0.31), a 55% increase over using just synthetic images (IOU=0.20).

The work presented in this chapter can be seen as an important step towards improved sensing for applied computer vision domains such as in agricultural robotics, medical support systems or autonomous navigation. It facilitates CNN semantic part segmentation learning without or minimal requirement of annotated images.

Acknowledgement

This research was partially funded by the European Commission in the Horizon2020 Programme (SWEEPER GA no. 644313) and the Dutch Ministry of Economic Affairs (EU140935).

References

Bac, C., Hemming, J., & van Henten, E. (2013). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture*, *96*, 148 – 162. doi: http://dx.doi.org/10.1016/j.compag.2013.05.004.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Barth, R., IJsselmuiden, J., Hemming, J., & van Henten, E. J. (2017a). Data synthesis methods for semantic segmentation in agriculture: a capsicum annuum dataset. *Computers and Electronics in Agriculture*, . doi: 10.1016/j.compag.2017.12.001.

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. V. (2017b). Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture*, . doi: https://doi.org/10.1016/j.compag.2017.11.040.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR*, *abs/1206.5533*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015a). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015b). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, .

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *ArXiv e-prints*, . `arXiv:1706.05587`.

Dittrich, F., Woern, H., Sharma, V., & Yayilgan, S. (2014). Pixelwise object class segmentation based on synthetic data using an optimized training strategy. In *Networks Soft Computing (ICNSC), 2014 First International Conference on* (pp. 388–394). doi: 10.1109/CNSC.2014.6906671.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Gabriela Csurka, F. P., Diane Larlus (2013). What is a good evaluation measure for semantic segmentation? In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, *116*, 8 – 19.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 2672–2680). Curran Associates, Inc.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*, 610–621. doi: 10.1109/TSMC.1973.4309314.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 1263–1284. doi: 10.1109/TKDE.2008.239.

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III* (pp. 346–361). Cham: Springer International Publishing. doi: 10.1007/978-3-319-10578-9_23.

He, K., Zhang, X., Ren, S., & Sun, J. (2015a). Deep residual learning for image recognition. *CoRR*, *abs/1512.03385*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015b). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, *abs/1502.01852*.

Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *CoRR*, *abs/1611.07004*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, .

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

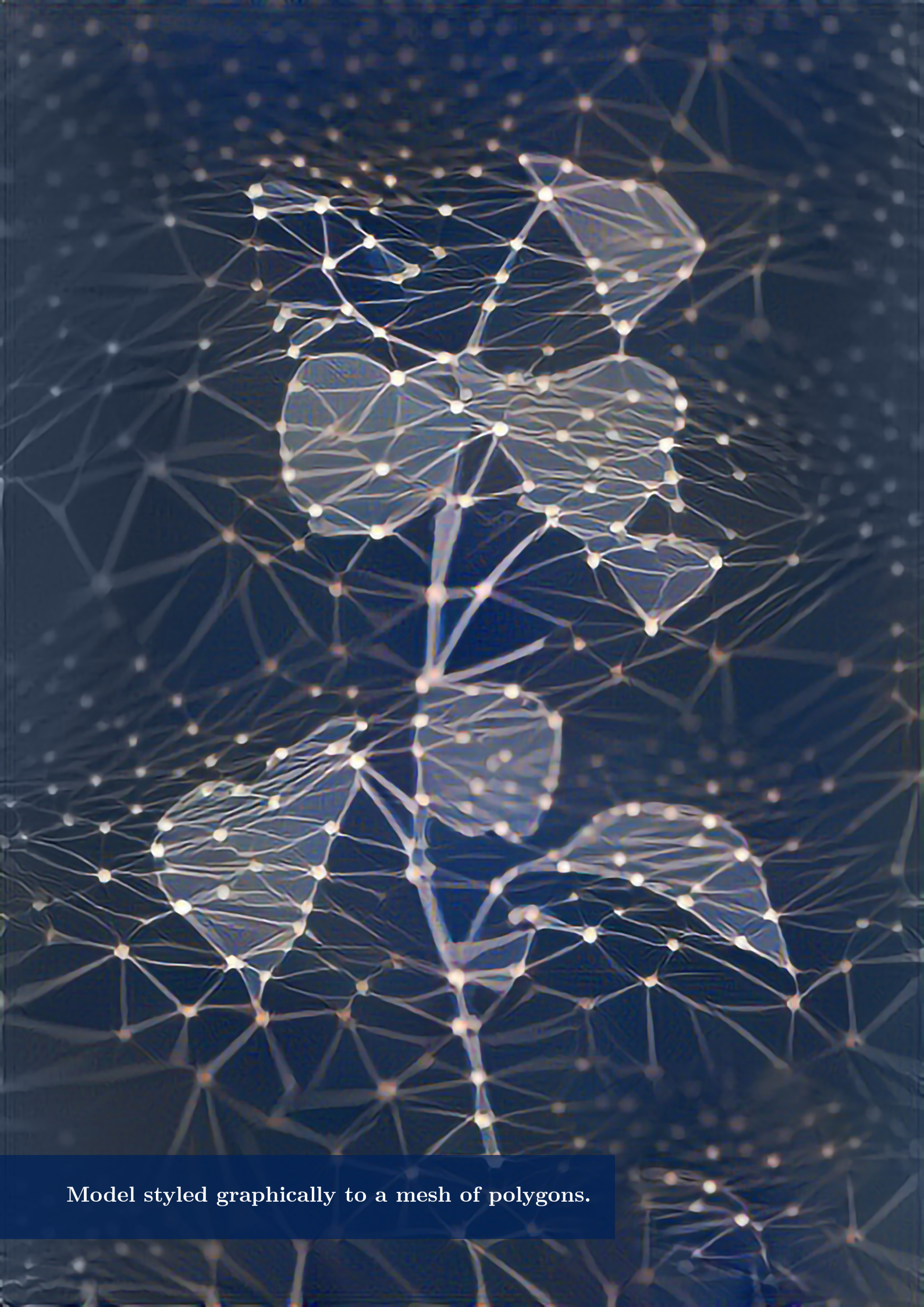
Mishkin, D., & Matas, J. (2015). All you need is a good init. *CoRR*, *abs/1511.06422*.

Papandreou, G., Chen, L.-C., Murphy, K., & Yuille, A. L. (2015). Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. (2016). The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, .



Model styled graphically to a mesh of polygons.

Chapter 6

Estimating Angles between Fruit and Stems to Support Grasping in a Sweet-Pepper Harvesting Robot

This chapter is based on:

Barth, R., Hemming, J., and van Henten, E.J. van (2017). Estimating Angles between Fruit and Stems to Support Grasping in a Sweet-Pepper Harvesting Robot. *Submitted to Computers and Electronics in Agriculture.*

Abstract

A method for estimating the angle between the plant stem and fruit is presented to support the grasp pose optimisation in a sweet-pepper harvesting robot. Our main hypothesis states that from color images, this angle in the horizontal plane can be accurately derived under unmodified greenhouse conditions. For this, we hypothesised that the location of a fruit and stem could be inferred in the image plane from sparse semantic segmentations. The scope of the chapter was focussed on 4 sub-tasks of the robot's harvest sequence. Each task was evaluated on 3 conditions: simplified laboratory, simplified greenhouse and unmodified greenhouse. The requirements for each task were propagated back from the end-effector design that required a 25° positioning accuracy. In Task I, color image segmentation for classes background, fruit and stem plus wire was performed, meeting the requirement of an intersection-over-union > 0.58. In Task II, the stem pose was estimated from the segmentations. In Task III, centers of the fruit and stem were estimated from the output of previous tasks. Both tasks met the requirement of 25 pixel accuracy on average. In Task IV, the centers were used to estimate the angle between the fruit and stem, meeting the accuracy requirement of 25° for 73% of the cases. The impact of the work lies in the support of successful grasping for robotic harvesting in the greenhouse.

6.1 Introduction

Successful harvest performance of agricultural robotics in dense crops relies on robust motion control and end-effector placement at the target fruit or vegetable (Bac et al., 2016, 2014b, 2017). The main objective presented in this chapter was to calculate the angle between the plant stem and the attached target fruit, with respect to the robot in the aisle. The contribution of the work lies in the potential to improve the grasp performance of a *Capsicum annuum* (sweet-pepper) robotic harvester, by optimising the visual servo control starting pose using this angle. This optimum was previously determined as the end-effector being in line with the fruit and stem (Bac et al., 2016). Our main hypothesis is that the angle between fruit and stem can be derived from color images under unmodified greenhouse conditions, whilst satisfying the accuracy requirement (25°) as imposed by the robot's end-effector. A key novelty of our method is the use of monocular 2D images to derive these angles, using a basic geometrical model. Hence no image depth information was required.

To distinguish between fruit, hard obstacles (i.e. wires, stems) and soft obstacles (i.e. leafs, leaf stems) (Bac et al., 2013b,a), detailed computer vision methods such as image segmentation on a plant-part level are needed. To realise state-of-the-art performance in plant part image segmentation, our previous research created synthesis methods for artificial images to overcome the machine learning requirement of large manually annotated data sets (Barth et al., 2018). A data set specifically for sweet-pepper was generated and was used to pre-train or bootstrap convolutional neural networks (CNN). When such synthetically bootstrapped networks were fine-tuned with a small manually annotated data set (30 empirical images taken in a greenhouse), a segmentation performance was achieved that was previously would allow for accurate fruit and stem localisation (Barth et al., 2017). In the research presented here, we pursued a similar hypothesis stating that from disconnected and sparse semantic segmentations, the location of a target fruit and corresponding stem can accurately be found in the image.

This research builds on other investigations that determined a stem-dependent grasp pose in sweet-pepper (Bac et al., 2017). Although their main hypothesis was that grasp success was improved by stem-dependent grasp optimisation, the research yielded statistically insignificant results. However, a positive effect was suggested under simplified and altered crop conditions (e.g. leaf removal). A key difference with the work presented here is that we did not evaluate overall end-effector grasp or cut success, but instead the end-effector positioning accuracy (Bac et al., 2016). This eliminated the end-effector's grasping and

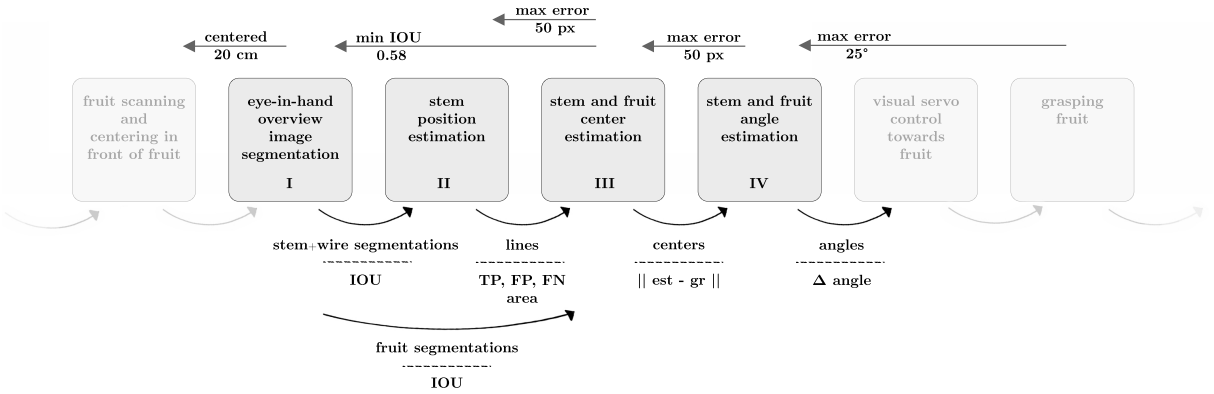


Figure 6.1: Part of the operation cycle of a sweet-pepper harvesting robot. Sub-tasks I through IV are presented this chapter. Tasks out of scope are shown in opaque for context. Underneath the tasks, the output is described (top) with their corresponding performance measures (bottom): intersection-over-union (IOU), true positives (TP), false positives (FP), false negatives (FN), Euclidean distance between estimated image pixel coordinates and ground truth ($||est - gr||$), and difference in angles with respect to the ground truth. Requirements are represented on top of the tasks, propagated back through the pipeline.

cutting performance as confounding factors in the evaluation. We thus focussed the scope of our research on determining the accuracy of the calculated angle and assumed consecutive steps performed according to their specifications. Another difference with the previous attempt is that our research estimates a continuous angle around the whole plant stem, as opposed to binned into 15 degree increments constrained within within 120° in front of the stem (Bac et al., 2017).

Recent related research on sweet-pepper harvesting (Lehnert et al., 2017) used a similar approach where the crop was first scanned to obtain a 3D scene whereafter an optimal grasp pose was determined for an open-loop motion control strategy towards the fruit. The end-effector placement and attachment combined had an overall success rate of 70%. One of their main conclusions was that improvements in the grasping success rate would result in greatly improved harvesting performance thereafter.

The work is part of an effort to integrate three previous research results into a closed loop, eye-in-hand robotic harvesting system (Barth et al., 2016, 2018, 2017). The scope of this chapter is depicted in Figure 6.1, decomposed in sub-tasks I through IV. We separated each step such that the performance could be evaluated and discussed intermediately, as opposed to evaluating only a final performance. This should give more insight in performance bottlenecks and their causes.

We started with the semantic segmentation of eye-in-hand overview images in Task I, which resulted in a per pixel segmentation for the classes fruit and stem. From these segmentations, stem poses in the image plane were estimated in Task II. Together with the fruit segmentation from Task I, the stem poses were used to calculate the stem and fruit centers in Task III. Using these centers and a model in Task IV, the angles between fruit and stem were estimated in Task IV.

Each task was performed under 3 conditions: i) simplified laboratory, ii) simplified greenhouse and iii) unmodified greenhouse in order to separately obtain the effect of natural variability of stems and fruit introduced from condition (i) to (ii) and the effect from adding leaf occlusions and neighbouring fruit introduced from condition (ii) to (iii).

To evaluate each task within the context of the final goal of positioning the end-effector accurately, we propagated the requirements of the next task back to the current task.

The chapter was structured to address the individual Tasks I-IV sequentially, each with their materials, methods, results, discussion and conclusion section. The chapter ends with an overall discussion and conclusion.

6.2 Task I: Semantic Image Segmentation

To ultimately determine the angle between stem and fruit in Task IV, a semantic segmentation of the classes *stem* (plus its support wire that supports the plant), *fruit* and *background* in images from the 3 conditions was first developed.

In the domain of computer vision, semantic segmentation divides images into non-overlapping class regions. Most methods consist of supervised learning, although weakly or non-supervised learning approaches have also been successful for certain problems (Wehrens, 2010; Zhu et al., 2016). Semantic segmentation localises classes in the images on a per-pixel level, as opposed to other computer vision methods that generate a high level label description of the image. High resolution classification is specifically useful for applications such as robotics where object manipulation and navigation is dependent on accurate positional information, e.g. autonomously driving cars (Shapiro, 2016; Badrinarayanan et al., 2017), warehouse order picking robots (Zeng et al., 2016) or in our case agricultural robotics (Bac et al., 2013b, 2016). In this section we build upon our previous segmentation research (Barth et al., 2018, 2017).

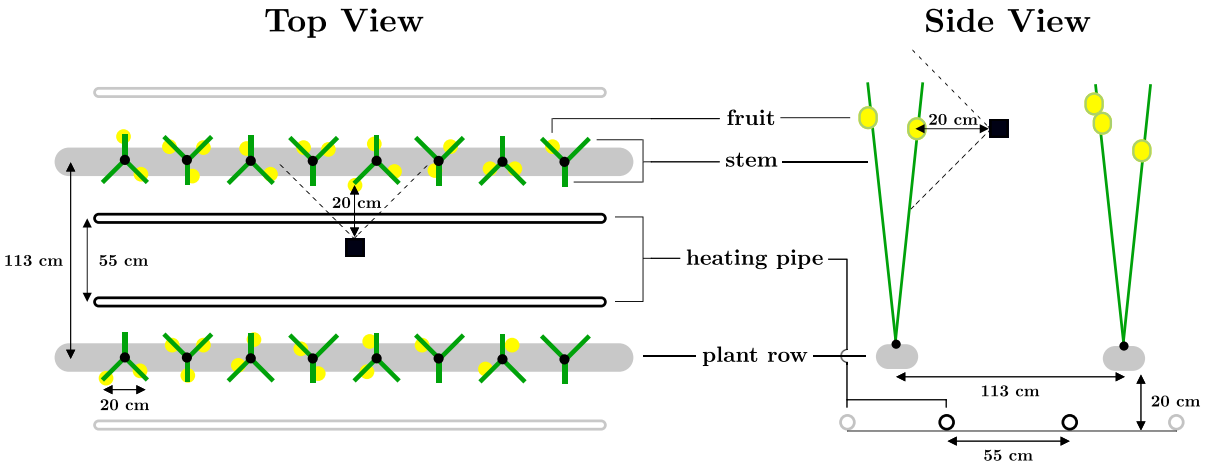


Figure 6.2: Schematic top and side view of one aisle in the greenhouse. The heating pipes formed the central infrastructure of the greenhouse on which harvest carts or image acquisition hardware could move between 2 rows of plants. Each plant consisted of 3 stems and the plants were placed to form double plant stem rows with 20 cm spacing between stems in the direction of the aisle. Camera pose indicated as a black square and dotted lines.



Figure 6.3: An artificial plastic leaf.

Figure 6.4: An imitation sweet-pepper.

Figure 6.5: An artificial plant stem section

6.2.1 Task I: Materials

Hardware

Eye-in-hand images were acquired using an F80 camera (Fotonic, Stockholm). The camera was mounted on a LR Mate 200iD robot arm (Fanuc, Japan). The scene was illuminated with a matrix of white LEDs, flashed for 50 μs, producing a light level of approximately 200.000 lx at a distance of 50 cm from the crop. The light distribution was highly centered with a sharp falloff at the edges of the camera's field of view.

Artificial Crop

An artificial sweet-pepper crop was created for the laboratory condition (see Figure 6.6) similar to the greenhouse situation (Figure 6.2). It consisted of imitation fruit and deformable stems and leafs. Plastic imitation leafs were shaped based on a real leaf outline (Figure 6.3). Leaf stems were deformable by their metal wire core. Commercial imitation peppers (Flora Bunda, China) consisted of a soft plastic outer shell and a foam core with a flexible rubber peduncle (Figure 6.4). For the plant stem, a malleable plastic hollow tube was fitted with metal wire to allow deformations that remained rigid (Figure 6.5). As plant nodes, metal shaft collars were placed along the stem, at 9 cm intervals (see (Barth et al., 2018)). The collars allowed magnetic attachment of the artificial leafs and fruit which were also fitted with magnets.



Figure 6.6: The artificial crop for laboratory condition tests. Background in image was removed for clarity.

Greenhouse Crop

For the greenhouse conditions, measurements were performed in a typical high-tech Dutch greenhouse of which a schematic in Figure 6.2 is shown. This situation differed from the growing system in previous research (Bac et al., 2017) where double plant rows were used with single stems as opposed to our situation of single plant rows with a double row of stems. However, both plant growing systems resulted in an average crop density of 7.2 stems / m^2. The image acquisition system was placed on the heating pipes between the plant rows, facing one plant row. The *Capsicuum Annuum* cultivar grown was Kaite (E20B.0073, Enza Zaden, the Netherlands).

Software

The publicly available semantic segmentation framework DeepLab V2 was used, which implements convolutional neural network (CNN) models (Papandreou et al., 2015; Chen

et al., 2015) on top of Caffe (Jia et al., 2014). Specifically, the VGG-16 network was used with a modification to include *à trous* spatial pyramid pooling for image context at multiple scales by convolutional feature layers with different fields-of-view (Chen et al., 2016; He et al., 2014).

6.2.2 Task I: Methods

For 3 conditions (simplified laboratory, simplified greenhouse and unmodified greenhouse), the following steps were taken to obtain semantic segmentation results.

Image Acquisition

For each category, 45 Images were taken binned to a resolution of 800x600 pixels at 20 cm in front of the center of the artificial and the greenhouse fruit. The pose of the camera was such that the optical axis was parallel to the ground plane.

Under the simplified laboratory condition, the fruit was placed in increments of 8 degrees around the stem for each consecutive image, hence covering 360 degrees around the stem for the whole image set.

For the unmodified greenhouse condition, the acquisition setup was placed at the next fruit available in the plant row. First, the unmodified condition images were taken. Next, the position of the acquisition setup remained equal and the leafs and neighbouring fruit were removed, whereafter the modified condition images were taken.

For Task IV the ground truth angle of the fruit and the stem was manually measured at this point in time using a protractor. For the exact method we refer to Section 6.5.1.

Ground Truth

Each image was pixel-wise manually annotated for the classes background, stem (plus wire) and fruit. All neighbouring ripe and unripe fruit and stems were included in the annotation. These annotations were used as training data for the CNN and to evaluate the performance of the segmentations afterwards.

CNN Training

A convolutional neural network was bootstrapped (Barth et al., 2017) with 8,750 synthetic bell pepper plant images (Barth et al., 2018). Because the limited data set size per

condition (45 images), we performed a two-fold cross-validation by first fine-tuning the CNN on one half of the data set and testing on the other and vice versa.

The hyperparameters of the network were manually optimised using the validation data set for combination of models and data set configurations as suggested by (Goodfellow et al., 2016; Bengio, 2012). This resulted in using Adaptive Moment Estimation (ADAM) (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ and a base learning rate of 0.001 for 30,000 iterations with a batch size of 4. These chosen hyper-parameters were previously found to be consistently optimal (Barth et al., 2017) and therefore we fixed them across conditions. An adjustment was made in the layer weight initialisation procedure, by updating the model to using MSRA weight fillers (He et al., 2015; Mishkin & Matas, 2015). Furthermore, the dropout rate (Srivastava et al., 2014) was adjusted to 0.50 to circumvent early overfitting and facilitate generalisation.

Performance Evaluation

To calculate the performance of the segmentation, we used the Jaccard Index similarity coefficient as an evaluation procedure, also known as the intersection-over-union (IOU) (He & Garcia, 2009) which is widely used for semantic segmentation evaluation (Gabriela Csurka , Xerox Research Centre Europe ; Everingham et al., 2010). The measure is defined in Equation 6.2, where the mean IOU per class equals the intersection of the segmentation and the ground truth divided by their union. A higher IOU implies more overlap, hence better performance. To derive the measure, a pixel-level confusion matrix C is calculated first for each image I in data set D :

$$C_{ij} = \sum_{I \in D} \left| \left\{ p \in I \mid S^I_{gt}(p) = i \wedge S^I_{ps}(p) = j \right\} \right|, \tag{6.1}$$

where $S^I_{gt}(p)$ is the ground truth label of pixel p in image I and $S^I_{ps}(p)$ is the predicted segmentation label. This implies that C_{ij} equals the number of predicted pixels i with label j. The average IOU per class L was calculated as:

$$IOU \ = \frac{1}{L} \sum_{i=1}^{L} \frac{C_{ii}}{G_i + P_i - C_{ii}} \ , \text{ where} \tag{6.2}$$

$$G_i = \sum_{j=1}^{L} C_{ij} \ \text{ and } \ P_j = \sum_{i} C_{ij} \tag{6.3}$$

Hence G_i denotes the total number of pixels labeled with class i in the ground truth and P_j the total number of pixels with prediction j in the image.

Requirements

The requirements for the semantic segmentation in Task I propagated back from Task IV (maximum 25 degree error, see Section 6.5.1), Task III (maximum 50 pixel center distance, see Section 6.4.1) and Task II (maximum 50 pixel center distance, see Section 6.3.2).

As the performance in this task was measured by the overlap between segmented pixel regions and the ground truth (Section 6.2.2), we could approximate how much IOU per class was minimally required in order to not exceed the 50 pixel center distance error limit in the next tasks.

For the fruit segmentation, we assumed a fruit could be described by a circular model. In the images, on average a fruit at 20 cm had a radius of 125 pixels in the image. We calculated the IOU of two of such overlapping circles, shifted by 50 pixels (see Figure 6.7).

First the intersecting area $area_{int}$ in pixels of the two circles was obtained, as defined in Equations 6.4 and 6.5 using distance $d = 50$, radii r_1=125 and r_2=125. Note that we used the four-quadrant inverse tangent $atan_2$ to allow calculating the arctangent of all quadrants in one function. Second, the union area was obtained by adding to the intersection, twice the subtraction of the intersection from a full circle. By dividing by the intersection over the union area of the two circles, the minimally required IOU for fruit of 0.58 was obtained.

$$area_{int} = {r_1}^{2*atan_2(t,d^2+{r_1}^2-{r_2}^2)} + {r_2}^{2*atan_2(t,d^2-{r_1}^2+{r_2}^2)} - \frac{t}{2}, \quad where \tag{6.4}$$

$$t = \sqrt{(d+r_1+r_2)*(d+r_1-r_2)*(d-r_1+r_2)*(-d+r_1+r_2)} \tag{6.5}$$

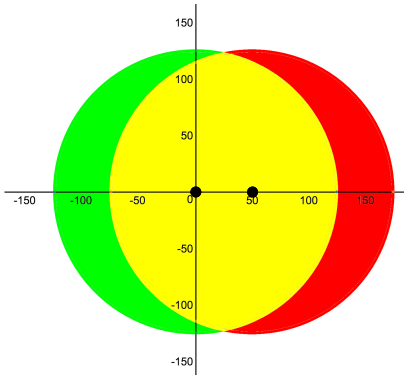


Figure 6.7: Maximum fruit center distance of 50 pixels as a requirement for image segmentation. Fruit modelled as circles (green, red) with a radius of 125 pixels. Yellow area indicates intersection area. Yellow segments equal the union area. Circles' centers are represented as black smaller circles.

To approximate the minimum required IOU for the stem segmentation, we assumed the stem could be described in the image as a rectangle of height of 600 pixels and 50 pixel width. Given this thin stem geometry, a translation of 50 pixels between classified pixels and ground truth would mean no overlap would occur. Hence, any overlap of the stem segmentation and the ground truth should result in a stem center estimation within 50 pixels. However this only holds for vertical stem orientations, which we arbitrarily defined as stems with a slope smaller than $\pm$ 0.15 in the pixel plane. For slanted stems, additionally it would be required to have overlap of top and bottom regions to allow a correct estimation of the pose of the stem in Task II. We evaluated this requirement qualitatively by observing the segmentation outputs of the slanted stems.

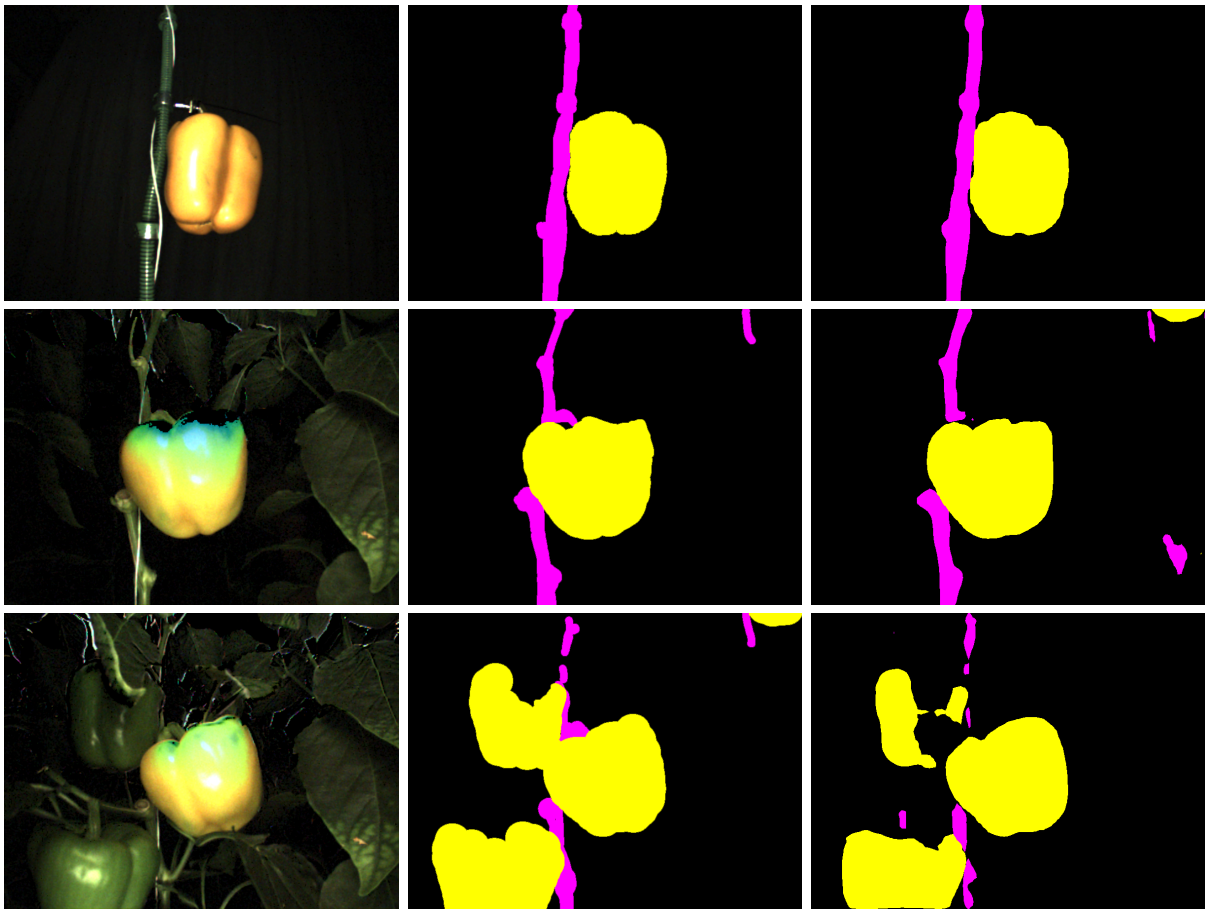


Figure 6.8: Segmentation examples for each condition of simplified laboratory (96°, top), simplified (#28, middle) greenhouse and unmodified greenhouse (#28, bottom). Color image (left) with its ground truth (middle) and classification (right). Class labels: ● background, ● stems (including wires) and ● fruit.

6.2.3 Task I: Results

In Figure 6.9 the mean IOU per class over the test sets for each condition is shown. For the simplified laboratory condition, the average IOU over all classes was 0.93. Regarding the simplified greenhouse conditions, this was 0.84 and for the unmodified condition 0.77. An example of a per-pixel classification for each condition, including source image and ground truth can be seen in Figure 6.8.

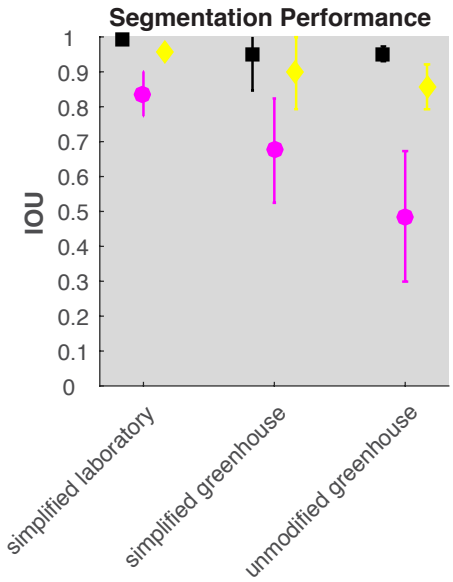


Figure 6.9: Mean IOU results per class per condition, including 1 standard deviation error bars. Class labels: ■ background, ● stems (plus wires) and ◆ fruit.

6.2.4 Task I: Discussion & Conclusion

Previous work on plant part segmentation in sweet-pepper applied classification and regression trees (CART) on hyper-spectral images (Bac et al., 2013a). Comparison with our method was not quantitatively performed, as a different performance measure was used. However, the reported qualitative results and true-positive detection rates of 40.0% and 54.5% for stem and fruit respectively and a scaled false-positive detection rate ($\frac{100 \cdot FP}{TP+FN}$) of 179.0% for stems and 17.2% for fruit, suggested our segmentation method outperformed the previous method significantly.

Under optimised laboratory conditions, the CNN's semantic segmentation performance (0.93) can be considered very high. After evaluating the raw CNN output, the remaining error was considered likely due to the CNN being cautious of assigning the stem class, favouring often the background label instead.

In the simplified greenhouse condition the performance was lower than in the laboratory (0.84), which could be expected when introducing natural plant diversity. Given the low data set fine-tuning size, not all variance could likely be captured. The CNN's generalisation was often correct, but not flawless.

For the unmodified greenhouse images, the average IOU was lowest of all conditions (0.77) and included the effect of introducing natural occlusions and neighbouring fruit in the scene.

First we note that for each class, the performance decreased about the same relative amount per class as the transition from the simplified laboratory to the simplified greenhouse condition. This is a remarkable pattern, though since the condition change was categorial and not parametric, the similar performance decrease between conditions was likely to be coincidental.

Secondly, the effect of occlusions and neighbouring fruit had a negative impact on performance. This might be explained by the convolutional nature of the model. Feature kernels can be learned more distinctly for each class when the previous feature map exists out of a uniform and consistent feature distribution, for example when only one class is present in the current perceptive field. When occlusions are introduced, a kernel has to be learned with additional and more overlapping class feature distributions as opposed to when non-occluded images are used. In turn this introduces uncertainty at a more early stage of the classification, likely resulting in larger guesses by the network and lower overall performance.

Given this explanation, one would expect that this effect should be larger for classes with a small area (e.g. stem class) and that were potentially more prone to occlusions than other parts. Vice versa, the effect should be smaller for classes with a large area (e.g. fruit), as those should be relatively less affected by occlusions. Hence, the relative loss of local information due to occlusion is higher for smaller parts than for large parts, which should result in lower performance. Indeed this effect can also be observed in Figure 6.8.

Some error can be explained by manual annotation faults. For example, for the simplified greenhouse condition in Figure 6.8 we note that the ground truth was incorrectly labeled at 2 locations. First, the peduncle of the simplified greenhouse image was labeled as stem, which should have been labeled as background. Second, in the top right of the image a bottom part of the pepper was left out during annotation, though the classifier correctly identified this as part of the fruit.

The requirements for this task were based on an assumption of geometric similarity of the segmented shape and an ideal shape of the fruit and stem. However, occlusions and natural variability could have made these shapes deviate from the assumed model.

Therefore the set requirements for this task should be considered an approximation and might be optimistic.

For each condition, regarding the *fruit*, the minimal IOU requirement of 0.58 was met for each image by a large margin. For the *stem* class any overlap would suffice for vertical stems, which was met by an IOU of on average 0.51. For the slanted stems (simplified: image #30, unmodified: #4,#17) the top and bottom regions should be included in the segmentation. We evaluated the latter qualitatively and the requirements was met for all samples except #17 in the unmodified condition. Closer evaluation of this sample showed the stem of interest was fully occluded and therefore could not meet any requirement.

6.3 Task II: Stem Pose Estimation

The resulting image segmentations from Task I were used to estimate one stem of interest per image and its pose in the image plane. In the unmodified greenhouse condition, this task especially was considered a challenge due to the sparse segmentations that arise due to leaf occlusions. Stem localisation is an important task for a range of applications from phenotyping (Sodhi et al., 2017; Pound et al., 2017) to robotic harvesting (Bac et al., 2017) as it often provides the central starting point for further sensing and manipulation tasks.

6.3.1 Task II: Materials

For each condition, the 45 image segmentations of the stem class from Task I were used as input for this task. To fine-tune the image processing hyper-parameters, the output from the previous task on novel and similar validation images for each condition were used.

To qualitatively investigate the robustness of the stem pose estimation algorithm for other distances and illumination conditions, an additional previously obtained data set was segmented by Task I and used in this task. This data set was imaged 50 cm in front of the crop and under an angle of 20 degrees looking upwards using an UI-5250RE-C-HQ PoE Rev.2 camera (uEye, Germany) with resolution of 800x600 pixels and a CMFA0420ND lens (Lensagon, Germany) with focal length of 4.16 mm.

As image processing software, the commercial software package Halcon 13 (MVTech, 2016) was used.

6.3.2 Task II: Methods

Stem Pose Estimation

For each stem (plus wire) segmentation image, first a region of interest was set by removing all segmentations at the outer left and right 20% of the image. We assumed that if the image was taken with a fruit in the center at 20 cm, only neighbouring or background stem parts could be situated in the outer areas. Connected regions of pixels were then identified as individual stem parts. Regions with an area below 300 pixels were discarded to remove noise from false positive segmentations.

On the remaining stem regions, a Canny edge detector (Canny, 1986) was applied. The lower threshold for the hysteresis threshold operation was set to 20 and the upper threshold to 40. The resulting edge direction image was used to detect straight lines by a Hough transform (Illingworth & Kittler, 1988) that was extended to use the local gradient (Petkovic & Loncaric, 2015).

Lines with a minimum distance of 10 between two maxima in the Hough image, in both the angle and image distance dimensions, were considered candidates for the stem estimation. By using the domain knowledge that plants grow mostly upward, candidate lines with a slope exceeding $\pm$ 0.15 in the pixel plane were filtered out. The remaining lines were then considered to represent a true path of a stem in the image plane.

The candidate line broadened to a width of 50 pixels (about one stem width in the image) that had the maximum overlap with the stem segmentation image, and with at least 20 pixels, was selected as the final estimated stem line.

For the additional data set, the same algorithm was used with slight adjustments for the parameters above to account for the difference in imaging distance. These parameters were obtained on a separate validation set.

Ground Truth

To quantitatively evaluate the performance of this task, a ground truth of the stem pose in the images was manually obtained by drawing straight lines in the image plane over the majority of the stem parts or subjective estimates thereof in case of occlusions. The pose consisted of an orientation that was saved as a line slope (S^m) and line intercept by the column pixel coordinate in the top of the image (S^c).

Performance Evaluation

The performance of this task for each image was measured by i) obtaining the stem detection true and false positives and negatives and by ii) an error based on the average pixel area between the estimated stem line and the ground truth line, within the dimensions of the image plane having a height $I_h = 600$ pixels and width $I_w = 800$ pixels.

The rationale behind the second performance measure was that the area between two lines summarises both angle and intercept errors into a single value that captures the intention of comparing overlap between lines. Separate evaluations of the angle and intercept parameters would be hard to interpret for line similarity. For example, two lines might be highly similar in intercept but highly dissimilar in slope, hence the interaction of the two parameters should be considered instead.

The coordinate system was defined as {0,0} in the top left corner of the image, with x increasing with decreasing image height and y increasing with image width.

In order to calculate the area between two lines, the integrals below both lines was subtracted. Therefore it was first required to obtain which line was the upper line at x=0 in our coordinate system (i.e. the stem positioned most to the right at the top in the image plane). For this Equation 6.6 defined the stem list index i as the line with the largest intercept and j as the line with the smallest intercept. In Equation 6.7 each stem was then defined as a line.

To calculate the area as defined in 6.8, the integral of the two lines was subtracted when they did not cross within the image plane. When the lines intersected, the combined integral of the subtraction left and right of the intersection of the lines was obtained.

The final performance measure is stated in Equation 6.9 as the area between the lines divided by the image height. Hence, the measure captures the expected average absolute horizontal pixel distance error between the calculated stem line and ground truth stem.

$$\text{arg find}(i,j) = \max(i \in 1,2 \mid S_i^c) \ \wedge \ \min(j \in 1,2 \mid S_j^c) \tag{6.6}$$

$$\begin{aligned} f(x) &= S_i^m x + S_i^c \\ g(x) &= S_j^m x + S_j^c \end{aligned} \tag{6.7}$$

$$area = \begin{cases} \int_0^{f(x)=g(x)} f(x) - g(x) dx + & \\ \int_{f(x)=g(x)}^{I_h} g(x) - f(x) dx & \text{if } \exists(x) : f(x) = g(x) \\ \int_0^{I_h} f(x) - g(x) \ dx & \text{if } S_i^c > S_j^{cs} \\ 0 & otherwise \end{cases} \tag{6.8}$$

$$\text{mean pixel distance error} = \frac{area}{I_h} \tag{6.9}$$

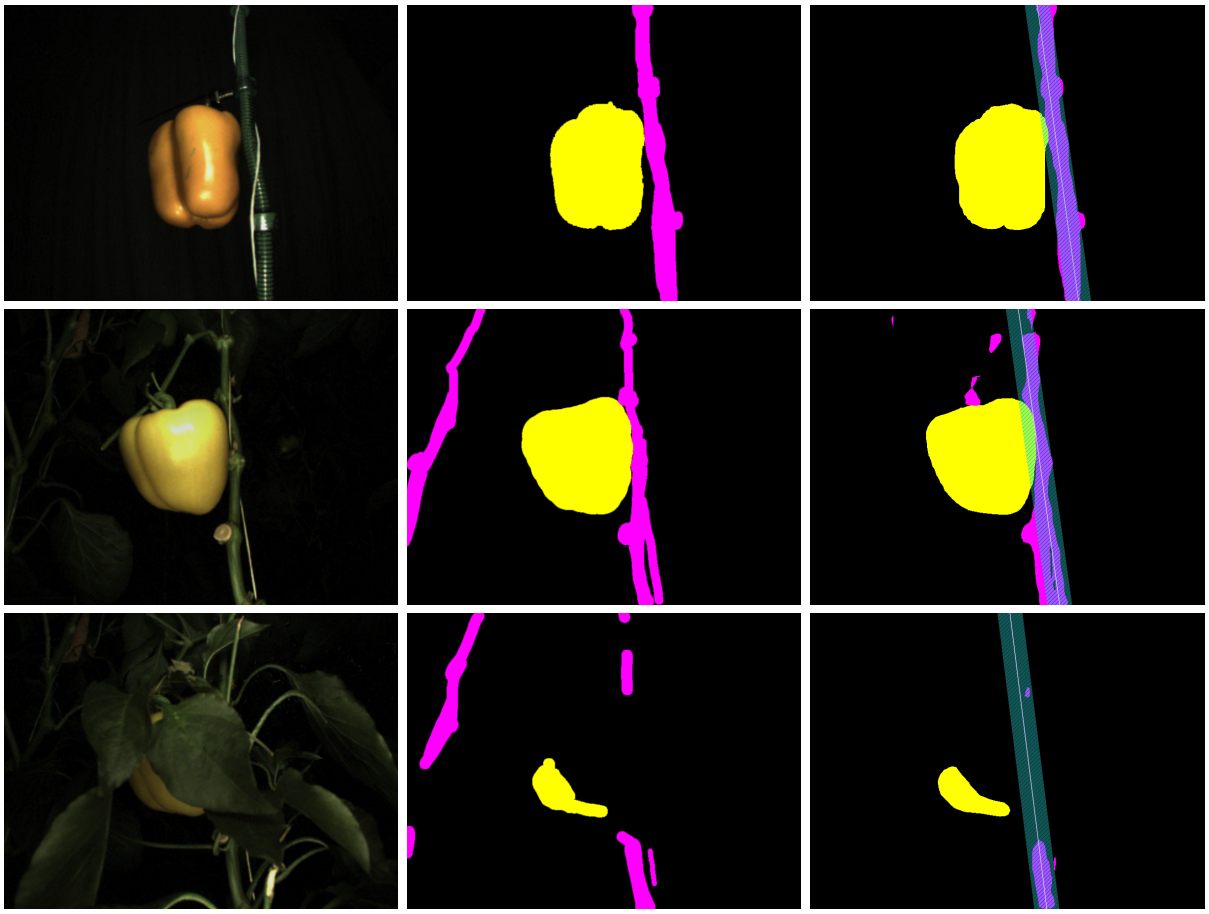


Figure 6.10: Examples of Task II results for each condition of simplified laboratory (112°, top), simplified greenhouse (#39, middle) and unmodified greenhouse (#39, bottom). Color image (left) with its ground truth class annotation (middle) and semantic segmentation (right) with estimated stem as a white line and striped overlay of an average stem width of 50 pixels. Class labels: ● background, ● stems (including wires) and ● fruit.

Requirements

The requirements for this task propagated back from Task IV (maximum 25 degree error, Section 6.5.1) and Task III (maximum 50 pixel error, Section 6.4.1). This implied that the mean pixel distance error of the stem estimation should also not exceed 50 pixels.

6.3.3 Task II: Results

Quantitative performance of the stem pose estimation on each image for each condition is shown in Figure 6.11. For the simplified laboratory condition, the average error over the image set was 6.6 (SD: 2.6) pixels. Regarding the simplified greenhouse conditions, the average error was 6.8 (SD: 6.1) pixels, whereas for the unmodified greenhouse condition the average error amounted to 12.1 (SD: 12.0) pixels. One false positive (FP) was detected

under unmodified greenhouse conditions (#10), hence for that condition a stem detection rate of 98% was achieved. For qualitative evaluation for each condition, an example stem estimation (including segmentation from Task I) is provided in Figure 6.10. The results on the stem estimation algorithm applied to the segmentations of the additional data set at 50 cm under illumination conditions are displayed in Figure 6.12.

6.3.4 Task II: Discussion & Conclusion

This task aimed to find and evaluate a stem pose estimation method from sparse segmentations, such as provided by Task I.

Previous work on stem pose estimation in sweet-pepper used the support wire as a visual cue in stereo images at a distance close to our additional 50 cm distance data set (Bac et al., 2014a). Our method on the 3 conditions differed i) in imaging distance, ii) by including stem segmentations and iii) the absence of depth information.

The performance measure of the previous work was similar to ours; the Euclidean distance from the ground truth stem and the interpolated support wire was averaged. Their method showed an average accuracy of 0.8 cm under laboratory conditions. Our method improves on those results as the 6.6 pixels amount to an error of 0.26 cm in real-world coordinates (see Section 6.5.1). Similarly, under unmodified greenhouse conditions previously an accuracy of 4.5 cm on average was obtained, compared to our average error of 0.48 cm in real-world coordinates (12.1 pixels). Regarding detection rates, our stem detection rate of 98% was slightly higher than the previously reported 94%, barring statistical proof.

For this task, we assumed that the best geometry to describe a stem was a straight line. Although stem parts themselves were straight, two adjacent stem parts could grow under an angle. Although the top and bottom stem parts could be connected with a straight line, intermediate stem segments could differ naturally from this line. At a distance of 20 cm, about 3-4 stem parts were generally in view. Nonetheless, the plants did grow straight up on average. Therefore to interpolate a stem from sparse parts, the straight line was considered the most likely overlapping geometry. Qualitative evaluation of the images confirmed this was the case.

The simplified laboratory condition could be considered to provide a baseline performance under near optimal conditions. Therefore in this condition an error close to zero with no false positives or false negatives was aimed for. However results showed an average error of 6.8 pixels. When observing the qualitative results in Figure 6.10, it can be seen that

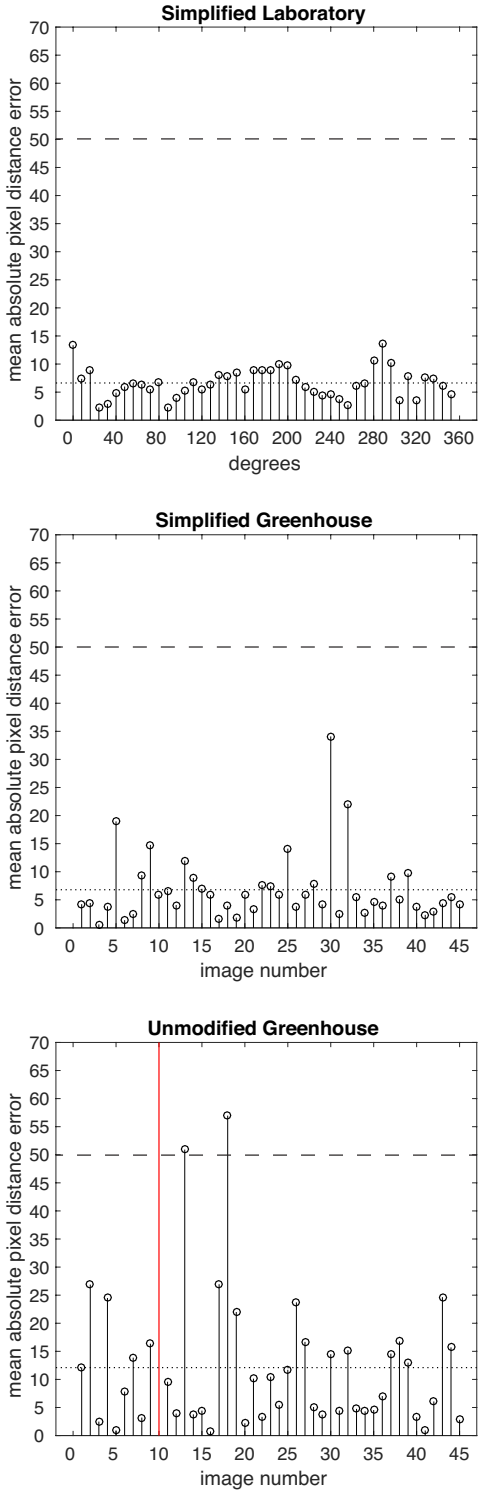


Figure 6.11: Mean pixel distance error between the calculated and ground truth stem for each image in each condition of simplified laboratory, simplified greenhouse and unmodified greenhouse. Horizontal dotted line indicates condition average. Horizontal dashed line indicates requirement level. Vertical red line indicated false positive. Note for the laboratory condition, images were sorted by their increments of 8 degrees (see Section 6.2.2).

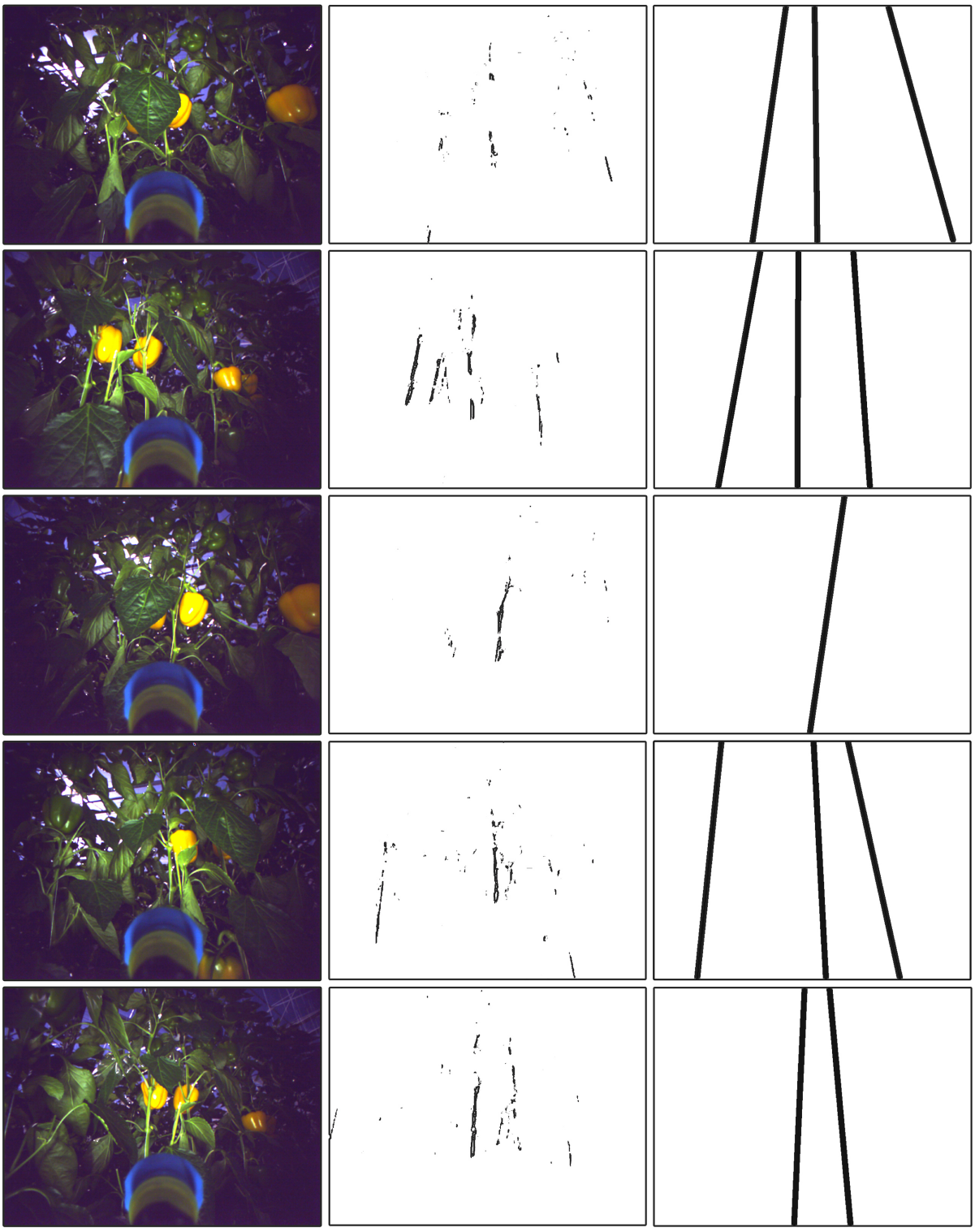


Figure 6.12: First 5 image examples (left) of the additional data set and their stem (plus wire) segmentations from Task I (middle) and the corresponding stem detection from Task II (right). Images were brightened for reader clarity.

the estimated stem line often runs diagonally through the stem segmentation. This can be explained by our method's definition of approximation a stem by a line, which maximised segmentation overlap whilst being 25 pixels wide. This differs from the ground truth lines, which were annotated through the core of each stem. Hence this difference caused part of the error.

In general, the qualitative performance for each condition was satisfactory as the estimated stem overlaps largely with the found segmentation (see Figure 6.10). Looking at the distribution of the error in Figure 6.11, we note for the worst cases (disregarding the FP) in the most difficult unmodified greenhouse condition, the estimated stem was about 1 stem width of 50 pixels away from the ground truth (image # 4, 13, 18, and 33).

The requirement of the stem estimation not to exceed 50 pixels on average was met for each condition, although 3 samples (including one FP) individually exceeded this limit. In the Figure 6.18 of the Appendix, these samples are displayed to allow for qualitatively evaluation of the worst case results. We observe that when occlusions were high, Task I was only able to segment a small stem area in the image. In return, this seemed not enough information to accurately estimate the slope of the stem as this required at least one large region or two smaller regions with some vertical distance between them.

Regarding the additional data set with images from 50 cm, we observe in Figure 6.12 that Task I qualitatively generates a sparse but matching stem segmentation from the color images. From this mask, our method was able to correctly find most stems in the image, with a few false negatives but no false positives. Note that images were taken under 20 degree angle pointing upwards, which was reflected in the convergent poses of the estimated stems.

6.4 Task III: Center Estimations

In this task the center of the target fruit was estimated in the image plane, along with the center of its corresponding stem. Although seemingly a trivial step, occlusions and neighbouring fruit required a fruit localisation based on a few assumptions.

6.4.1 Task III: Materials & Methods

The estimated stem poses from Task II were used, together with the fruit segmentations of Task I. To fine-tune the image processing hyper-parameters, the output from previous tasks on novel and similar validation images for each condition were used. In Figure 6.13 an overview of the steps in this task is displayed.

Based on the original image (Figure 6.13 A), first the vertical center of the estimated stem in the image segmentation (Figure 6.13 B) was calculated (Figure 6.13 C). Around this image coordinate, a rectangular search region of 200 pixels width and height was set with

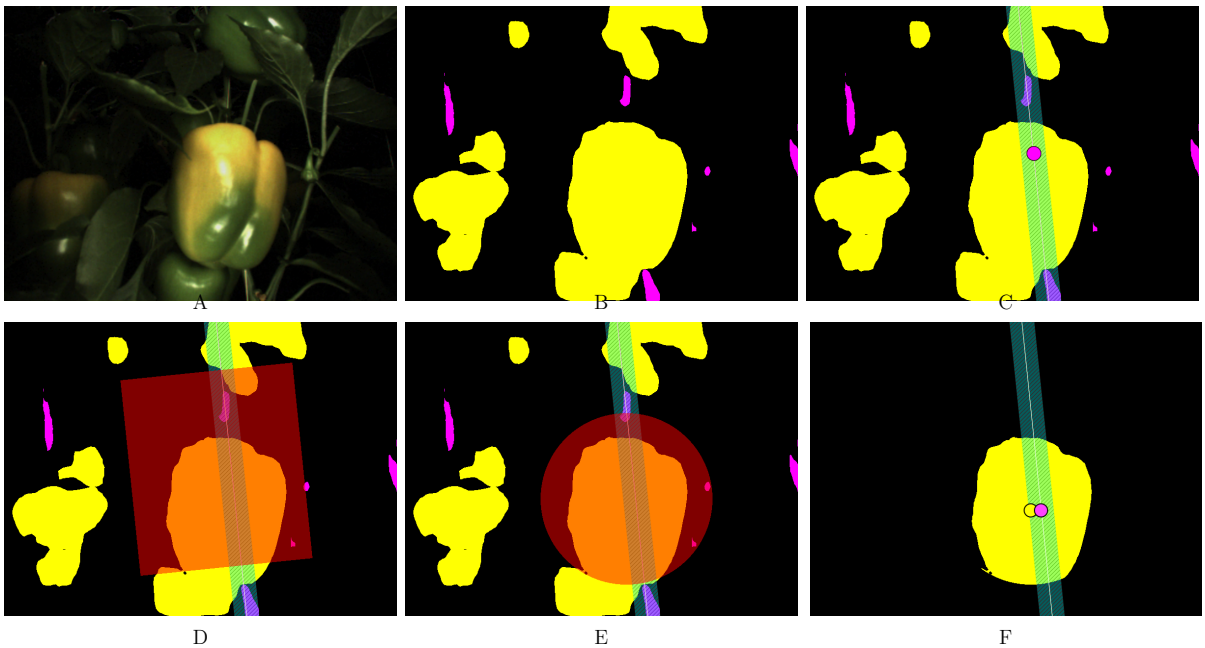


Figure 6.13: Method for estimating fruit and stem centers. (A) Original color image. (B) Semantic segmentation into classes background (black), stems (magenta) and fruit (yellow). (C) Stem pose estimation and found stem center in the image. (D) Search region for fruit within 200x200 pixel rectangle; the largest connected region was selected. (E) Around center of mass of selected region, the fruit estimation was refined with a 125 pixel circular mask. (F) Center of mass of refined fruit estimation was selected as final fruit center and corresponding stem center in the horizontal plane as final stem center.

the same orientation as the stem (Figure 6.13 D). We assumed that within this region our target fruit should be present, as images were taken with the fruit in the center of the image. The largest connected region in the original segmentation was then selected as a rough estimation of the target fruit mask. This mask excluded most fruit on adjacent stems, although fruit clusters on the same stem or fruit in the background still could be included in the remaining fruit segmentation.

To refine the remaining fruit segmentation and to suppress any background or clustered fruit, any pixels outside an average fruit radius in the image of 125 pixels (see Section 6.2.2) from the center of mass of the remaining fruit segmentation were discarded (Figure 6.13 E). From this refined region, again the center of mass was taken as final estimated center of the fruit.

Finally, the center of the stem was estimated by finding the point on the estimated stem line that was at the same vertical coordinate as the estimated fruit coordinate (Figure 6.13 F).

Performance Evaluation

To evaluate performance of this task, the ground truth of stem positions from Task II were used. Additionally, the ground truth of the fruit centers was obtained by manual annotation by drawing an enclosing circle around the fruit. The center of this circle and the closest point on the stem at the same vertical coordinate was used as ground truth.

We defined the performance measure for each image as the error described in Equation 6.10, where the Euclidean distance was calculated between the fruit and stem center estimations $est = (x_{est}, y_{est})$ and their corresponding ground truth centers $gr = (x_{gt}, y_{gt})$.

$$error = ||est - gr|| = \sqrt{(x_{gt} - x_{est})^2 + (y_{gt} - y_{est})^2} \tag{6.10}$$

Requirements

The requirements for this task were propagated back from Task IV (maximum error of 25°, see Section 6.5.1). Given the camera parameters as obtained in Section 6.5.1 and the model to obtain the angle between the fruit and the stem in Figure 6.16, a difference of 50 pixels at 20 cm distance would amount to a 25.4 degrees difference. Hence, the requirement for this task of an error less than 50 pixels per class was set.

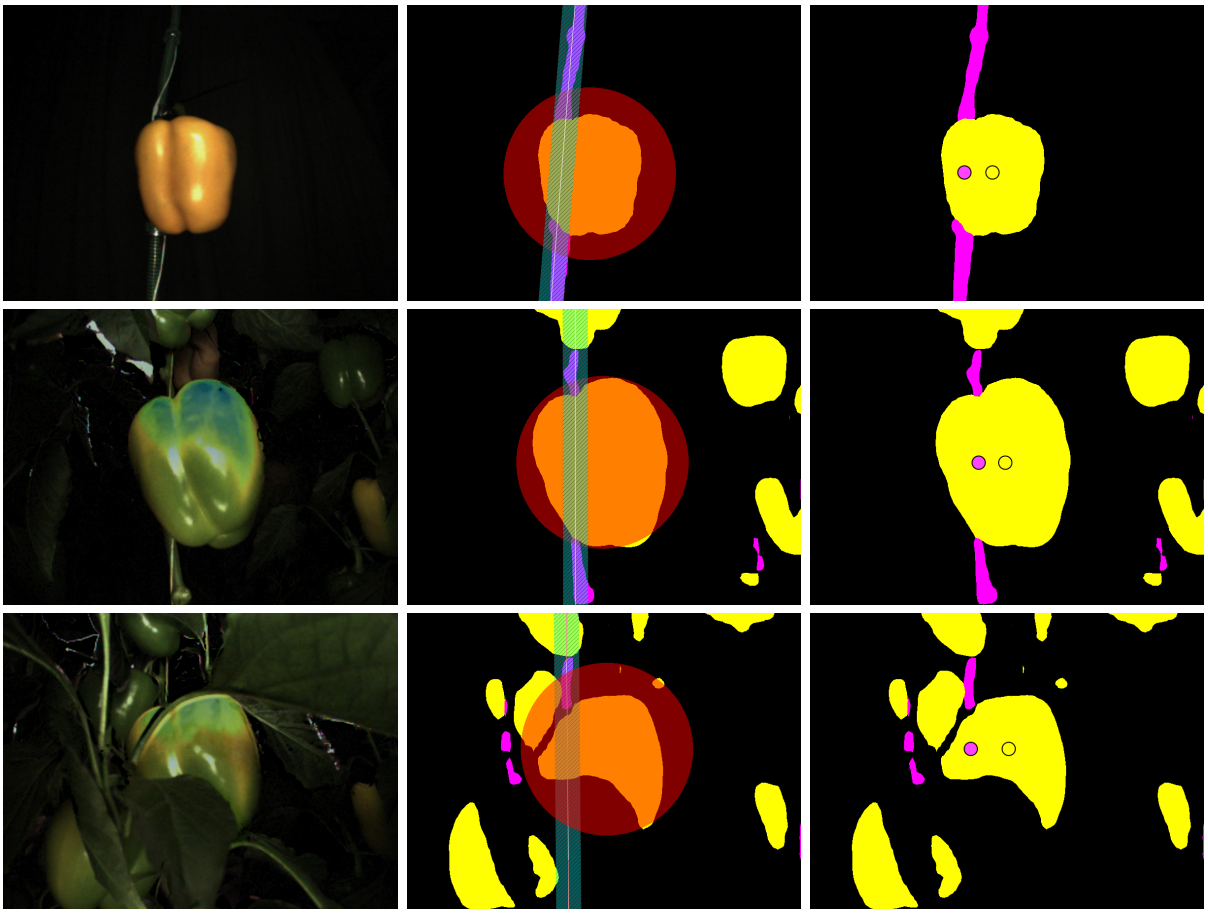


Figure 6.14: Examples of Task III results for each condition of simplified laboratory (320°, top), simplified greenhouse (#21, middle) and unmodified greenhouse (#21, bottom). Color image (left) with intermediate result (step E) with esitmated stem and refined fruit estimation area (middle) and final fruit and stem center estimates (right). Class labels: ● background, ● stems (including wires) and ● fruit.

6.4.2 Task III: Results

Qualitative examples are displayed in Figure 6.14 for each condition, including intermediate step E as demonstrated in Figure 6.13.

The performance of stem and fruit center estimation for each condition for all images is displayed in Figure 6.15. For the simplified laboratory condition, stem and fruit center errors were on average 5.2 (SD: 3.5) and 3.7 (SD: 2.1) pixels respectively. Regarding the simplified greenhouse condition, the errors were on average 5.0 (SD: 4.4) pixels for the stem centers and 7.1 (SD: 7.3) pixels for the fruit centers. Under unmodified greenhouse conditions, the stem center error averaged 15.0 (SD: 16.5) pixels (excluding outlier #10) and for the fruit centers 37.3 (SD: 30.3) pixels.

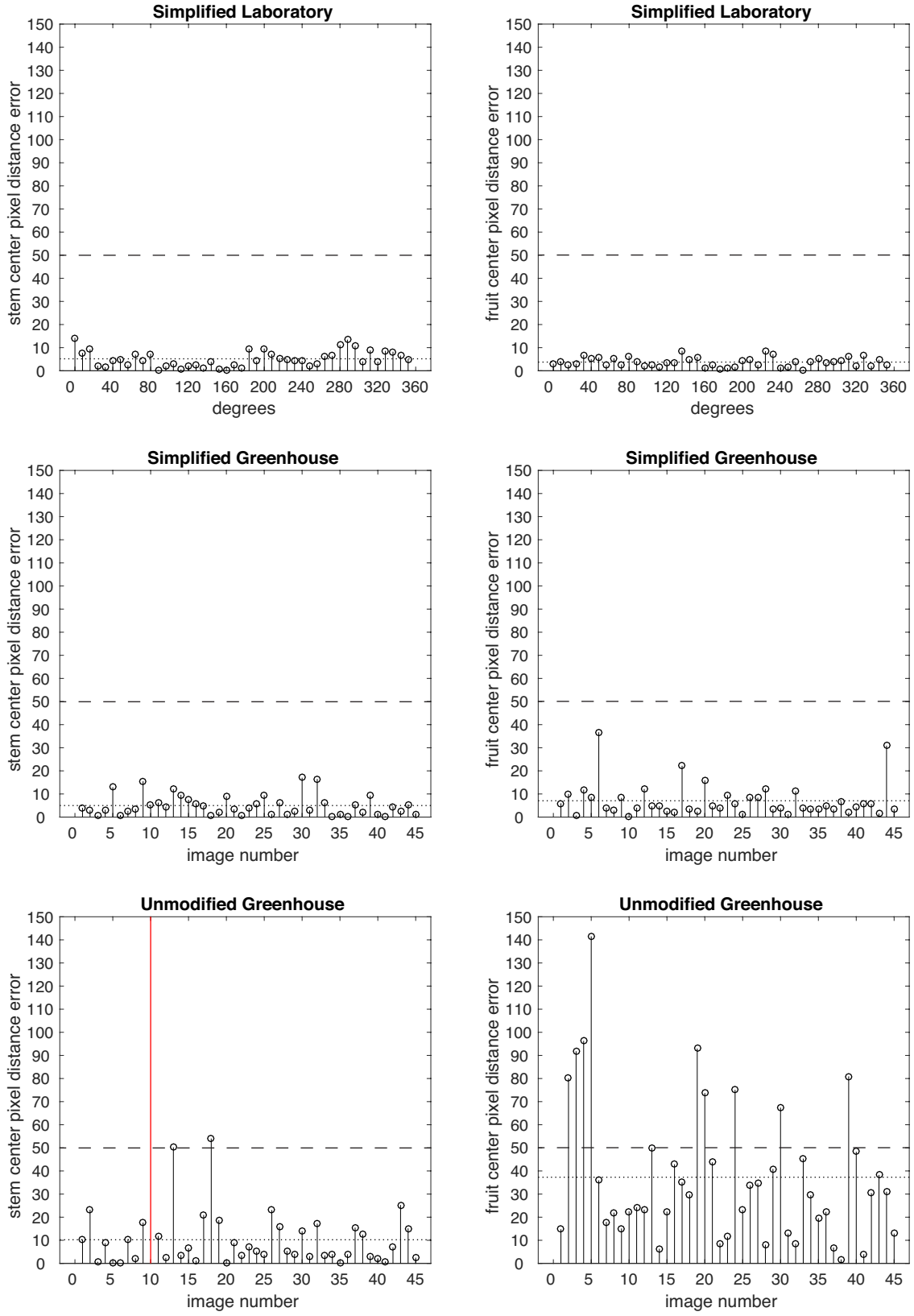


Figure 6.15: Stem (left column) and fruit (right column) center pixel distance error for each image in conditions of simplified laboratory (top), simplified greenhouse (middle) and unmodified greenhouse (bottom). Horizontal dotted line indicates condition average. Horizontal dashed line indicates requirement level. Red vertical line indicates false positive detection.

6.4.3 Task III: Discussion & Conclusion

It has been show that with our method, under simplified conditions highly accurate centers of stems and fruit can be found with an average of 5.1 (stem) and 5.4 (fruit) pixel distance between the calculation and the ground truth. When converted to world coordinates (see Section 6.5.1), pixels in the image plane at 20 cm were distanced 0.40 mm from each other. Hence, the error under simplified conditions (both laboratory and greenhouse) for each plant part was on average about 2 mm.

The real challenge of finding accurate centers of plant parts was introduced by taking neighbouring fruit and occlusions of other leafs into account. Under unmodified greenhouse conditions the error increased to an average of 15.0 pixels for stem centers, which equaled to a real-world error of 6 mm. For fruit centers this was 37.3 (SD: 30.3) pixels or 15.0 mm. Figure 6.15 informs us that there was a sub-class of 9 worst case scenario samples (#2,3,4,5,19,20,24,30 and 39) of around 80 pixel error, or 3 cm.

The error under modified greenhouse conditions for fruit centers was on average fourfold than for the stems. Although stem parts were more prone to occlusion (as discussed in Section 6.2.4), their relative better performance can be explained. Because the full stems had a vertical and narrow shape property, any found segmentation already constrains the error mostly to the horizontal component. For fruit this does not hold, as they have a circular shape property the error has components in both dimensions. Furthermore we must note that the vertical component of the stem center error was also constrained by the found center of the fruit, as described in Section 6.4.1. Hence, first the fruit center was found and that was used to estimate the stem center at that same height in the image. Therefore, the effect of the vertical component on the error was effectively reduced to zero.

In an attempt to constrain the error further, we have tried to use the fruit shape property by fitting inner and outer enclosing circles around the segmentation to make up for partial segmentation or occlusions. However, we found that neighbouring and background fruit in combination with a high variance of occlusions resulted in fruit shapes that could not be reasonably en- or inclosed to predict the original fruit shape.

In order to improve the fruit center estimation, background and neighbouring fruit could be independently segmented as separate instances, for example with a Mask-RCNN approach (He et al., 2017).

Regarding the chosen performance measure, we must note that because errors from both classes *stem* and *fruit* might add equally to the angle estimation error, their sum might need to be considered as the requirement instead. However, this sum was not chosen as the requirement because the errors of stem and fruit were assumed to correlate, given the localisation method described in Section 6.4.1. In other words, we assumed it to be unlikely that a stem was detected with an 50 pixel error with a fruit center detected with another 50 pixel error in the opposite direction. Nonetheless, the plant part errors were likely at least in part to accumulate and hence the evaluation of the requirements of this task was therefore an approximation.

When the results are compared to the desired requirements, they were met on average for each condition and for each plant part. However, for the stem and fruit centers under unmodified greenhouse conditions, respectively 2 (4%) and 9 (20%) of the samples did not meet the requirements individually. This might affect the angle estimation in the following Task IV.

6.5 Task IV: Angle Estimation

Based on the stem and fruit center estimates, in Task IV the angle between the fruit and the stem was derived, specifically the angle ϕ in the horizontal world plane, relative to the crop aisle from which the images were taken.

In order for this estimation to work by just using color images, a priori knowledge of the geometry of the plant was used, as well as certain assumptions regarding sizes and distances in combination with parameters of the camera. This was combined into a model as depicted in Figure 6.16.

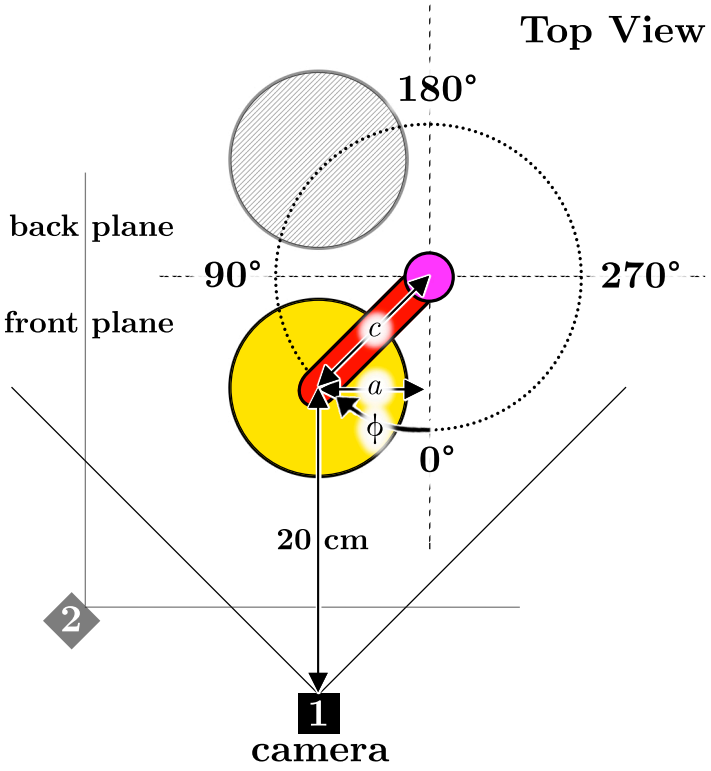


Figure 6.16: World top view schematic of the model used to estimate stem-fruit angle ϕ. Plant parts indicated by color: ● stem, ● fruit and ● peduncle. Note that the projection of the fruit on the camera image could be the result of the fruit being in the front or back plane of the stem. To remove this angle ambiguity, a second viewpoint (2) using ϕ was required.

6.5.1 Task IV: Materials & Methods

The estimated centers of fruit and stems from Task III were used as input for this task. In our model, angle $\phi = \arcsin(\frac{a}{c})$, where c was assumed to be a fixed value of half the average fruit diameter plus average stem diameter: $\frac{84+10}{2} = 47{,}5$ mm. These plant parameters were previously empirically obtained (Barth et al., 2018). To calculate a, the estimated stem and fruit centers from the previous tasks were mutually subtracted after their transformation from pixel to world coordinates in mm. We noted ϕ from the first viewpoint as $angle_{v1}$. The transformation required the intrinsic camera parameters, which were obtained through the calibration procedures in Halcon 13 (MVTech, 2016) for a distance of 20 cm. A translation of 1 pixel in the image amounted to 0.4 mm in the real world at a distance of 20 cm.

The ground truth was measured manually using a protractor enclosed around the stem and level with the ground plane, with 0° and 360° pointed towards the aisle. The angle of the center of the fruit with the protractor was noted as $angle_{gr}$.

Given that the projection of the fruit and stem on the camera image could be always the result of 2 angles (a fruit in the front plane or mirrored in the back plane of the stem, see Figure 6.16), our method initially obtained an ambiguous $angle_{v1}$. Therefore, to disambiguate $angle_{v1}$, an angle calculation of $angle_{v2}$ from a second viewpoint was required. If the second viewpoint was chosen to be $angle_{v1}$, then $angle_{v2}$ would be approximately 0° (plus some error) if $angle_{v1}$ was in the correct plane. If $angle_{v2}$ would be ±90° (plus some error), then $angle_{v1}$ was estimated in the incorrect plane. Because we did not foresee the requirement of a second viewpoint for this task, our data sets did not include such pairs of viewpoints and corresponding ground truths. Therefore we could not directly calculate the angle estimation performance after a single viewpoint for this task. However, we were able to estimate the performance under the assumption that the second viewpoint was taken and would have provided correct angle disambiguation as $angle_{dis}$.

To calculate the disambiguation, Algorithm 1 was used. Given $angle_{v1}$ from the first viewpoint and the ground truth $angle_{gr}$, first it was checked if the angle was ±90°, as for these angles there is no ambiguity. Otherwise, for fruit on the left or right side of the image, 3 errors were calculated: i) the $error_1$ of $angle_{v1}$ and the ground truth, ii) the $error_2$ of the mirrored $angle_{v1}$ with the ground truth and iii) $error_3$ to take into account a possible ground truth on the opposite left or right side of the stem, which due to discontinuity at 0° and 360° would otherwise introduce a wrong $error_1$ and $error_2$.

Using these errors disambiguation was performed as shown in the following algorithm.

For each image performance measure for this task is defined as an error in Equation 6.11.

$$Error = absolute(angle_{dis} - angle_{gr}) \tag{6.11}$$

Requirements

The requirement of the accuracy of the angle estimation, propagated back from the next tasks (out of scope of this chapter) of the end-effector placement at the target fruit. For sweet-pepper, ideally the exact angle should be known in order to position the end-effector in line with the stem and fruit (Bac et al., 2017). For the current end-effector, an error of up to 25° was expected to result in successful a fruit harvest by design.

6.5.2 Task IV: Results

Results are summarised in Figure 6.17. Under simplified laboratory conditions, an average error of 11.0 (SD: 6.5) degrees was found. Under the simplified greenhouse conditions, the error increased to an average of 18.2 (SD: 14.7) degrees. When occlusions and neighbouring fruit were present under unmodified greenhouse conditions, the average error was 24.5 (SD: 36.5) degrees.

6.5.3 Task IV: Discussion & Conclusion

Estimating an angle between two centers of plant parts from their estimated centers was performed in this task. Under laboratory conditions, all angle estimates were within the set requirement by the end-effector of 25°. Under simplified greenhouse conditions, on average the requirement was met, though 12 (27%) of the estimates (#2,3,8,16,20,22,32,35,39,42,43, and 45) did not satisfy the end-effector constraint. Although the errors under unmodified greenhouse conditions were higher, on average the requirement was met. However, also for this condition 12 estimates (#5,16,19,20, 22,23,33,34,39,40,42 and 43) exceeded the 25° requirement, hence 27% of the fruit would not have been harvested at the initial trial under normal greenhouse conditions.

input : $angle_{v1}$ with interval [-90,90];
$angle_{gr}$ with interval [0,360]
output: $angle_{dis}$ with interval [0,360];

if $angle_{v1} == 90$ **then**
  $angle_{dis} = 90$
else if $angle == -90$ **then**
  $angle_{dis} = -90$
else if $angle_{v1} >= 0$ **then**
  $error_1 =$ absolute($angle_{gr} - angle_{v1}$);
  $error_2 =$ absolute($angle_{gr} - (180 - angle_{v1})$;
  $error_3 =$ absolute($(360 - angle_{gr}) + angle_{v1}$);
  if $error_3 < error_1 \wedge error_3 < error_2$ **then**
    $angle_{dis} = angle_{v1}$;
  else if $error_2 > error_1$ **then**
    $angle_{dis} = angle_{v1}$;
  else
    $angle_{dis} = (180$ - $angle_{v1})$;
  end
else if $angle_{v1} < 0$ **then**
  $error_1 =$ absolute($angle_{gr} - (360 + angle_{v1})$);
  $error_2 =$ absolute($angle_{gr} - (180 - angle_{v1})$;
  $error_3 =$ absolute($angle_{gr} + (angle_{v1} * -1)$;
  if $error_3 < error_1 \wedge error_3 < error_2$ **then**
    $angle_{dis} = (360 + angle_{v1})$;
  else if $error_2 > error_1$ **then**
    $angle_{dis} = (360 + angle_{v1})$;
  else
    $angle_{dis} = (180 + (angle_{v1} * -1)$;
  end
end

Algorithm 1: Given an estimated $angle_{v1}$ from the frontal viewpoint (Figure 6.16) that might yield a fruit angle estimate in the wrong plane due to ambiguity from a single viewpoint and given the corresponding ground truth $angel_{gr}$, the fruit angle $angle_{dis}$ that corresponds to the correct plane was calculated, based on the assumption the second viewpoint would remove the ambiguity. Note that the intervals of the input and output differ.

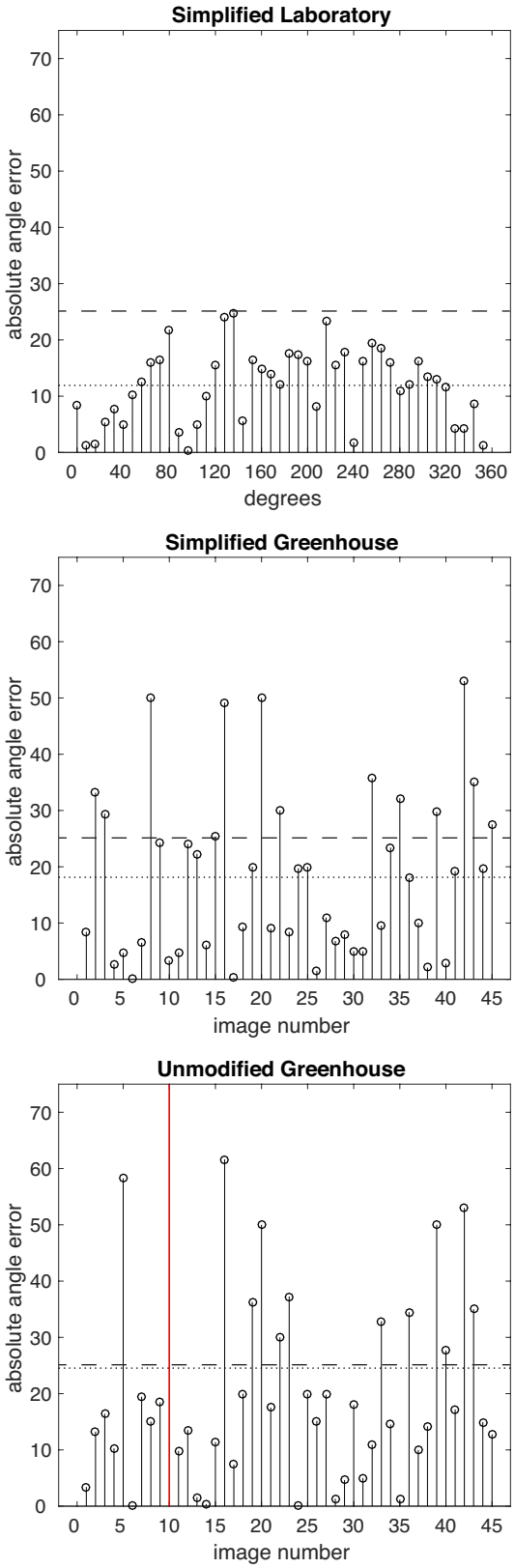


Figure 6.17: Error for each image for each condition between estimated disambiguated angle and ground truth. Horizontal dotted line indicates the condition average. Horizontal dashed line indicates requirement level. The red vertical line indicates false positive detection from a previous task.

For the angle estimation method, we assumed a fixed diameter of the fruit and stem. Although this holds under the simplified laboratory condition where we used plant parts with fixed dimensions, for the greenhouse conditions the fruit diameter had a standard variance of 11 mm. Therefore, this variance was an additional factor in the error of the estimated angle under greenhouse conditions. A fruit size estimation might decrease this effect, although such an estimation under unmodified conditions was considered hard due to occlusions. For the stem, the variance was measured close to zero.

Additionally, as opposed to the laboratory condition where the robot could be precisely positioned 20 cm in front of the fruit center, this was manually cumbersome to perform in the greenhouse. The limited working space influenced the ability to position the fruit in front of the camera accurately and consistently across simplified and unmodified greenhouse conditions. This added another factor to the error. Indeed when we evaluated the color images, we noted that many samples did not have an equal size fruit as compared to the laboratory setting. If the manipulator had a distance sensor near the camera, e.g. laser or time-of-flight based, this factor could have been suppressed.

Similarly, accuracy errors were likely introduced in the ground truth. Predominantly true for the greenhouse conditions, precisely positioning the protractor enclosed around the stem and level with the ground plane, with 0° towards the aisle, in combination with the limited workspace, was a challenge. These errors accumulated in the final performance error measurement of this task. To limit this factor it is recommended for future work to average multiple measurements by different experts.

For our performance evaluation it was assumed that the second viewpoint would have provided the correct front or back plane in which the fruit was situated. This assumption only holds when there would be no false positive detections, which under unmodified greenhouse conditions was not the case for 1 out of 45 samples.

Although this discussion shows options for future improvements, the overall angle estimation performance shows promising results from monocular color images.

6.6 General Analysis and Reflection

We have contributed to the task of grasp-pose estimation for greenhouse harvesting robotics by proposing a method for estimating relative plant part angles based on color images. Although this task was not new in this domain, our approach has shown improvement over previous results.

In previous work that determined a grasp pose angle for a sweet-pepper harvesting robot (Bac et al., 2017), the angles of approach were constrained in a range of 120° facing the aisle. Within this range, a fruit and stem were localised in 3D space (Bac et al., 2014a), from which their relative angle was binned to increments of 15°. Hence, this approach introduced an default average angle error of 7.5° as opposed to our method which estimates a continuous angle fully around the stem.

Unfortunately, the previous work did not report an angle estimation performance evaluation and therefore we cannot compare results quantitatively. Instead, the overall harvest performance was previously reported. This implies that the effect of the angle estimation could not be independently evaluated from the two end-effector compositions and their corresponding harvest performance. Our approach of splitting the angle estimation as a separate task allowed us to exclude the confounding end-effector performance factor.

We were able to compare performance on the sub-task of stem detection; under unmodified greenhouse conditions previously an accuracy of 4.5 cm on average was obtained, compared to our error of 0.48 cm in real-world coordinates. Our stem detection rate of 98% was higher than the previously reported 94%.

Other recent related research on sweet-pepper harvesting (Lehnert et al., 2017) had an comparable overall success rate of 70% under unmodified greenhouse conditions whereas ours potentially 73%. One of their main conclusions was that improvements in the grasping success rate would result in greatly improved harvesting performance thereafter. We must note that this was achieved in a different cultivation system (Australia) as our research (the Netherlands), that might set different end-effector and placement requirements and moreover makes success rates difficult to compare.

On average the angle error met the end-effector requirement of 25° accuracy, although 27% of the fruit would not be harvestable in the first try under normal greenhouse conditions. We did not investigate what the effect would be of multiple tries or merging information from multiple viewpoints. However, we hypothesise that this would improve harvesting performance significantly, at cost of an increased harvest time, as was also indicated previously for cucumber harvesting (Henten et al., 2003).

Although the division into tasks was arbitrary, they were each delimited to represent an independent module. Yet, functionality within each task was shaped by output from previous tasks and had to meet the requirements from future tasks. Nonetheless each module should be freely substitutable.

The modules' dependency on previous module output was reflected by additional experiments we ran. Actual data was replaced with ground truth data for Tasks II through IV and the performance difference was compared. Contradictory to our intuition, results showed Tasks II and III had worse performance with ground truth data than actual data. One explanation for this was that certain characteristics of the CNN output, that were the differences with ground truth, were used to shape the stem pose and fruit position detection algorithms.

For Task IV, using ground truth data, the average angle error for the simplified laboratory, simplified greenhouse and unmodified greenhouse were 12.7 (SD: 7.7), 16.8 (SD: 15.2) and 18.2 (SD: 15.0) pixels respectively. Hence the error was 15% higher (laboratory), 8% lower (unmodified greenhouse) and 26% lower when using ground truth data (see Section 6.5.2). This suggested that the error made solely in Task IV was more substantial than the error accumulated from previous tasks. This error created in Task IV likely consisted of at least 3 components; i) modelling and assumption deficiencies (Figure 6.16), ii) camera calibration errors (0.13 pixels) and iii) manual ground truth labeling errors. Hence, it is advisable for future performance enhancement that these components should be investigated first over improving the output from previous tasks.

To improve the angle estimation error, the model could be improved by plant measurements, like fruit size. Multiple viewpoints by fruit scanning and merging the information into a world model (as performed in (Lehnert et al., 2017; Henten et al., 2003)) might provide such information, though given the amount of occlusions, such parameters would be hard to accurately obtain.

For Task I we assumed the positioning at a distance of 20 cm in front of the center of the fruit was achieved automatically by a previous task, although this was not yet available to the robot at the time of our research. Therefore, the distance was manually fixated. Under laboratory conditions this could be accurately achieved as opposed to under greenhouse conditions. This might have introduced a large component of error as our pixel to world coordinate transformations were calibrated for a single camera distance of 20 cm. Furthermore, the distance had influence on the center position of the plant parts in the image. Depth information (e.g. laser distance reader or time-of-flight) to accurately position the camera in front of the fruit would likely reduce this error in the future. Depth information could also disambiguate if the fruit would be in the front or back plane, removing the need a second viewpoint.

Although the quantitative analyses provided an approximation of the performances in each task, it could not prove statistical significance given the low amount of samples in our conditions. Instead, results should be interpreted as an indication of the direction of the effects.

Also because the requirement from the end-effector was propagated back through the pipeline, requirements per task had to be interpreted from the perspective of the final grasping requirement, introducing some assumptions. Therefore the evaluation with regard to the requirements should also be seen as an approximation.

Overall the performance of the method would allow up to 73% of the fruit to be harvestable in the first try, depending on the performance of other tasks like grasping and cutting thereafter. Future research should verify this performance in the full pipeline of the robotic harvester. We have shown that although each task has a satisfactory performance, error does accumulate through all tasks. For a greenhouse robot, ideally the harvesting performance should be near 100%. However it might be economically viable to perform an initial harvest with robots and the remainder with human efforts, either through a remote interface or physically in the greenhouse.

6.7 Conclusion

A method for estimating the angle between the plant stem and fruit was presented to support the grasp pose optimisation in a sweet-pepper harvesting robot. Our main hypothesis is that from color images, this angle in the horizontal plane can be accurately derived under unmodified greenhouse conditions. For this, we additionally hypothesised that the location of a fruit and stem could be inferred in the image plane from sparse semantic segmentations.

The scope of the chapter was focussed on 4 sub-tasks of the robot's harvest sequence. The requirements for each task were propagated back from the end-effector design that required a 25° positioning accuracy.

Experiments were performed under simplified laboratory, simplified greenhouse and unmodified greenhouse conditions in order to compare the effect of the factors of natural variability, occlusions and neighbouring fruit on the performance of each task. Each factor was found to increase baseline error of the simplified laboratory condition.

In Task I, color image segmentation for classes background, fruit and stem plus wire was performed, meeting the requirement of an intersection-over-union greater than 0.58.

In Task II, the stem pose was estimated from the segmentations. In Task III, centers of the fruit and stem were estimated from the output of previous tasks. Both tasks met the requirement of 25 pixel accuracy on average, confirming our additional hypothesis.

In Task IV, the centers were used to estimate the angle between the fruit and stem, meeting the accuracy requirement of 25° for 73% of the cases and confirming our main hypothesis.

The impact of the work lies in the support of successful grasping for robotic harvesting in the greenhouse.

6.8 Acknowledgements

This research was partially funded by the European Commission in the Horizon2020 Programme (SWEEPER GA no. 644313). We would like to thank Joris IJselmuiden for his reflections on the experiments and Bart van Tuijl and Toon Tielen for helping with the data collection.

References

Bac, C., Hemming, J., & van Henten, E. (2013a). Pixel classification and post-processing of plant parts using multi-spectral images of sweet-pepper. *IFAC Proceedings Volumes*, *46*, 150 – 155. doi: http://dx.doi.org/10.3182/20130327-3-JP-3017.00035. 5th IFAC Conference on Bio-Robotics.

Bac, C., Hemming, J., & van Henten, E. (2013b). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture*, *96*, 148 – 162. doi: https://doi.org/10.1016/j.compag.2013.05.004.

Bac, C., Hemming, J., & van Henten, E. (2014a). Stem localization of sweet-pepper plants using the support wire as a visual cue. *Computers and Electronics in Agriculture*, *105*, 111 – 120. doi: http://dx.doi.org/10.1016/j.compag.2014.04.011.

Bac, C. W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, . doi: 10.1002/rob.21709.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014b). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Bac, C. W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & van Henten, E. J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, *146*, 85 – 97. doi: https://doi.org/10.1016/j.biosystemseng.2015.07.004. Special Issue: Advances in Robotic Agriculture for Crops.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PP*, 1–1. doi: 10.1109/TPAMI.2016.2644615.

Barth, R., Hemming, J., & van Henten, E. J. (2016). Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, *146*, 71 – 84. doi: https://doi.org/10.1016/j.biosystemseng.2015.12.001. Special Issue: Advances in Robotic Agriculture for Crops.

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. V. (2017). Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture*, . doi: https://doi.org/10.1016/j.compag.2017.11.040.

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. V. (2018). Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Computers and Electronics in Agriculture*, *144*, 284 – 296. doi: https://doi.org/10.1016/j.compag.2017.12.001.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR*, *abs/1206.5533*.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*, 679–698. doi: 10.1109/TPAMI.1986.4767851.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, .

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Gabriela Csurka (Xerox Research Centre Europe), F. P. X. X. G., Diane Larlus (2013). What is a good evaluation measure for semantic segmentation? In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 1263–1284. doi: 10.1109/TKDE.2008.239.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. *CoRR*, *abs/1703.06870*. `arXiv:1703.06870`.

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III* (pp. 346–361). Cham: Springer International Publishing. doi: 10.1007/978-3-319-10578-9_23.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, *abs/1502.01852*.

Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., Bontsema, J., & Os, E. V. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering*, *86*, 305 – 313. doi: http://dx.doi.org/10.1016/j.biosystemseng.2003.08.002.

Illingworth, J., & Kittler, J. (1988). A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, *44*, 87 – 116. doi: http://dx.doi.org/10.1016/S0734-189X(88)80033-1.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, .

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Lehnert, C., English, A., McCool, C., Tow, A. W., & Perez, T. (2017). Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, *2*, 872–879. doi: 10.1109/LRA.2017.2655622.

Mishkin, D., & Matas, J. (2015). All you need is a good init. *CoRR*, *abs/1511.06422*.

MVTech (2016). Halcon. url: `http://www.halcon.com/`.

Papandreou, G., Chen, L.-C., Murphy, K., & Yuille, A. L. (2015). Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*.

Petkovic, T., & Loncaric, S. (2015). An extension to hough transform based on gradient orientation. *CoRR*, *abs/1510.04863*.

Pound, M. P., Atkinson, J. A., Wells, D. M., Pridmore, T. P., & French, A. P. (2017). Deep learning for multi-task plant phenotyping. *bioRxiv*, . doi: 10.1101/204552. `arXiv:https://www.biorxiv.org/content/early/2017/10/17/204552.full.pdf`.

Shapiro, D. (2016). Accelerating the race to autonomous cars. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '16 (pp. 415–415). New York, NY, USA: ACM. doi: 10.1145/2939672.2945360.

Sodhi, P., Vijayarangan, S., & Wettergreen, D. (2017). In-field segmentation and identification of plant structures using 3d imaging. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Wehrens, R. (2010). Self-organising maps for image segmentation. In A. Fink, B. Lausen, W. Seidel, & A. Ultsch (Eds.), *Proceedings of the 32nd Annual Conference of the Gesellschaft für Klassifikation.* (pp. 373–383). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-01044-6_34.

Zeng, A., Yu, K., Song, S., Suo, D., Jr., E. W., Rodriguez, A., & Xiao, J. (2016). Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. *CoRR*, *abs/1609.09475*.

Zhu, H., Meng, F., Cai, J., & Lu, S. (2016). Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, *34*, 12 – 27. doi: http://dx.doi.org/10.1016/j.jvcir.2015.10.012.

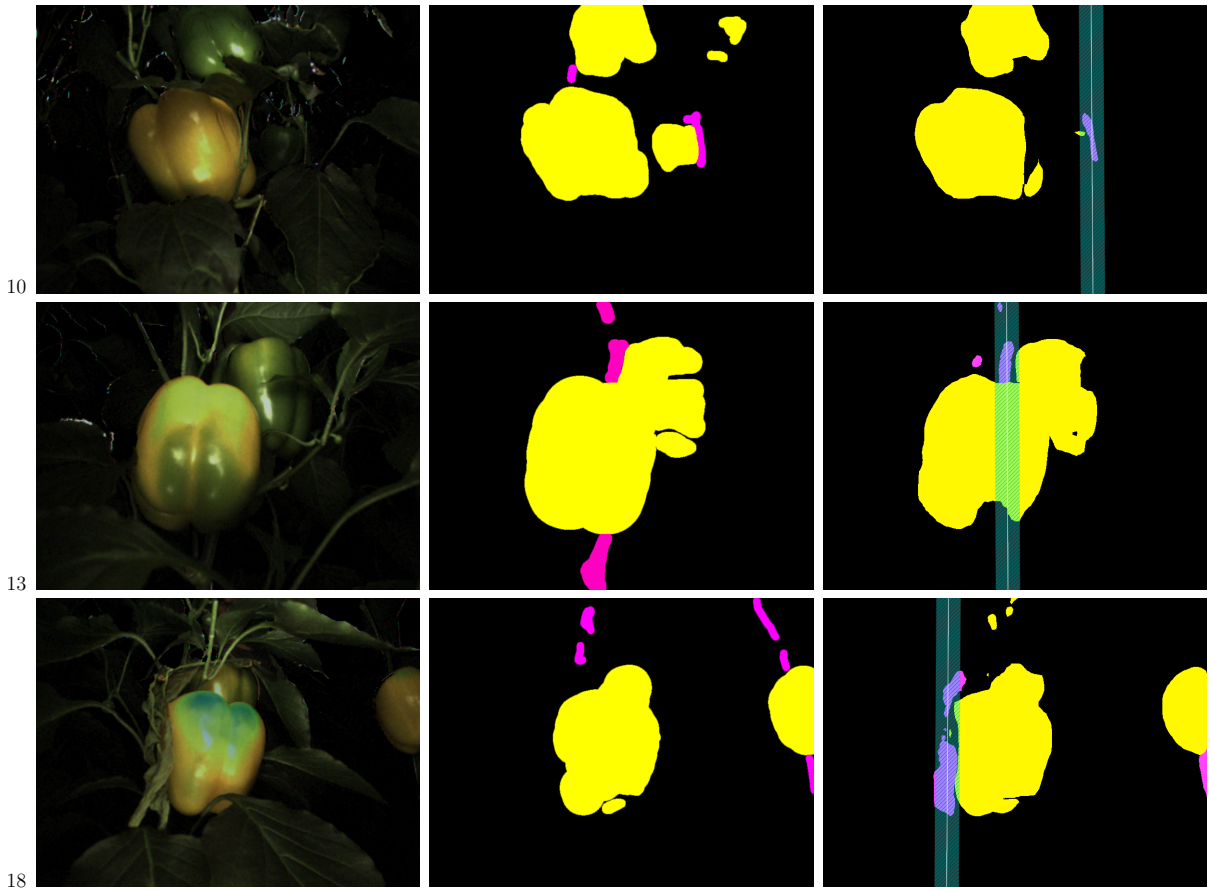
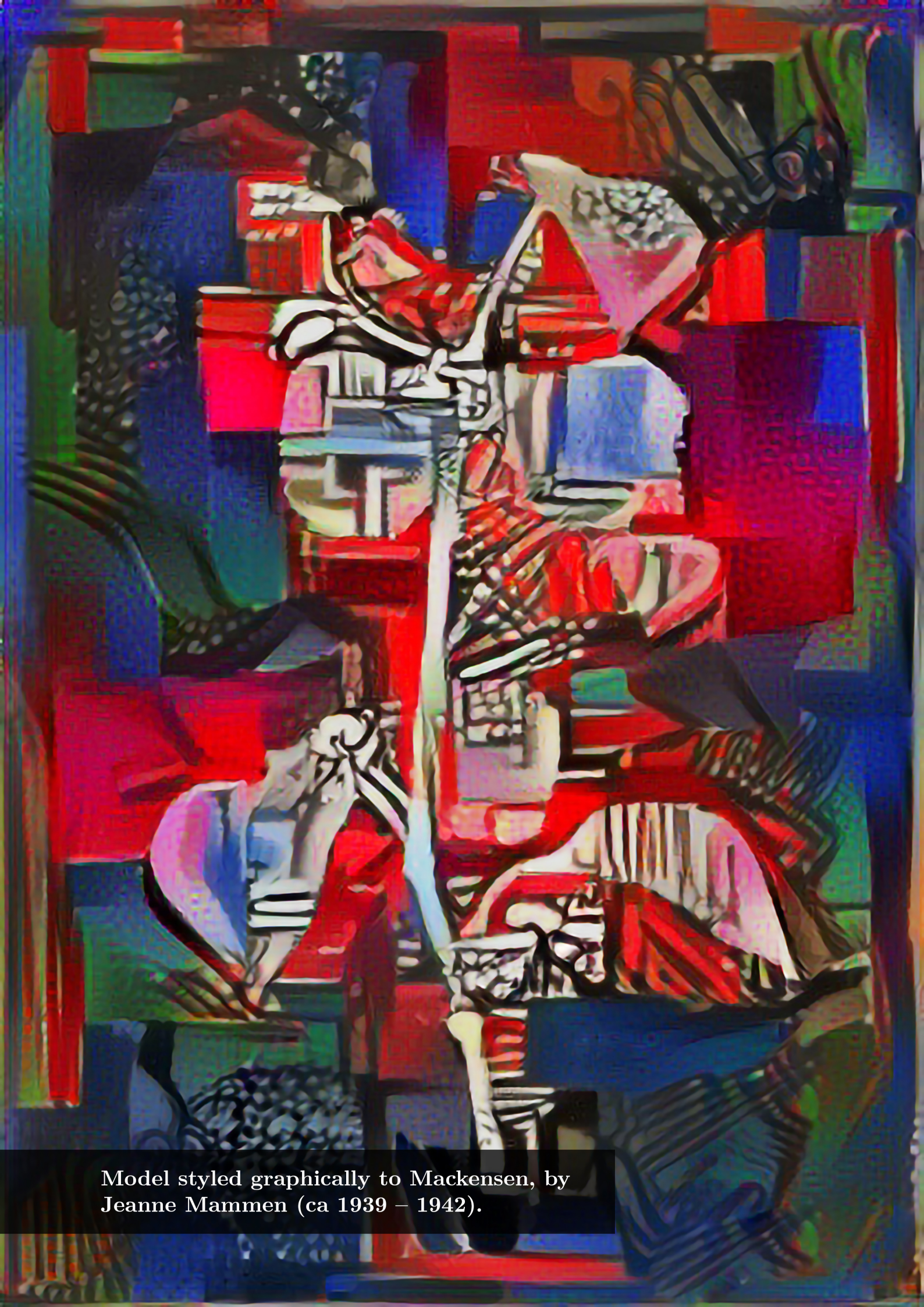


Figure 6.18: Images (left) 10, 13, 18 of Task II results for the unmodified greenhouse condition with its ground truth class annotation (middle) and semantic segmentation (right) with estimated stem as a white line and striped overlay of an average stem width of 50 pixels. Class labels: ● background, ● stems (including wires) and ● fruit.



Model styled graphically to Mackensen, by Jeanne Mammen (ca 1939 – 1942).

Chapter 7

General Discussion, Reflection and Recommendations

7.1 General Discussion, Reflection and Recommendations

'Sowing artificial intelligence in agriculture' has been a metaphor for the work presented in this thesis, in the hope that intelligent robotics will eventually sprout and mature to partially address the pressure posed on the field of agriculture by current and future societal challenges in the food chain. In Chapters 2 through 6, principles were researched for more advanced agricultural systems, by seeking improvements in the computer vision domain.

The objective of this work was to advance the technology of agricultural robotics, specifically in the scope of computer vision for the task of harvesting in high-tech greenhouses. The underlying aim was to cope with a main performance bottleneck caused by the variation in the crop and environment in order to improve harvest success rates and cycle times. To achieve this objective, methods were explored such as visual servo control and eye-in-hand sensing, based on state-of-the-art machine learning. To meet the inherent training data requirements of the latter, plant modelling and data synthesis were researched as well. Explicitly,

from the hypotheses that:

'Agrobotics has not yet matured primarily because its sensing and control cannot yet handle all crop variation.'

&

'Computer vision is the most effective sensing solution to handle the crop variation.',

followed the main research question:

'Which novel and proven vision principles can improve agrobotics performance by coping with the crop variation and interaction?',

of which it was hypothesised that:

'State-of-the-art machine learning, notably deep learning based semantic segmentation, could improve sensing crop variation for agrobotics.'

&

'Visual servo control can help to sense more of the crop variation and provide feedback to make corrections for interaction with the crop.'

In this section, it is evaluated how the presented work in the context of the research question and corresponding hypotheses, contributes to increasing the level of smart mechanisation in agriculture, other or even higher level domains. A more detailed discussion of each chapter's contribution to the overarching aim of this thesis will be also provided, after which a description follows of future challenges and recommendations.

Regarding the topic of eye-in-hand sensing and visual servo control (explored in Chapter 2), the work in this thesis aimed to encapsulate proven sub-systems into a coherent, well documented framework that could be applied to any agricultural robotics task. Indeed it was shown that through the Robotic Operating System and ViSP libraries, a structure could be built that can act as a backbone for robotic harvesting using eye-in-hand sensing and visual servo control. After this framework and paradigm was formed, the use-case of a sweet pepper harvesting robot adopted significant parts thereof in the *Sweeper* project. The framework helped to cope with the variation in the crop and environment, because it allowed to scan and approach the crop using multiple viewpoints, thereby reducing uncertainty with the increased amount of information. Preliminary qualitative field test results show that indeed the positioning accuracy of the end-effector was improved compared to previous approaches, by ensuring that the actions of the manipulator were adjusted when the crop moved during the fruit harvest approach. Furthermore, additional sensing feedback later in the harvest cycle, was able to partially correct errors made in the initial sensing of fruit detection. For these reasons visual servo control can help to sense more of the crop variation and provide feedback to make corrections for interaction with the crop.

Because a widely used and modular open source Robotic Operating System software package was chosen as a basis, the framework could also be adopted for robotic applications outside the domain of agriculture, e.g. grasping of objects in warehouses. Even so, the novelty of the constructed architecture must not be overstated, as the framework presented in Chapter 2 is one of many possible and plausible solutions to solve automation of such tasks. Figuratively, the main contribution of the work should be considered as the addition of a new spice flavour blend, made with previously proven and high quality ingredients, that can act as the backbone of a range of recipes. However, even ROS started with and introduced flaws after 10 years of additive development and might soon reach its expiration date. Hopefully, other open frameworks alike Orocos (Bruyninckx, 2001) or ROS 2.0 will be able to continue to provide progress in the domain of robot control.

The framework could be improved by the addition of a component to construct a world model, based on all sensing information combined. Currently, the sensing and visual

servo control only uses a single observation of the scene to act or replan. Integrating this knowledge with previous sensing information could further allow for mechanisms to cope with variation in the crop. This could for example be performed by deep learning, using Generative Query Networks (GQN), that internalise a representation of the scene (Eslami et al., 2018). An GQN takes input images of a scene taken from different viewpoints, constructs an internal representation, and uses this representation to predict the appearance of that scene from previously unobserved viewpoints.

Chapter 4 aimed to tackle the topic of sensing in agriculture, specifically by optimising how the robot divides the visual scene into meaningful regions. The efforts in this work introduced and tailored a state-of-the-art machine learning based image segmentation using convolutional neural networks geared towards achieving highly detailed segmentation of the highly variable crops and environments observed in agricultural settings. The level of detail that was achieved, i.e. on a plant part level, has been an unique addition to the field. It enables computer vision approaches that can determine which plant part is where in the image, on a per pixel level. This application of machine learning in sensing could have a significant impact on agricultural robotics, especially as it suggests that future image recognition algorithms could become less reliant on manual feature crafting. Instead, algorithms could automatically learn significant features based on large sets of high quality annotated data. For these reasons, the deep learning based semantic segmentation helped to improve sensing the variation for agrobotics.

The application of convolutional neural networks has reduced the sensing error early in the action planning phase of the harvesting pipeline, as well as in the execution of the planned action, by improving the interpretation of visual feedback. Furthermore, the approach makes systems more robust to variation of the crop and environment in two ways. First, because plant parts can now be discerned on a high resolution, the *planning* component of the robot has access to more accurate information to navigate a broader range of unique situations. Second, the resulting classification is usually more robust for variation (e.g. illumination or shape) compared to manually crafted vision algorithms.

A main requirement of state-of-the-art machine learning-based image segmentation is the availability of large high-quality annotated datasets for training of semantic plant part extraction. This work (Chapter 3) presents a synthesis method to automatically generate such a dataset based on empirical data on crop dimensions and variation. The synthesis method contributes an approach to empirically model the variation in the crop and environment and to render visually realistic agricultural scenes. Moreover, with the further transformation of synthetic images to the empirical domain using cycle generative

adversarial networks, Chapter 5 provided a novel bootstrapping approach to boost the classification performance and to further reduce the reliance on manual annotation.

Application of the semantic plant part image segmentation approach presented in Chapter 4 is not necessarily restricted to agricultural robotics alone. For example, in the domain of plant phenotyping, where physical traits of plants are quantified, an accurate and detailed plant part segmentation could be a prior for high throughput geometric measuring systems. For example, fruit heights or peduncle lengths can be derived using this approach when combined with registered depth information. Such a phenotyping system could for example facilitate more efficient trait selection for species with higher yields, more nutrients or resilience against local climate change.

In other domains such as medicine, a similarly large biological variation is observed as in agriculture. Methods such as provided in this work (Chapter 3 through 5) could be applied in the medical imaging domain to improve the segmentation of body parts or tissues. Indeed, work has already headed into that direction, e.g. by using generative adversarial networks to improve the realism of computed tomography (Wolterink et al., 2017).

For the general field of computer vision, the use of synthetic images transformed to the empirical domain in combination with a synthetic ground truth, could be a first step towards a novel method where only synthetic data is required for classifier training and the necessity of manual annotation is eliminated. Such an approach could be coined *'synthetically supervised'*. However, this does not imply that the learning intrinsically becomes *unsupervised*. Nor does it entail that no manual effort is needed, unless the synthesis of data is similarly automated. Future research could be dedicated to automatically deriving plant models and render parameters from images to create synthetic learning datasets. In a sense, this could result in a strange loop; *'a self-perceiving, self-inventing, locked-in mirage, that can become a little miracle of self-reference'*, as Douglas Hofstadter might say (Hofstadter, 2007).

Taken together, regarding the main research question ***'Which novel and proven vision principles can improve agrobotics performance by coping with the crop variation and interaction?'***, it can be concluded that through the combined approaches presented in this work, important progress has been been made to find such vision principles. The eye-in-hand sensing and visual servo control framework (Chapter 2) ensures that the feedback loop can correct for earlier errors and is able to average out uncertainty caused by the variation of the crop. The deep learning approach for semantic plant

part segmentation (Chapter 2 through 5) established a new landmark of performance for agricultural computer vision, providing a new level of detail of the classification that allows for improved sensing the variation in the scene. The synthetic image generation (Chapter 3) reduces the main requirement of dataset size and the need of manual annotation, but moreover provides a way to inject high level domain knowledge for improved learning.

With these vision principles described in this thesis, mechanical technology for agriculture has been advanced with another modest step along the spectrum towards truly intelligent agricultural robotics. Using these vision principles, such systems should be able to better cope with its main bottleneck of sensing and making sense of the variation they could possibly encounter in an agricultural setting. Regardless, there remains work to be done, especially in bringing the current state-of-the-art knowledge to a practical and mature market ready product.

In the following sections, a more detailed discussion and reflection is given on the methods and results for each individual chapter of this work, integrating from a higher bird's eye view the contribution of each chapter to the the goals of this thesis as a whole.

7.2 Eye-in-hand Sensing and Visual Servo Control

The goal of this part of the research was to design a framework for eye-in-hand sensing and motion control to facilitate the development of new robotic harvest applications, especially in dense crops. One of the key aspects of the framework design was to add a high degree of implementation flexibility to meet any specific use-case constraints. The secondary aim of our research was to demonstrate a framework implementation for a dense sweet-pepper use-case. The example implementation shows the framework can be applied for sweet-pepper with custom hardware. The added sensing functionality of the 3D scene reconstruction shows the extendibility of the framework. As technology advances, the coded framework itself will be outdated in time, as the backbone ROS evolves in ROS 2.0 and other libraries might get deprecated. Regardless, given the modular approach, the framework allows for partial updates such that the underlying paradigm and design can continue to be employed.

Results indicated that the framework can be successfully used for solving sensing and robot motion control in a dense crop from visual information from the end-effector, although this was not quantitatively further explored in Chapter 2. Whilst the artificial crop setup provided a reasonable reflection of the occlusion problems faced in everyday practice, it remained a simplification of the real crop situation. Occlusions from stems and fruit clusters were not taken into account, as well as neighbouring plant stems. It was suggested as future research that the implementation should also be tested quantitatively under greenhouse conditions.

The latter was to be investigated in the follow-up project *Sweeper* (Wageningen University and Research, 2018). However, the framework was not directly and fully applied, although the paradigm of eye-hand-sensing and visual servo control was incorporated and a similar high-level control structure was used. The main difference was that the ViSP library was omitted and replaced by a custom visual servo control algorithm because it appeared a simple closed loop control strategy sufficed for only the motion of the last 20 cm towards the target.

The implemented paradigm for a sweet-pepper harvesting robot was quantitatively tested in the greenhouse (June 2018). Results are just out of the scope of this thesis, but could thereafter be compared to the remote sensing and point-to-point motion control implementation tested in the *Crops* project (Bac et al., 2017). However, between the *Crops* and *Sweeper* tests, other components were also updated, making it therefore hard

to conclude if the paradigm shift influenced performance directly. Nonetheless, it can also be evaluated qualitatively, if for example visual servo control solves issues regarding motions of the target whilst harvesting.

In Chapter 2, the use-case implementation successfully used a single geometric point as feature for the visual servo control. For other use-cases that require a fixed end-pose, multiple geometric features could be used. However, this requires accurate depth information which was not available at the time. In the *Sweeper* project, the Fotonic time-of-flight sensor could provide such depth information. Future research could therefore also include visual servo control to a specific end-pose that is optimised for the end-effector and task, e.g. harvesting at the peduncle under a certain angle similar to (Lehnert et al., 2017).

Although visual servo control and eye-in-hand sensing have their benefits, a point-to-point motion planning without online correction could still be a viable solution for some harvest robotics applications in general, if for example the end-effector mechanically could cope with a greater positioning error or does not interact much with the crop. Also, sensing that is not performed inside the end-effector could still be an option (e.g. to save space), as long as it would run in parallel with the robot's harvest motions to save time. Such systems could implement sensing that works one step or plant ahead of the robot. For this, it is key that detected targets remain stationary between harvest cycles and that proper calibration has been performed to ensure that transformations from the sensing frame to the robot frame are accurate in the next step.

Regardless, the benefits of eye-in-hand sensing and visual servo control remain. Motions can be corrected whilst approaching the target, which may have moved due to interaction of the robot and the crop. Furthermore, eye-in-hand sensing might simplify the overall task because sensing and acting are closer integrated within the system, omitting additional calibration and frame transformation challenges. However, eye-in-hand sensing does increase the size of the end-effector, especially when relatively bulky 3D time-of-flight sensors are used. This reduces available space for grasping and harvest mechanics, potentially decreasing the harvest success rate. For future research it is therefore suggested to miniaturise the image acquisition to small RGB cameras. Depth perception should then be achieved by simultaneous localisation and mapping (Engel et al., 2014) and/or deep stereo learning approaches (Godard et al., 2017).

7.3 Synthetic Image Generation

Chapter 3 described synthesis methods for large-scale semantic image segmentation datasets of agricultural scenes. The motive for this work was to ensure that the dataset size requirement could be met for state-of-the-art methods, primarily convolutional neural networks, whilst reducing the need of manual annotations that are often infeasible to obtain. The latent objective was to eventually bridge the performance gap between state-of-the art computer vision performance and computer vision in the agricultural robotics domain, by facilitating the use of such deep learning approaches.

Regarding this motive however, the dataset size requirement for such methods is dependent on the complexity of the classification task, which in turn dictates the complexity of the neural network that should be used. For example, the detection of only yellow ripe pepper parts in an overall green image, would require a simple convolutional neural network with relatively small amount of parameters to learn. In turn, this would also require only a relatively small dataset size. However, when a multitude of plant parts are to be recognised, a more complex model with more parameters is to be learned, also requiring more training data.

At the time of developing the data synthesis methods, it was unknown what the complexity of the future task would be, let alone the complexity of the neural network and its corresponding dataset size requirement. Chapter 3 was the first step to find an answer to this question by first providing methods for large dataset synthesis that included a relatively high amount of complexity. In Chapter 4 then a state-of-the-art segmentation solution was implemented to verify the dataset size and neural network complexity requirements, for the specific task of plant part segmentation.

Nevertheless, the dataset size and complexity in Chapter 3 were arbitrarily chosen. Although rendering 10,500 images took considerable time on a supercomputer, one might argue that other datasets such as ImageNet has three orders of magnitude more labeled images. However, one must note these images were labeled on a high semantic level, not a per-pixel part level. If we look at the benchmark PASCAL VOC dataset for segmentation, a comparable amount of 10,000 labeled images are available, with up to 20 classes (Everingham et al., 2012).

The complexity of the dataset could have been higher by introducing more (sub-)classes (e.g. stem nodes or discerning side-shoots from leaf stems). Furthermore, additional plant parameters could have been measured and implemented in the model (e.g. leaf curl, fruit width and height or peduncle thickness). This would not only have made the plant model

more realistic, but it would introduce more visual variance in the dataset when each plant is randomly generated. Regarding the latter, one might argue that after 10,500 images sampled from the same model distribution, some repetition in local and global visual features might occur. Refining the model with more plant parameters would allow to synthesise a larger distinct dataset.

The question can be asked if the manual modelling efforts counteract the intention of saving manual annotation efforts. Indeed data synthesis also takes substantial effort, e.g. measuring plant parameters, modelling the plant and scene. However, after the initial investment in obtaining the model, it becomes possible to generate new datasets relatively quickly under different conditions (e.g. illumination, growth architecture, camera parameters, added depth, stereo or hyper-spectral information) that allows for rapid prototyping towards new applications.

Regarding the visual repetition of visual features, one would expect that given a convolutional neural network model that is deep enough to cope with large datasets, a point would be reached where more data sampled from the same distribution would not lead to increased learning performance. Indeed this is what was observed in Chapter 4, where the most complex model did not see learning improvement beyond around 2,000 images given this particular dataset. However, it might also be the case that the model was lacking complexity to be able to learn beyond 2,000 images. As future research, the performance effect of increased model complexity should be investigated. This would determine if either the repetition of visual features in the data or the complexity of the model is constraining the learning.

Future research should also be directed to the segmentation of instances, to provide masks for each individual plant part. The Mask R-CNN architecture (He et al., 2017a) could be explored for this. The main benefit of such an approach is that agricultural robotics would be more capable to target individual plant parts. For example, currently with our method fruit clusters are segmented as a single mask whereas instance segmentation would allow for de-clustering of fruit. This would require a different type of ground truth however, where each plant part instance has its own mask. For our synthetic dataset, this requires some adaptations to split all plant part models in unique objects and re-rendering the ground truth frames. Currently, efforts are made to realise this amended ground truth.

7.4 Plant Part Segmentation

With Chapter 3 a foundation for data synthesis in agriculture has been created. Further evidence whether such datasets are useful to reduce the need of manual annotation and bridge the performance gap in agricultural computer vision, was gathered in Chapter 4.

Regarding the methods, the choice for using DeepLab V1 and V2 architectures was based on open source availability and being the best performing models for segmentation at the time. However, within the next 2 years after our research, 25 new methods have since superseded the models used in Chapter 4 on the PASCAL VOC segmentation benchmark. Currently, the best performing method on this benchmark is the DeepLab V3 + JFT architecture (Chen et al., 2018), with about a 10% absolute increase in mean IOU performance[1]. This underlines the rapid development within the domain of computer vision and might indicate our reported results in this thesis can already be further improved upon.

In Chapter 4, seven experiments were performed, each to validate or find evidence for a certain part for the main hypotheses. However, given that the model was fixated after the first experiment, the chosen architecture might have played a large role in shaping the conclusions. Possibly, different architectures (e.g. the newer DeepLab V3 models) would have had different learning and classification characteristics. For example, it was shown that without synthetic images, uncommon classes were not recognised. However, a different architecture that could cope with class imbalance might not have needed synthetic images to achieve comparable performance. Hence, it remains important to keep asking the same questions when new architectures are explored in the future, especially regarding the validity of using synthetic data. Regardless, our methods have shown a way to approach these questions that can be used in such experiments.

One might argue that less complex computer vision methods for plant part recognition might also have met the requirements (such as posed in Chapter 6) for the task of harvesting. Indeed, bounding box detections through methods such as fast region-based convolutional networks (Girshick, 2015) might have sufficed and furthermore could have decreased the processing time with an order of magnitude (about 20 ms versus 200 ms). However, part of the aim of our work was to provide also methods for other tasks in agri-

[1] http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?cls=mean&challengeid=11&compid=6&submid=15347

cultural robotics, e.g. phenotyping or localised spraying. For such tasks, a more refined detection than bounding boxes is required like our approach using a per-pixel classification. Regarding phenotyping, one of the goals is to obtain accurate dimensions of plant parts that cannot otherwise be derived from bounding boxes alone. When using a per-pixel classification together with registered depth, this would become possible in future research.

Unfortunately, a downside of the per-pixel classification approach to recognise plant parts in this work is that instances of parts are not recognised. In other words, the model classifies all unique plant parts of the same class with the same label. This implies that only disconnected and non-occluded parts could therefore be instanced, for example by using post-processing algorithms. In contrast, bounding box approaches do provide instances, but again lack the level of classification detail. Fortunately, both methods can be combined using the most recent deep learning approaches, like mask-rcnn (He et al., 2017b) and this is suggested as future research. In Figure 7.1 preliminary results of these efforts, falling just outside the scope and timeframe of this thesis, are shown.

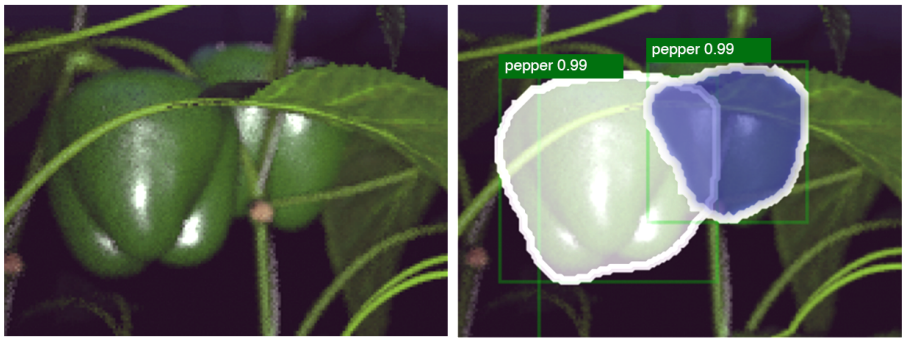


Figure 7.1: Instance segmentation and object completion of two partially overlapping peppers (right image) from input color image (left) using mask-rcnn.

In general, the field of deep learning for computer vision has proven to move relatively fast, compared to previous progression in the image recognition domain. In the new field of deep learning, it is not unconventional that methods are superseded within months and paradigm shifts can occur whilst one has still been implementing the former state-of-the-art. Projecting these developments to the future, some of the work in this thesis might get caught up before it has been applied to a broader set of challenges. For example, data synthesis was explored to reduce the manual annotation requirement whilst still ensuring large datasets that convolutional neural networks need for training. However, new

paradigms like unsupervised deep learning (Higgins et al., 2016) might (hopefully) make data synthesis methods superfluous. Nonetheless, data synthesis can still be a valuable tool to obtain image sets that strictly adhere to controllable experimental parameters. For example, using data synthesis, one might be able evaluate the effect of leaf size on fruit recognition performance to optimise the crop for robotisation through future plant breeding. Obtaining a dataset manually for the latter would be tedious, if not impossible.

7.5 Improved Segmentation Performance by Optimising Realism of Synthetic Images

Although the synthetic images from Chapter 3 have already proven useful for increasing overall performance and the recognition of uncommon classes, it must be admitted that the realism of the synthetic images could be improved. Quantitatively, we showed that a gap remained in average color distributions per plant part. Qualitatively, although the synthetic images look realistic to the human eye, differences in local texture with the empirical images can be observed. In Chapter 5 the improvement in realism of the synthetic images was explored by applying cycle generative adversarial networks. Furthermore, the effect of the translated images on learning was investigated.

The selection for cycle-GAN was made for its ability to transform images from one domain to another, without the need of geometrical pairing that other generative adversarial methods required. This enabled that the same synthetic ground truth could be used after transformation. Other deep learning approaches could also have been chosen to obtain geometrically consistent domain translations, e.g. neural style transfer (Gatys et al., 2015). However, the main advantage of cycle-GAN is that the geometric similarity is actively used as a constraint through the loss function, whereas other methods are not bounded by this. This resulted in more geometric preservation using cycle-GAN. However, approaches like neural style transfer do provide qualitatively interesting artistic transformations, as can be seen at the beginning of each chapter in this work.

The strong geometrical constraint could also be considered a downside however. In the case of plant modelling, when the plant model is not accurate, cycle-GAN won't be able to correct these mistakes by projecting the real plant parameter distribution from the empirical domain to the synthetic one. The transformation is purely local and focussed on color and texture. Future research should investigate if it is possible to translate also

the geometric distribution. However, it must then also be tracked how each part changed in order to translate the ground truth accordingly.

A requirement for the method that was used, is that both domains should have the same objects or parts. Additional objects will otherwise be mixed into the translation, resulting in artefacts as the network cannot seem to place the color and texture sensibly to a target image lacking that type of object. In the datasets used for this work, the blue suction cup in the empirical images had to be removed. Otherwise, this resulted in blue peppers after translating the synthetic images.

Unfortunately, the transformation could only be applied to relatively low resolution images (424x424 pixels) due to memory constraints. However, efforts by others are made for conditional generative adversarial networks to also be applied on higher resolution images (Wang et al., 2017).

Although another aim with this research was to completely remove the requirement of manual annotations, this was not fully realised. The hypothesis was confirmed that without any empirical image fine-tuning, learning can be improved with translated images (IOU=0.31), which constituted a 55% increase compared to just synthetic images (IOU=0.20). However, by fine-tuning with empirical data the performance was still highest at IOU=0.52. A possible explanation might be that indeed the geometric similarity of the synthetic data was not yet sufficient for bootstrapping without empirical images. Future research might be aimed at validating this hypothesis by increasing the geometric similarity, e.g. by introducing a more comprehensive set of plant parameters in the model or methods that can translate the synthetic images to the geometric empirical domain.

7.6 Estimating Angles between Plant Parts

In the final core Chapter 6, the previous results from theoretical experiments were put into practical use for a sweet-pepper harvesting robot. A method for estimating the angle between the plant stem and fruit was presented to support the grasp pose optimisation in a sweet-pepper harvesting robot.

The incentive for this research comes from the *Sweeper* project's aim to also perform obstacle avoidance. Although grasp pose optimisation seems to have little connection with obstacle avoidance at first, there is a relation. Given that the end-effector performs a visual servo control motion towards the fruit, it must be made sure the stem is not in the path of the grasping by the robot. In such case, the stem can be considered an obstacle.

The scope of the research was focussed on four sub-tasks of the robot's harvest sequence, ranging from image segmentation to the estimation of the angle between stem and fruit. Unfortunately, at the time the robot prototype was not yet fully functional. Therefore, further steps like visual servo control towards the fruit and grasping and cutting thereof could not integrally be tested. Nonetheless, given the requirements of the end-effector, it was still possible to evaluate each task individually.

As depth information was also not available at the time, the angle estimation required two concurrent viewpoints that could be considered a form of pragmatic stereo vision. Although this is an effective way to cope with the lack of information, it is not the most elegant solution. Preferably, only a single image acquisition action would be required to obtain the angle of interest, for example by using a 3D sensor. Not only would this save time, but it would allow for multiple consecutive estimations during the fruit scanning phase of the robot. These multiple measurements could then be integrated to a median value for a more accurate estimation, possible averaging out occlusions that make an individual measurement less reliable.

Depth information might also make the calculation of the angle more precise, as currently a constant distance value between the fruit and stem is assumed that could otherwise be measured. Furthermore, it would allow to estimate angles not only at the calibrated 20 cm distance, but from any robot pose.

The method in Chapter 6 would allow up to 73% of the fruit to be harvestable in the first try. However, this is only a single factor determining harvest success. Depending on the performance of other tasks like grasping and cutting thereafter, overall performance would

likely be lower. Simplifying the crop proved not to increase the performance however. It might be investigated if multiple views can improve the accuracy of the angle estimation. Additionally, the end-effector might be mechanically improved to allow for a larger angle estimation error, although there is an upper limit where mechanics can pick up the slack of the computer vision performance.

The current work was focussed on the task of fruit grasping and hence was limited to one pair of plant parts. Future research should be dedicated to estimating angles between any set of connected plant parts. By using 3D information, this would enable to map the geometry of the plant. Measuring geometry in this manner is a pre-requisite for automated phenotyping and can be used for further improving plant models as used in Chapter 3.

Regarding vision guided manipulation, future research is likely headed towards end-to-end training of deep visuomotor policies (Levine et al., 2016). Given a color or segmented image, such methods can directly learn grasping strategies. Our approach still required to manually craft perception and control software. Although shown to work for a specific use-case, this remains a low-level of artificial intelligence that cannot cope with high crop variation or generalise well. In the near future, end-to-end deep methods will likely supersede our methods in that regard.

7.7 Recommendations and Future Directions

The domain of agricultural robotics currently is positioned on the intersection of late stage fundamental research and the practical application in the market. In terms of the *'Technological Readiness Levels'*, many prototypes are somewhere on level 8 (system development) to 9 (system test in operational environment, launch and proven operations). Regarding the sweet-pepper harvesting robot, the first part of level 9 was achieved by tests in the real greenhouse during the summer of 2018.

However, the pepper harvesting robot and other systems have not yet launched or have proven themselves in continuous field operations, which could be considered the last requirements to call an agrobotics system mature. In this thesis it was argued that in order to make the transition towards a mature level of agrobotics, more advanced intelligent systems by means of computer vision should be developed to cope with the inherent variation in the crop and environment. Notwithstanding that this has been partially addressed by this thesis, there remains room for more improvement on this topic and other subjects

to ensure agrobotics moves away from the laboratory and onto the greenhouse or field of the farmer.

First, regarding the future direction of sensing, unsupervised machine learning paired with transfer learning would be an ideal approach to pursue to lose the need of annotation or data synthesis. Using such methods, what then remains would be to collect a reasonable amount of images of the target domain, which can already be fairly easily be automated.

Sensing is not restricted by computer vision alone, but also includes other modalities like touch (Tenzer et al., 2014). Hence, future efforts should also integrate these types of advanced sensing, possibly providing feedback during the grasping and cutting, when vision can no longer provide much information.

After sensing, the domain of end-effector design is the next most critical module to further optimise to increase the performance of agricultural robotics. Coping with the variety in shapes and poses can only be partially solved by proper recognition using computer vision, after which the tool to grasp and cut the target must be able to physically manage the geometrical variation. The pursuit of soft robotics research might be good suggestion in order for end-effectors to become more compliant to handle the fruit and vegetable variation. Furthermore, soft robotics can also reduce the risk of damages compared to when hard materials like steel are used.

Another vital element to mature agricultural robotics, might be the awareness that humans will initially be not out of the loop in two ways. First, no robot in the beginning will be able to perform with a 100% success rate. If we assume a robot can automate about half of the tasks correctly (without damage to the remainder) then still about half of the original labour is needed. However, this is not a wrong starting point for the introduction of agrobotics, as long as the combination with humand is economically viable or adds other benefits for the grower, like easing labour demands during peak production. Second, humans can be in the loop directly for the cases where the robot fails a task. For example, remote teleoperation via cloud services can further increase the success rate whilst labour does not have to be performed in confined spaces and harsh climate conditions.

The notion that an agrobotic system will not be working alone, can be further projected to the whole logistics of the greenhouse, field or orchard. Creating a system that can automatically perform a task like harvesting is only part of the story. The fruit or vegetable needs to be placed in a bin, transported to a central location, sorted, packed, stored and made transport ready. Hence, agrobotics systems need to be adapted to connect seam-

lessly to this infrastructure, which requires an integrated level of autonomous planning of a fleet of robots. Without such implementations, agrobotics cannot mature.

Also it might be important to research adaptations in the greenhouse, field or orchard growing architectures to afford robotisation. For example for sweet-pepper, a single stem row system would be preferred to optimise robot reachability as compared to the double stem row standard used today. In line with adapting the growing architectures, is the adaptation of plant architectures should be investigated to facilitate robotic tasks. Some crop varieties for example have a more open growing structure, meaning less occlusions for the computer vision to handle. Or specific plant parts like peduncles grow in different ways between crop varieties, for example in sweet-pepper there are varieties with long and short peduncles, which each require a different end-effector design. To find the optimal plant architectures, plant breeding for robotics will become an important direction to pursue.

Apart from the technological challenges, there are also organisational ones. To really progress in the agricultural robotics domain, the work needs shift away from research institutes and universities towards a more dedicated industry. As seen in the projects *Crops* and *Sweeper*, which by all means were very effective in maturing individual key technologies, there was the shortcoming to effectively integrate towards a single coherent system. This was mostly caused by the research being geographically dislocated in different groups and internal diversification of the research, i.e. researchers working on multiple parallel projects. The solution would be to continue in a dedicated, market oriented group that can focus fully on demarcated goals to *'sow artificial intelligence in agriculture'*.

Last but not least, economics play a conclusive role for viable agrobotics. Although technically a lot is possible, the entire picture of greenhouse, field or orchard automation should be competitive compared to the current situation. This implies the cost and the cycle time of the robots should be low enough for them to be adapted.

Concluding, these recommendations should be considered as a guideline to future directions of the agrobotics research and development to ensure that the next level of agricultural mechanisation will be fully reached.

7.8 Epilogue

Agricultural robotics can make an important contribution to ensuring the quality of meals and nourishment of a growing population. Furthermore, it can become an important technological tool for addressing societal issues surrounding food production in developed nations, e.g. by solving labour problems. Through this, it could help further improve upon the quality of life and contribute to reaching an exemplary equilibrium of sustainable agricultural production. In time, everyone might be able to harvest prosperity due to A.I. in agriculture. Importantly, its primary contribution will likely to be found in crops that provide other essential nutrients, e.g. fruits and vegetables, for meeting the future caloric demand of the growing population technological development alone is unlikely to be an adequate solution.

Given the current technological disparity between developed and developing countries, another partial solution to the increased demand on the food chain might present itself; the technological level in developed countries could be projected to developing countries to increase their efficiency, in turn reducing the pressure on agriculture. However, for developing countries the assimilation of technical solutions remains a challenge and is, therefore, currently limited by factors such as environmental technology compatibility, resource availability to facilitate the adoption of technology, investment space and government policies (Onwude et al., 2016). Nonetheless, when in time these issues are addressed, technological projection could become part of the answer.

Preliminary data analysis of the field test with the fully integrated sweet-pepper harvesting robot (see Figure 7.2), provided the quantitative results of 62% and 31% harvesting success in a modified crop and unmodified commercial crop respectively (under the assumption that a single stem row cropping system is used). Compared to the previous performance of the earlier sweet-pepper harvesting robot where the harvest success rates were 26% and 6% respectively (Bac et al., 2017), this is a significant improvement. Also the average cycle time per fruit was improved from 94 seconds to 24, although the latter was not yet optimised.

In all, machine learning assisted agricultural systems using vision principles for sensing and visual feedback provide plenty of advances, but also open lots of opportunities and challenges for (entrepreneurial) scientists.



Figure 7.2: The final version of the integrated sweet-pepper harvesting robot of the *Sweeper* project.

References

Bac, C. W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, . doi: 10.1002/rob.21709.

Bruyninckx, H. (2001). Open robot control software: the orocos project. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)* (pp. 2523–2528 vol.3). volume 3. doi: 10.1109/ROBOT.2001.933002.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv:1802.02611*, .

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision ECCV 2014* (pp. 834–849). Springer International Publishing volume 8690 of *Lecture Notes in Computer Science*. doi: 10.1007/978-3-319-10605-2_54.

Eslami, S. M. A. et al. (2018). Neural scene representation and rendering. *Science*, *360*, 1204–1210. doi: 10.1126/science.aar6170. arXiv:http://science.sciencemag.org/content/360/6394/1204.full.pdf.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *CoRR*, *abs/1508.06576*. arXiv:1508.06576.

Girshick, R. B. (2015). Fast R-CNN. *CoRR*, *abs/1504.08083*. arXiv:1504.08083.

Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *CVPR*.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017a). Mask R-CNN. *CoRR*, *abs/1703.06870*.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017b). Mask R-CNN. *CoRR*, *abs/1703.06870*. arXiv:1703.06870.

Higgins, I., Matthey, L., Glorot, X., Pal, A., Uria, B., Blundell, C., Mohamed, S., & Lerchner, A. (2016). Early Visual Concept Learning with Unsupervised Deep Learning. *ArXiv e-prints*, . arXiv:1606.05579.

Hofstadter, D. (2007). *I Am a Strange Loop*. Basic Books.

Lehnert, C., English, A., McCool, C., Tow, A. W., & Perez, T. (2017). Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, *2*, 872–879. doi: 10.1109/LRA.2017.2655622.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, *17*, 1334–1373.

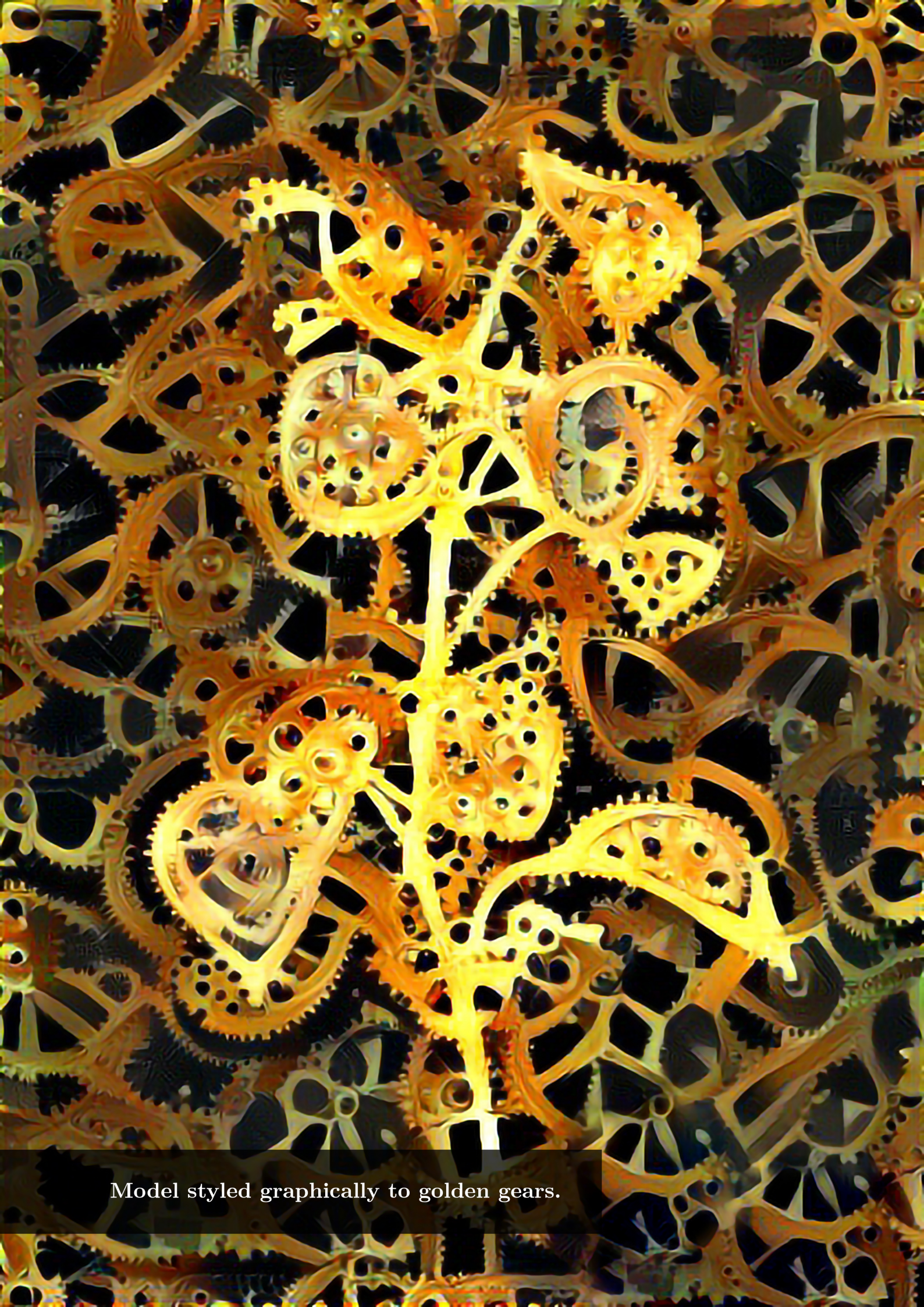
Onwude, D. I., Abdulstter, R., Gomes, C., & Hashim, N. (2016). Mechanisation of large-scale agricultural fields in developing countries – a review. *Journal of the Science of Food and Agriculture*, *96*, 3969–3976. doi: 10.1002/jsfa.7699.

Tenzer, Y., Jentoft, L. P., & Howe, R. D. (2014). The feel of mems barometers: Inexpensive and easily customized tactile array sensors. *IEEE Robotics Automation Magazine*, *21*, 89–95. doi: 10.1109/MRA.2014.2310152.

Wageningen University and Research (2018). Sweet pepper harvesting robot. url: `http://www.sweeper-robot.eu/`.

Wang, T., Liu, M., Zhu, J., Tao, A., Kautz, J., & Catanzaro, B. (2017). High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, *abs/1711.11585*. `arXiv:1711.11585`.

Wolterink, J. M., Leiner, T., Viergever, M. A., & Išgum, I. (2017). Generative adversarial networks for noise reduction in low-dose ct. *IEEE Transactions on Medical Imaging*, *36*, 2536–2545. doi: 10.1109/TMI.2017.2708987.



Model styled graphically to golden gears.

Chapter 8

Summary

Voor een Nederlandse versie, zie Hoofdstuk 9.

8.1 General Summary

Technology in agriculture has brought many positive developments throughout the ages. Although it enabled societies to diversify faster over time and allowed substantial relief in labour for essential needs, unfortunately not all issues around agriculture are yet solved. For the more developed nations, physically and mentally straining work remains required that the local labour force tends to decline. For example the harvesting of sweet-pepper in greenhouses, broccoli in the open field and apples in orchards. As a part of the solution, advanced mechanisation could be researched and introduced through agricultural robotics. Such systems should become able to cope with the crop variation that currently only can be done by humans.

From a more global perspective that includes countries undergoing rapid development, it can be foreseen that demand of food in terms of caloric need will grow with the anticipated population increase. Different solutions like a shift in diet, proper loss management or more equality in the global food distribution might become the most relevant answers, although technology will most likely play its part in all these cases too. The eventual role of agricultural robotics in this issue is hard to currently assess, although it seems plausible it won't be responsible for a significant increase in total caloric production. It is more likely that agricultural robotics could ensure the availability of crops that provide other essential nutrients, e.g. fruits and vegetables. Therefore it might be better seen in the light of ensuring the quality of meals and nourishment of a growing population.

During the 4 years of this research, one cannot fully research and improve all aspects of the agrobotics domain and form it into an enveloping practical solution. Hence, the scope in this work was limited to the automated harvesting of greenhouse crops, with the specific use-case of sweet-pepper as running example. Within this scope, 5 causes were previously identified for agricultural robotics not yet maturing (Bac et al., 2014). The primary cause determined was the inherent crop variation within the task that is limiting performance in harvest success and cycle time. As a solution, it was suggested to further enhance robotic systems with sensing, world modelling and reasoning. For example by pursuing approaches like visual servo control and machine learning. In this work, this suggestion was followed by researching computer vision principles for agricultural robotics with the aim to facilitate a higher level of artificial intelligence.

8.2 Chapter 2

In Chapter 2, first an eye-in-hand sensing and visual control framework was investigated specifically for dense crops such as sweet-pepper. The goal of this work was to provide methods to overcome issues of occlusion and image registration that were previously introduced when sensing was performed externally from the robot manipulator (Bac et al., 2017; Henten et al., 2003). Harvesting robots in cultivars with dense vegetation might require multiple viewpoints and on-line trajectory adjustments in order to reduce the amount of false negatives and correct for fruit movement during a harvest attempt. The hypothesis is that having local sensing information should resolve these issues, e.g. by using cameras within the robot's end-effector itself where the crop is manipulated. Briefly, this Chapter also explored the possibility of adding simultaneous localisation and mapping (SLAM) to obtain a 3D world model using a monocular color camera.

A modular software framework was designed that allows flexible implementation of eye-in-hand sensing and motion control. In contrast to specialised software, the framework proposed aims to support a wide variety of agricultural use cases, hardware and extensions. A set of robotic operating system (ROS) nodes was created to ensure modularity, implementing functionalities for application control, robot motion control, image acquisition, fruit detection, visual servo control and SLAM. System coordination was implemented by an application control node, based on a finite state machine (Hellstrom & Ringdahl, 2013). For the visual servo control and simultaneous localisation and mapping functionalities, libraries ViSP (Marchand, 1999) and LSD-SLAM (Engel et al., 2014) were wrapped in ROS nodes. The framework coherently integrates these parts to provide a higher level functional implementation.

The capabilities of the framework were demonstrated by an example implementation for use with a sweet-pepper crop, combined with hardware consisting of a Baxter robot and a color camera placed on its end-effector. Qualitative tests were performed under laboratory conditions using artificial dense vegetation. Results indicated the framework can be implemented for sensing and robot motion control in sweet-pepper using visual information from the end-effector. Furthermore, we have shown that the framework can be effective for solving sensing and robot motion control in a dense crop from visual information from the end-effector, although this should be further explored in a quantitative study under greenhouse conditions. Regarding the three-dimensional scene reconstruction through SLAM, a world model was obtained using a monocular camera, although calibration for accurate depth measurements remained required.

The framework was a starting point for the SWEEPER project (Wageningen University and Research, 2018). It encourages agricultural systems to not separately sense, plan and act, but perform more intelligent feats such as actively scan the scene, model an internal world state and act accordingly. In turn, this paradigm should overcome issues with poor plant part visibility due to occlusion, image registration but also improve time performance as sensing and acting is combined locally in the same hardware that manipulates the crop. Although this approach is not a novel idea in itself (Ayala et al., 2008; Mehta & Burks, 2014; Baeten et al., 2008), the main contribution was to provide and release a package that integrates all required components that can be adapted for any agricultural application. Through these contributions, Chapter 2 facilitates a higher level of artificial intelligence for agricultural robotics.

8.3 Chapter 3

In Chapter 3 we zoomed in on the sensing component of the framework postulated in the previous chapter. We identified that in order to bring a new level of artificial intelligence to sensing for agricultural robotics, state-of-the-art machine learning methods should be applied. Approaches like convolutional neural networks have dominated the computer vision field the past years (LeCun et al., 2015), though are sparsely applied in the agriculture domain (Hamuda et al., 2016). One of the possible causes is that such networks require large amounts of annotated training data to satisfy their learning needs. Manual annotation can quickly become a bottleneck, as potentially hundreds of images are required. Regarding the use-case of sweet-pepper, annotation time averaged 30 minutes, hence it was deemed infeasible to collect large and diverse sets manually.

To overcome this issue, Chapter 3 put the fundamentals in place for a method to synthetically generate large sets of automatically annotated images, specifically for agricultural scenes. Based on a set of empirically measured plant parameters, a plant generation model was used to generate random instances of 3D meshes. Using render software, realistic scenes were generated that corresponded to the real world greenhouse situation. We hypothesised high similarity between synthetic images and empirical images, which we showed by analysing and comparing both sets qualitatively and quantitatively, although some gap in realism remained. A synthetic dataset of sweet-pepper was created of 10,500 images. Additionally, we obtained and annotated 50 empirical images manually. All images were annotated on a per-pixel level for 7 plant part classes.

The dataset was publicly released with the intention to allow performance comparisons between agricultural computer vision methods, to obtain feedback for modelling improvements and to gain further validations on usability of bootstrapping with synthetic images and fine-tuning with empirical images. A brief perspective was also given on the hypothesis that related synthetic image bootstrapping and empirical image fine-tuning can be used for improved learning, to be further investigated in Chapter 4.

The work in Chapter 3 contributes to solving image segmentation challenges in the plant domain by providing a method to generate annotation training images and thereby reducing the requirement of annotated empirical images. Furthermore, the approach allows to create datasets under a range of conditions, e.g. global an local illumination, season effects or background. This could enable to extent the scope of what the CNNs can learn. CNN methods have been shown to be able to absorb large ranges in variance of the training data (e.g for the iNaturalist dataset of 5000 organism species in 670,000 images (Horn et al., 2017)). In turn, this could improve the robustness and generalisability of computer vision in agriculture, which was previously lacking and only focussed on a few fixed conditions (Hamuda et al., 2016). For these reasons, the methodology supports a higher level of artificial intelligence for agricultural robotics.

8.4 Chapter 4

In Chapter 4 we further investigated how synthetic images can be used to bootstrap CNNs for successful learning of empirical images as compared to other learning strategies. Moreover, the main objective was to find an approach that minimised the requirement of annotated empirical images. We hypothesised that a small manually annotated empirical dataset is sufficient for fine-tuning a CNN bootstrapped with synthetic images. Furthermore we investigated i) multiple deep learning architectures, ii) the correlation between synthetic and empirical dataset size on part segmentation performance, iii) the effect of post-processing using conditional random fields (CRF) and iv) the generalisation performance on other related datasets. For this we have performed 7 experiments using the dataset from Chapter 3.

Results confirmed the hypothesis that only 30 empirical images were required to obtain the highest performance on all 7 classes when a CNN was bootstrapped on related synthetic data (mean intersection-over-union (IOU) of 0.40). Furthermore we found optimal empirical performance when a VGG-16 network was modified to include *à trous* spa-

tial pyramid pooling. Adding CRF only improved performance on the synthetic data. Training binary classifiers did not improve results. We have found a positive correlation between dataset size and performance. Using the synthetic dataset, performance stabilises around 3,000 images. Using the empirical dataset, performance seemed not yet to stabilise after 30 images and indicated that more annotated empirical images would further improve the results. Generalisation to other and related datasets proved possible to a certain extent.

The work in Chapter 4 provides the field of computer vision in agriculture a pioneering machine learning based methodology for state-of-the-art plant part segmentation performance, whilst simultaneously reducing the reliance on labour intensive manual annotations. It has shown that by using synthetic images, recognition performance can be increased. Moreover, classification of uncommon and small plant parts can be boosted (e.g. by about 418% for *peduncles*) and allow for classification of previously unrecognised classes (e.g. *cut peduncles*). It is for these reasons that Chapter 4 contributes of the further introduction of A.I. in agriculture.

8.5 Chapter 5

Although the synthetic images in Chapter 3 were modelled to have an high similarity with the empirical situation, a realism gap still remained. Possibly, by using these synthetic images for bootstrapping, the plant part classification generalisation performance to empirical images was restrained. To investigate this and to improve performance further on empirical images, Chapter 5 explored applying a cycle consistent generative adversarial network (cGAN) to the dataset with the objective to generate more realistic synthetic images by translating them to the feature distribution of the empirical domain.

We first hypothesised that plant part image features such as color and texture become more similar to the empirical domain post translation. We performed 7 image segmentation experiments using convolutional neural networks with different combinations of synthetic, synthetic translated to empirical and empirical images. We hypothesised that i) the translated images can be used for improved empirical learning and ii) that without any empirical fine-tuning, improved empirical learning can be achieved with translated images compared to only using synthetic images.

To find evidence for the first hypothesis, in Part I of Chapter 5, a cGAN was applied to the dataset. The analysis showed that the image feature distributions of these translated

images, both in color, texture and other image features, were improved towards the empirical images. Qualitatively, the translated synthetic images looked highly similar to the real world situation, barring some introduced translation artifacts. Although cGan could not improve upon geometrical dissimilarities, this proved an advantage as the synthetic ground truth matched post-translation images. In turn, this allowed for the experiments on improved learning using translated images in the second part of this chapter.

In Part II, it was evaluated to what extent synthetic translated images to the empirical domain could improve on CNN learning with empirical images over other learning strategies. We confirmed the hypotheses that by using translated images and fine-tuning with empirical images, the highest performance for empirical images can be achieved (IOU=0.52) over training with only empirical (IOU=0.41) or synthetic data (IOU=0.48).

Besides improving performance on empirical images, another key contribution of the work is the further minimisation of the CNN dependency on annotated empirical data. We confirmed the hypothesis that without any empirical image fine-tuning, learning can be improved with translated images (IOU=0.31), a 55% increase over using just synthetic images (IOU=0.20).

The work presented in Chapter 5 can be seen as an important step towards improved sensing for agricultural robotics. It support raising the level of artificial intelligence in agricultural robotics by facilitating CNN semantic part segmentation learning without, or reducing, the requirement of annotated images.

8.6 Chapter 6

In Chapter 6 it was aimed to bring all previous chapters into practice. The objective was to estimate angles between fruit and stems to support visual servo control grasping in a sweet-pepper harvesting robot. We hypothesised that from color images, these angles in the horizontal plane can be derived under unmodified greenhouse conditions, whilst meeting the end-effector positioning requirements. For this we additionally hypothesised that the location of a fruit and stem could be inferred in the image from sparse semantic segmentations.

The scope of Chapter 6 was focussed on 4 sub-tasks of the robot's harvest sequence. Each task was evaluated in 3 conditions: simplified laboratory, simplified greenhouse and unmodified greenhouse. The requirements for each task were propagated back from the

end-effector design that needed a 25° positioning accuracy to perform correctly. In Task I, color image segmentation for classes background, fruit and stem plus wire was performed using the method from Chapter 4, meeting the IOU requirements. In Task II, the stem pose was estimated from the segmentations. In Task III, centers of the fruit and stem were estimated from the output of previous tasks. Both tasks met the requirement of 25 pixel accuracy on average, confirming the additional hypothesis. In Task IV, the centers were used to estimate the angle between the fruit and stem, meeting the accuracy requirement of 25°, confirming the main hypothesis for 73% of the cases.

The impact of the work can be found in the support of successful grasping for robotic harvesting in the greenhouse from monocular eye-in-hand sensing, using state-of-the-art image segmentation methods. It enables robotic systems to use image sensors, analyse its contents, form a basic world model and act based on this knowledge. It facilitates a higher level of artificial intelligence for agricultural robotics by showing how to apply visual servo control and plant scanning from Chapter 2 with deep learning for plant part image segmentation from Chapters 3 through 5.

References

Ayala, M., Soria, C., & Carelli, R. (2008). Visual servo control of a mobile robot in agriculture environments. *Mechanics Based Design of Structures and Machines*, *36*, 392–410. doi: 10.1080/15397730802409301. arXiv:https://doi.org/10.1080/15397730802409301.

Bac, C. W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, . doi: 10.1002/rob.21709.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Baeten, J., Donné, K., Boedrij, S., Beckers, W., & Claesen, E. (2008). Autonomous fruit picking machine: A robotic apple harvester. In C. Laugier, & R. Siegwart (Eds.), *Field and Service Robotics: Results of the 6th International Conference* (pp. 531–539). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-75404-6_51.

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision ECCV 2014* (pp. 834–849). Springer International Publishing volume 8690 of *Lecture Notes in Computer Science*. doi: 10.1007/978-3-319-10605-2_54.

Hamuda, E., Glavin, M., & Jones, E. (2016). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, *125*, 184 – 199. doi: https://doi.org/10.1016/j.compag.2016.04.024.

Hellstrom, T., & Ringdahl, O. (2013). A software framework for agricultural and forestry robots. *Industrial Robot: An International Journal*, *40*, 20–26. doi: 10.1108/01439911311294228.

Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., Bontsema, J., & Os, E. V. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering*, *86*, 305 – 313. doi: http://dx.doi.org/10.1016/j.biosystemseng.2003.08.002.

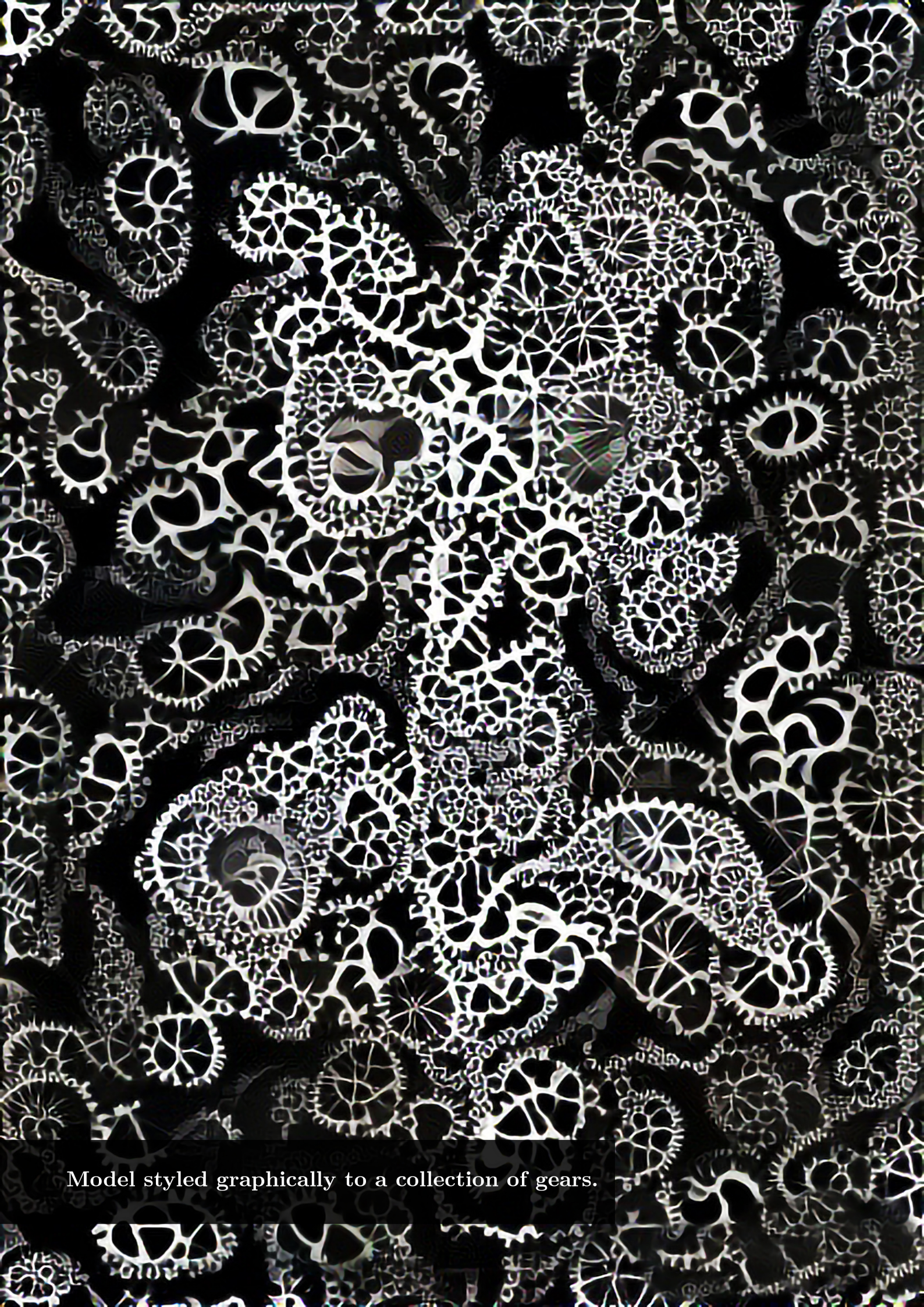
Horn, G. V., Mac Aodha, O., Song, Y., Shepard, A., Adam, H., Perona, P., & Belongie, S. J. (2017). The inaturalist challenge 2017 dataset. *CoRR*, *abs/1707.06642*. arXiv:1707.06642.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning, . *521*, 436–44.

Marchand, E. (1999). Visp: a software environment for eye-in-hand visual servoing. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* (pp. 3224–3229 vol.4). volume 4. doi: 10.1109/ROBOT.1999.774089.

Mehta, S., & Burks, T. (2014). Vision-based control of robotic manipulator for citrus harvesting. *Computers and Electronics in Agriculture*, *102*, 146 – 158. doi: https://doi.org/10.1016/j.compag.2014.01.003.

Wageningen University and Research (2018). Sweet pepper harvesting robot. url: `http://www.sweeper-robot.eu/`.



Model styled graphically to a collection of gears.

Chapter 9

Samenvatting

For the original English version, please be referred to chapter 8.

To alliviate the repitetive labour of writing and then translating, this chapter was machine translated to Dutch by an (semi-) artificial intelligence, after which small manual corrections have been made.

Om het repetitieve werk van het schrijven en het vertalen te verlichten, werd dit hoofdstuk machinaal vertaald naar het Nederlands door een (semi-) kunstmatige intelligentie, met een kleine handmatige nabewerking.

9.1 Algemene Samenvatting

Technologie in de landbouw heeft door de eeuwen heen veel positieve ontwikkelingen gebracht. Hoewel het de samenlevingen in staat stelde om zich in de loop van de tijd sneller te diversifiëren en daarbij een aanzienlijke verlichting van de arbeid mogelijk maakte, zijn helaas nog niet alle problemen rond de landbouw opgelost. Voor de meer ontwikkelde landen blijft fysiek en mentaal belastend werk vereist waarbij de lokale beroepsbevolking steeds meer de neiging heeft om dat te weigeren. Dit is bijvoorbeeld te zien bij het oogsten van paprika in kassen, broccoli in het open veld en appels in boomgaarden. Als mogelijke oplossing kan geavanceerde mechanisatie worden onderzocht en geïntroduceerd via agrobotica. Dergelijke systemen zouden moeten om kunnen gaan met de variatie in het gewas, iets wat nu alleen nog door de mens gedaan kan worden.

Vanuit een meer mondiaal perspectief kan worden verwacht dat de vraag naar voedsel in termen van calorische waarde zal toenemen met de verwachte toename van de bevolking. Verschillende oplossingen zoals een verschuiving in het dieet, een goede beheersing van voedselverliezen of meer gelijkheid in de wereldwijde voedselverspreiding kunnen de meest relevante oplossingen gaan vormen. De uiteindelijke rol van agrobotica in deze kwestie is moeilijk te beoordelen, hoewel het aannemelijk lijkt dat deze niet verantwoordelijk zal zijn voor een significante toename van de totale calorieproductie. Het is waarschijnlijker dat agrobotica de beschikbaarheid van gewassen kan garanderen die andere essentiële voedingsstoffen verschaffen, b.v. fruit en groenten. Daarom kan het nut van agrobotica misschien beter worden gezien in het licht van het waarborgen van de kwaliteit van maaltijden en voedingswaarde voor een groeiende bevolking.

Gedurende de 4 jaar van dit onderzoek kan men niet alle aspecten van het agrobotica domein volledig onderzoeken, verbeteren of een allesomvattende oplossing bieden. Vandaar dat de reikwijdte van dit werk beperkt was tot het geautomatiseerd oogsten van groenten in de glastuinbouw, met als specifiek lopend voorbeeld het paprika gewas. In dit kader zijn eerder 5 oorzaken geïdentificeerd waarom agrobotica nog niet volwassen is geworden (Bac et al., 2014). De primaire oorzaak was dat de variatie in het gewast de prestaties zoals het oogstsucces en de cyclustijd beperkt. Als oplossing werd eerder gesuggereerd om robotsystemen verder te verbeteren op het gebied van zijn waarneming. Bijvoorbeeld door methoden na te streven zoals visuele terugkoppeling tijdens de uitvoer van de bewegingen, maar ook patroonherkenning voor het begrijpen van de beeldinformatie. In dit werk werd deze suggestie gevolgd door onderzoek te doen naar principes voor beeldherkenning.

9.2 Hoofdstuk 2

In hoofdstuk 2 werd eerst een raamwerk onderzocht voor het zogenoemde oog-in-hand principe waarbij de camera in de grijper zit. Dit principe werd gecombineerd met de aansturing van de robot via visuele terugkoppeling. Het doel van dit werk was om methoden te bieden om problemen met occlusie en beeldregistratie te overwinnen die worden geïntroduceerd wanneer de detectie extern van de robot manipulator wordt uitgevoerd (Bac et al., 2017; Henten et al., 2003). Het oogsten door robots in gewassen met dichte begroeiing vereist meerdere kijkhoeken en aanpassingen tijdens de beweging naar de vrucht. De hypothese is dat extra lokale informatie deze problemen zou moeten helpen oplossen, b.v. door camera's te gebruiken in de grijper van de robot zelf. Ook onderzocht dit hoofdstuk kort de mogelijkheid om 'gelijktijdige lokalisatie en mapping' (SLAM) toe te voegen om zo een 3D-wereldmodel te maken op basis van een enkele kleurencamera.

Een modulair softwarekader werd ontworpen dat flexibele implementatie van oog-in-hand beeldherkenning en visuele terugkoppeling tijdens de uitvoer van de bewegingen (visuele servobesturing) mogelijk maakt. In tegenstelling tot gespecialiseerde software, is het voorgestelde kader bedoeld om een breed scala aan taken in de landbouw, hardware en uitbreidingen te ondersteunen. Het kader bestaat uit een set Robotic Operating System modules (ROS), waardoor modulariteit kan worden gegarandeerd voor functionaliteiten zoals de robot aansturing, beeldacquisitie, beeldherkenning, visuele servobesturing en SLAM. De algemene systeem coördinatie werd geïmplementeerd door een module die was gebaseerd op een zogenaamde 'eindige toestandsmachine', waarin de toestand van de robot wordt opgedeeld in staten en er transities tussen staten kunnen worden geprogrammeerd (Hellstrom & Ringdahl, 2013). Voor de visuele servobesturing en SLAM functionaliteiten waren de bibliotheken ViSP (Marchand, 1999) en LSD-SLAM Engel et al. (2014) verpakt in ROS modules. Het raamwerk integreert deze delen op coherente wijze om een functionele implementatie op een hoger niveau mogelijk te maken.

De mogelijkheden van het raamwerk werden gedemonstreerd met een voorbeeld implementatie voor de oogst van paprika. Hierbij werd een kleurencamera op de grijper van de Baxter-robot geplaatst. Kwalitatieve tests werden uitgevoerd onder laboratoriumomstandigheden met behulp van een kunstmatig gewas. De resultaten wijzen erop dat het raamwerk kan worden geïmplementeerd voor de detectie van, en robotbesturing naar de paprika's met behulp van visuele informatie vanuit de grijper. Dit werk zal echter verder onderzocht moeten worden in een kwantitatief onderzoek onder kasomstandigheden. Met

betrekking tot de driedimensionale scène-reconstructie via SLAM, werd een wereldmodel verkregen, hoewel kalibratie voor nauwkeurige dieptemetingen noodzakelijk bleef.

Het raamwerk was een startpunt voor het SWEEPER-project (Wageningen University and Research, 2018). Het ondersteund agrobotica systemen om niet slechts afzonderlijk te kijken, te plannen en te handelen, maar om actief de omgeving te bekijken en een interne wereldstaat te modelleren om daar naar te handelen. Dit paradigma moet het probleem van de slechte zichtbaarheid van plantendelen doorvan occlusie overwinnen, maar ook de tijdprestaties verbeteren doordat het waarnemen en handelen nu lokaal in de grijper kan worden gecombineerd. Hoewel deze aanpak op zich geen nieuw idee is (Ayala et al., 2008; Mehta & Burks, 2014; Baeten et al., 2008), was de belangrijkste bijdrage van dit werk het leveren en vrijgeven van een pakket waarin alle vereiste componenten zijn geïntegreerd die voor elke landbouwtoepassing zou kunnen worden aangepast. Door deze bijdragen vergemakkelijkt hoofdstuk 2 een hoger niveau van kunstmatige intelligentie voor agrobotica.

9.3 Hoofdstuk 3

In hoofdstuk 3 hebben we ingezoomd op de module voor beeldherkenning van het raamwerk uit het vorige hoofdstuk. Er is vastgesteld dat om een nieuw niveau van kunstmatige intelligentie te behalen in de agrobotica, moderne machinale leermethoden moeten worden toegepast. Methoden zoals convolutionele neurale netwerken (CNN) domineerden het gebied van de beeldherkenning de afgelopen jaren (LeCun et al., 2015), hoewel deze maar schaars worden toegepast in het landbouwdomein (Hamuda et al., 2016). Een van de mogelijke oorzaken is dat dergelijke netwerken grote hoeveelheden geannoteerde beelden nodig hebben om aan hun leerbehoeften te voldoen. Handmatige annotatie kan snel een knelpunt worden, omdat mogelijk honderden afbeeldingen nodig zijn. Met betrekking tot het voorbeeld van paprika, bedroeg de tijd voor annoteren gemiddeld 30 minuten per beeld, waardoor het onhaalbaar werd geacht om grote en diverse sets met de hand te verzamelen.

Om dit probleem op te lossen, heeft hoofdstuk 3 de basis gelegd voor een methode om grote synthetische sets van automatisch geannoteerde afbeeldingen te genereren, specifiek voor scènes in de landbouw. Op basis van een reeks empirisch gemeten plant parameters werd een plantenmodel gemaakt om willekeurige instanties te kunnen genereren. Met behulp van render-software werden realistische scènes gegenereerd die overeenkwamen

met de echte situatie in de kas. Een een hoge gelijkenis tussen synthetische beelden en empirische beelden was het uitgangspunt, wat aan werd getoond door beide sets zowel kwalitatief als kwantitatief te vergelijken. Er bleef echter wel een kleine kloof in realisme. Een synthetische beeldenset van paprika werd gemaakt bestaande uit 10500 afbeeldingen. Daarnaast zijn handmatig 50 empirische afbeeldingen geannoteerd. Alle afbeeldingen zijn onderverdeeld in 7 klassen, waarbij iedere pixel is geannoteerd.

De dataset werd publiekelijk vrijgegeven met de bedoeling om prestatievergelijkingen mogelijk te maken tussen verschillende beeldherkenningsmethoden, om zo feedback te krijgen voor verbetering van de gebruikte modellen. In het hoofdstuk werd ook een kort perspectief gegeven op de hypothese dat synthetische beelden gebruikt kunnen worden om een voor-training te doen, waardoor er minder handmatig geannoteerde beelden nodig zijn. Deze hypothese werd verder onderzocht in hoofdstuk 4.

Het werk in hoofdstuk 3 draagt bij aan het oplossen van problemen met de beeldsegmentatie door een methode te bieden om triainingsbeelden te genereren en daardoor de noodzaak van geannoteerde empirische beelden te verminderen. Verder maakt de methode het mogelijk om datasets te creëren onder een reeks van omstandigheden, b.v. globale lokale verlichting, seizoenseffecten of achtergrond. Dit zou het mogelijk kunnen maken om de reikwijdte van wat de neurale netwerken kunnen leren te vergroten.

9.4 Hoofdstuk 4

In hoofdstuk 4 is er verder onderzocht hoe synthetische beelden kunnen worden gebruikt om CNN's in te leren en toe te passen op empirische beelden, in vergelijking met andere leerstrategieën. Bovendien was het doel om een benadering te vinden die de vereiste van geannoteerde empirische beelden minimaliseerde. Er werd verondersteld dat een kleine handmatig geannoteerde empirische dataset voldoende zou zijn voor het fine-tunen van een CNN, welke was ingeleerd met synthetische afbeeldingen. Verder waren i) verschillende deep learning architecturen onderzocht, maar ook ii) de correlatie tussen de grootte van synthetische en empirische datasets op de prestatie van de beeldherkenning, iii) het effect van nabewerking op basis van 'conditionele random fields' (CRF) en iv) de mogelijkheden van generalisatie naar andere, gerelateerde beelden. Hiervoor zijn 7 experimenten uitgevoerd met behulp van de dataset van hoofdstuk 3.

De resultaten bevestigden de hypothese dat er slechts 30 empirische afbeeldingen nodig waren om de hoogste prestaties te behalen voor alle 7 klassen wanneer een CNN werd

ingeleerd met synthetische gegevens. Verder werden optimale empirische prestaties gevonden wanneer een VGG-16-netwerk werd gewijzigd met de toevoeging van '*à trous spatial pyramid pooling*'. Het toevoegen van CRF verbeterde alleen de prestaties op de synthetische gegevens. Het trainen van binaire classifiers heeft de resultaten niet verbeterd. Er was een positieve correlatie gevonden tussen de het aantal beelden beschikbaar voor het trainen en de prestaties van de beeldherkenning. Met behulp van de synthetische dataset stabiliseert de prestatie ongeveer 3.000 afbeeldingen. Met behulp van de empirische dataset leek de prestatie na 30 beelden nog niet te stabiliseren en wat aangaf dat meer geannoteerde empirische beelden de resultaten verder zouden verbeteren. Een verdere conclusie was dat de CNNs ook tot op een zekere hoogte op andere, gerelateerde datasets van toepassing waren.

Het werk in hoofdstuk 4 biedt het vakgebied van de beeldherkenning in de landbouw een leermethode voor het herkenning van plantenonderdelen, waarbij tegelijkertijd de afhankelijkheid van arbeidsintensieve handmatige annotaties wordt verminderd. Het heeft aangetoond dat door het gebruik van synthetische afbeeldingen de beeldherkenning kan worden verbeterd. Bovendien kan de classificatie van ongebruikelijke en kleine plantendelen worden gestimuleerd en de classificatie van eerder niet-herkende klassen mogelijk maken.

9.5 Hoofdstuk 5

Hoewel de synthetische afbeeldingen in hoofdstuk 3 werden gemodelleerd om een grote gelijkenis met de empirische situatie te hebben, bleef er nog steeds een redelijke kloof in realisme bestaan. Mogelijk kunnen de prestaties voor beeldherkenning verder worden verbeterd wanneer deze beelden nog realistischer worden gemaakt. Om dit te onderzoeken, werd in hoofdstuk 5 een zogenaamd *'cycle consistent generative adversarial network'* (cGan) toegepast met als doel de synthetische beelden te transformeren naar de eigenschappen van het empirische domein.

Een van de hypothesen was dat per synthetisch plantdeel, de eigenschappen zoals kleur en textuur meer zouden gaan lijken op de echte situatie. In totaal werden 7 beeldsegmentatie-experimenten uitgevoerd met met convolutionele neurale netwerken met verschillende combinaties van i) synthetische, ii) synthetische naar empirisch getransleerde en iii) empirische beelden. Er werd verondersteld dat i) de getransleerde afbeeldingen kunnen worden gebruikt voor verbeterd empirisch leren en ii) dat zonder empirische beelden ti-

jdens het trainen, een beter resultaat kan worden gehaald met de getransleerde beelden dan slechts met synthetische beelden.

Om bewijs te vinden voor de eerste hypothese, is in deel I van hoofdstuk 5 een cGAN toegepast op de dataset uit hoofdstuk 3. De analyse liet zien dat de verdeling in eigenschappen van deze getransleerde afbeeldingen, zowel in kleur en textuur als andere beeldkenmerken verbeterd waren en meer gingen lijken op de echte situatie. Kwalitatief leken de getransleerde synthetische beelden sterk op de werkelijkheid, met uitzondering van enkele geïntroduceerde artefacten. Hoewel cGan de geometrische ongelijkheden niet kon verbeteren, bleek dit een voordeel omdat de synthetische annotaties overeenkwam met getransleerde afbeeldingen. Dit maakte op zijn beurt de experimenten mogelijk voor verbeterd leren met behulp van getransleerde afbeeldingen in het tweede deel van dit hoofdstuk.

In deel II werd geëvalueerd in welke mate synthetische getransleerde afbeeldingen naar het empirische domein het CNN-leren zouden kunnen verbeteren. De hypotheses werd bevestigd dat door het leren met van vertaalde afbeeldingen samen met empirische beelden, de hoogste prestaties voor empirische beelden kunnen worden bereikt dan alleen empirische of synthetische beelden.

Naast het verbeteren van de prestaties op empirische beelden, is een andere belangrijke bijdrage van het werk de verdere minimalisatie van de afhankelijkheid van geannoteerde empirische beelden. De hypothese werd bevestigd dat zonder enig leren met empirische beelden, de prestaties kon worden verbeterd met getransleerde afbeeldingen, met een toename van 55% ten opzichte van alleen synthetische afbeeldingen.

9.6 Hoofdstuk 6

In hoofdstuk 6 was het de bedoeling om alle voorgaande hoofdstukken in de praktijk te brengen. Het doel was om hoeken tussen fruit en stengels in te schatten om visuele servobesturing te ondersteunen voor een paprika oogstrobot. Er werd verondersteld dat uit slechts kleurenbeelden deze hoeken in het horizontale vlak, onder niet-gemodificeerde praktijkomstandigheden, goed zou kunnen worden afgeleid. Een deel-hypothese was dat de locatie van een vrucht en een stengel kon worden afgeleid uit de afbeelding van semantische segmentaties uit hoofdstuk 4.

Hoofdstuk 6 omvatte 4 subtaken van de oogstsequentie van de robot. Elke taak werd geëvalueerd in 3 omstandigheden: vereenvoudigd kunstgewas in het laboratorium, vereen-

voudigde gewas in de kas en een niet-gemodificeerd gewas in de kas. De minimale prestatie vereisten voor elke taak werden afgeleid vanaf het ontwerp van de eindmanipulator dat een nauwkeurigheid in positionering van 25° nodig had om correct te presteren. In taak I werd de beeldherkenning van vrucht en stam uitgevoerd met behulp van onze methode van hoofdstuk 4. In taak II werd de positie van de stengel geschat op basis van de beeldherkenning. In taak III werden de centra van het fruit en de stengel geschat. In taak IV werden de centra gebruikt om de hoek tussen het fruit en de stengel te schatten. De resultaten bleken aan de nauwkeurigheidseis van 25° te voldoen voor 73% van de vruchten.

De impact van het werk kan worden gevonden in de ondersteuning van succesvol grijpen van vruchten in de kas, op basis van kleurenbeelden vanuit de grijper , met behulp van de modernste beeldsegmentatie methoden. Het stelt robotsystemen in staat om beeldsensoren te gebruiken, de inhoud ervan te analyseren, een basaal wereldmodel te vormen en op basis van deze kennis te handelen.

Referenties

Ayala, M., Soria, C., & Carelli, R. (2008). Visual servo control of a mobile robot in agriculture environments. *Mechanics Based Design of Structures and Machines*, *36*, 392–410. doi: 10.1080/15397730802409301. arXiv:https://doi.org/10.1080/15397730802409301.

Bac, C. W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., & van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, . doi: 10.1002/rob.21709.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*, 888–911. doi: 10.1002/rob.21525.

Baeten, J., Donné, K., Boedrij, S., Beckers, W., & Claesen, E. (2008). Autonomous fruit picking machine: A robotic apple harvester. In C. Laugier, & R. Siegwart (Eds.), *Field and Service Robotics: Results of the 6th International Conference* (pp. 531–539). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-75404-6_51.

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision ECCV 2014* (pp. 834–849). Springer International Publishing volume 8690 of *Lecture Notes in Computer Science*. doi: 10.1007/978-3-319-10605-2_54.

Hamuda, E., Glavin, M., & Jones, E. (2016). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, *125*, 184 – 199. doi: https://doi.org/10.1016/j.compag.2016.04.024.

Hellstrom, T., & Ringdahl, O. (2013). A software framework for agricultural and forestry robots. *Industrial Robot: An International Journal*, *40*, 20–26. doi: 10.1108/01439911311294228.

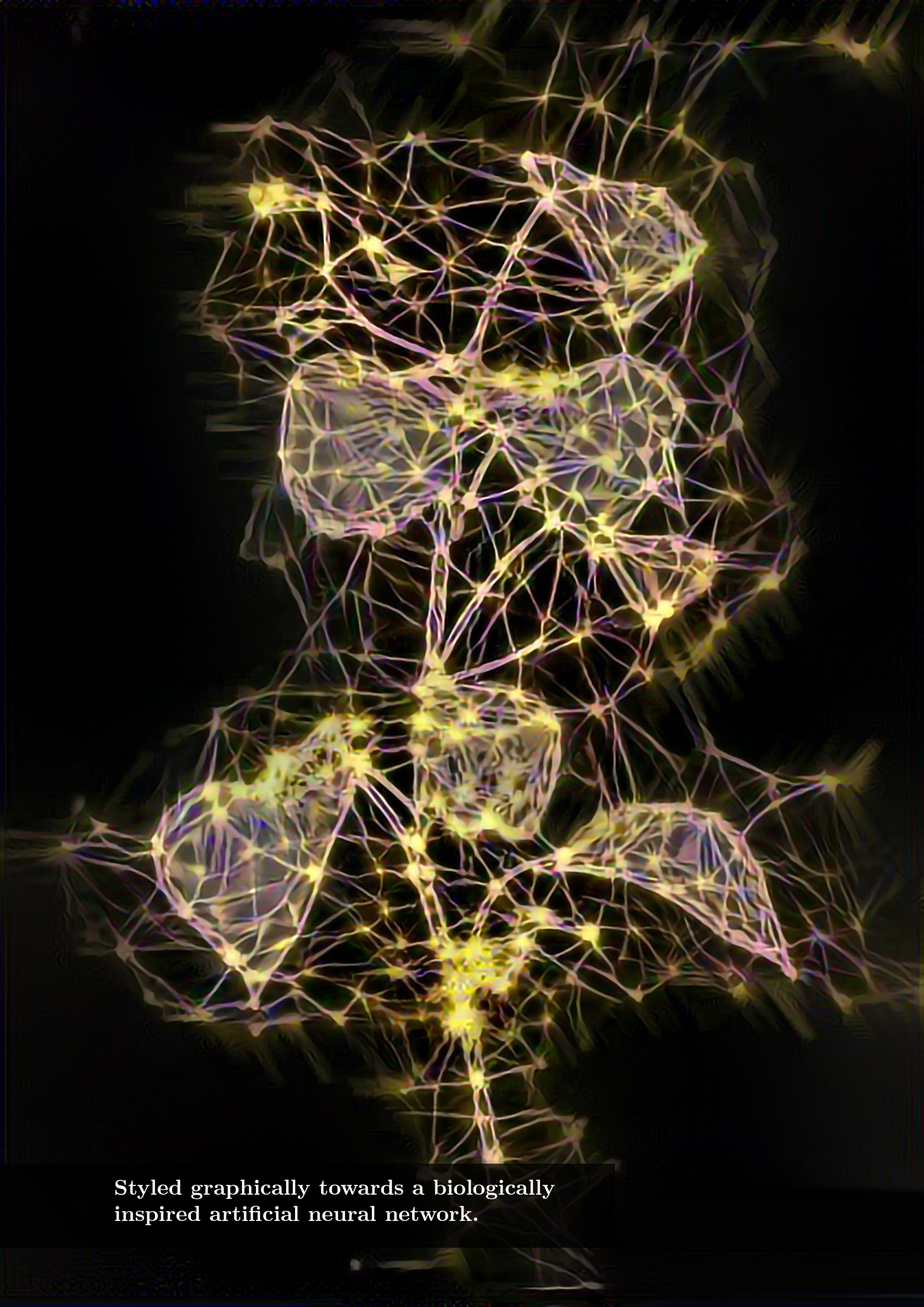
Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., Bontsema, J., & Os, E. V. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering*, *86*, 305 – 313. doi: http://dx.doi.org/10.1016/j.biosystemseng.2003.08.002.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning, . *521*, 436–44.

Marchand, E. (1999). Visp: a software environment for eye-in-hand visual servoing. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* (pp. 3224–3229 vol.4). volume 4. doi: 10.1109/ROBOT.1999.774089.

Mehta, S., & Burks, T. (2014). Vision-based control of robotic manipulator for citrus harvesting. *Computers and Electronics in Agriculture*, *102*, 146 – 158. doi: https://doi.org/10.1016/j.compag.2014.01.003.

Wageningen University and Research (2018). Sweet pepper harvesting robot. url: `http://www.sweeper-robot.eu/`.



Styled graphically towards a biologically inspired artificial neural network.

Chapter 10

Acknowledgements

Chronology

When I first started at Wageningen University and Research as junior *aspiring* scientist, I told my team leader Silke Hemming that it would take me about 2 years of orientation to decide upon pursuing a Ph.D. degree. Before getting in such a fierce endeavour, I wanted to have clear which topics I admired and how these would integrate the new group I just joined. At least I was somewhat sure that it was my intention to apply my previous academic education in artificial intelligence to the domain of computer vision in agriculture. After half a year, Silke asked politely if I would like to expand my domain to robotics. *"Of course!"*, I said and wondered why that was even a question. For me those two topics are in symbiosis and moreover, fit my background and interests more than nicely. I got assigned to the project *Crops*, supporting the team with all kinds of work for a sweet-pepper harvesting robot. I learned to program with the Robotic Operating System and got familiar with the hardware involved. After a year or so, the management team decided to grant me a permanent position to continue playing with robots and cameras in return for a more than decent salary. I think it is fair to acknowledge the effect on personal tranquility and security such a gesture achieves. Now that there was a stable foundation to build upon, I decided on the black beaches of Santa Cruz de la Palma that it was time to seek the next challenge of a doctoral degree. I said to Lianne, my then fiancée and who just started her own Ph.D. journey, that I very much liked her former neighbour professor Eldert van Henten as my promotor.

He was the kind of person I was looking for because, in my humble view, his perspective on matters is extremely sharp, rational and honest whilst keeping focus on the higher goals. I envied such a mindset and I still do. I have seen his approach on the sidelines of *Crops*, where he promoted other Ph.D. graduates and guided master students. Not always there seemed agreement between the professor and his pupils. Eldert could ask the right questions to make things shake with the intent to see if the scientific structure would still stand after some cutting arguments. His time was valuable and he made it clear one would always need to have thought things through before delivering any construct. Sometimes, Eldert would even take stance with an opposing view, playing the devil's advocate. Such an artificial intelligent disagreement helped in getting the right arguments on track and get deeper on the matter whilst burning away some outer layers, revealing the real core underneath. I too have tried to find disagreement with Eldert in order to find better answers. I waited for three years, hoping to get into a small argument to boil things through. It did not happen. I might have failed in that regard. Even when I tried to argue on the topic of finances (a very fine Dutch culturally matter to widely disagree upon) we could not get things truly lit. Eventually, somehow the opposite happened and I ended up cashing a blank cheque. Eldert, my apologies if I sometimes brought too much business-to-business culture to table.

After my decision, there on the beach, it took about 6 months to work out the Ph.D. track proposal details. Concurrently, I had pushed and worked on the writing of the follow-up grant *Sweeper* where the results from *Crops* would be alleviated towards the market. It was my hope this project with its humongous size of € 4,457,035 got funded to give time and room for my thesis. I knew that without a stable feed of money, I would get a hard time getting a coherent set of research papers within any reasonable time frame. But also a focus would be hard to achieve when doing multiple smaller projects instead of a few big ones. However, as such things generally go with the European Union, bureaucracy lets you wait for three quarters of a year.

In the meantime, Joris IJsselmuiden and Jochen Hemming joined my Ph.D. project as co-promotors. Joris just started as a post-doc and with a background in artificial intelligence as well, we instantly matched. As a person I came to like Joris very much, his approach was rigorous and our many experimental and paper reviews were very thorough. Jochen initially joined to keep an extra eye on me (in a good sense) and helped me with my then initial dabbling writings. Moreover, he made sure no other project got derailed for the sake of my promotion alone. As one of the core colleagues I worked with in previous years, I came to know him as someone who always puts in the extra effort until things

are right the way they belong. Perhaps this is a just a typical German trait to envy, but in any case a very good one to have in the team.

I must not forget to mention another important moment, as a year before starting my Ph.D project. I promptly asked Silke if I could go to Sicily for a week to attend a summer school for computer vision. She answered something along the lines of *"Well, I cannot hold you back anyway"* and there I went flying to Catania, only to find out on my first day that taxis drive you there around in circles, busses never actually show up, nobody speaks any English and the hotel I booked did not exist. Fortunately, 12 coherent digits on a black piece of plastic and some vivid pointing gets you quite comfortable in the world these days. The morning after this horrible start, I went to the bus stop to join the summer school when I bumped into *the* Douglas Perrin. Amongst all other random people, he seemed the most interesting person as he talked about bitcoins and this new hot thing called convolutional neural networks. I asked if he wanted to be bus mates and I found myself lucky to already have made a single new friend within five minutes. The organisers then followed to inform us that the 20 minute bus ride got extended to four hours because, not surprisingly anymore, the convention center went bankrupt the week before. Finally arrived, I told Doug that he should not feel obliged to keep hanging out with me and to also find new other friends. He laughed at me, but my reverse psychology for once worked like a charm; the next six years we have met annually since.

My Ph.D. research got a nice bootstrap with the preparation of my first visit to the biorobotics laboratory of professor Rob Howe at Harvard. It appeared that Douglas was affiliated with this lab and he was so kind to arrange an appointment there. For 3 months I had the privilege to fully focus on the first cornerstone of my thesis, leading to the work described in Chapter 2. About half-way of my visit, word came through that the *Sweeper* grant was approved. This was a major psychological boost, as it meant that upon my return work on my thesis could continue in the direction I had written it down.

Not all time passed fluently however. The middle part of my Ph.D. track was a bit more burdensome than expected. About half to three-quarters of my time I was busy with different projects like *Pick&Pack*. These projects were less focussed on science and did not directly contribute to the work written here. Nonetheless, it did allow to learn a more broad set of useful skills and I still think it was a valuable period.

Two years after my first visit to Cambridge, I again had the pleasure to be able to join Rob's lab and interact with Douglas and their fine graduate students. Over the course another three months, I deployed the rendering of my synthetic models on Harvard's

supercomputer Odyssey, organised the results and used them for training on these hot convolutional neural networks. Although the first results looked like spilled confetti, after a couple of weeks of tuning some real magic occurred right there. For the first time, I was able to recognise plant parts on a per-pixel level in empirical images. A milestone that had taken a full year of intermediate steps to reach; now that was some sincere delayed gratification. To spice up this already delightful visit a bit more, by mere coincidence the most wealthy person on the planet decided to pay us a lab visit and I got a handshake and my picture taken with Jeff Bezos, founder of Amazon.

After my return to Wageningen, I ran additional experiments and I could finalise Chapters 3 and 4. I submitted the former to a journal, but the dataset got promptly rejected. One enlightened reviewer claimed that instead I should use state-of-the-art generative adversarial networks to obtain my synthetic data. In my defence, I argued that I might have if they actually existed at the time of my modelling and moreover, one would better need synthetic data as a dependency first. My rebuttal did not appear to matter, so in discussion with the editor I resubmitted to try the reviewer lottery again. In the meantime, I still was a bit angry with the reviewer's comments, but I applied cycle generative adversarial networks anyway. Part of this work got accepted at the international Conference on Intelligent Robots and Systems (IROS) and was later extended to become Chapter 5. Thank you, reviewer #2 !

At IROS in Vancouver, I met Jim Ostrowski, CTO of Blue River Technologies, a Stanford spin-off at the time with a nice $245K investment. He seemed quite fond of my work and this was mutual, as over 4 years ago I had written to Blue River to cooperate. Now, after just a few years, they were acquired by John Deere for $300M. Most admirable, although I still have trouble seeing what they can do differently than my colleague Jochen and his successful mechanical weeding projects. Regardless, during his presentation he mentioned he had a Ph.D. from the California Institute of Technology, which happened to be the next visit of (my now wife) Lianne for her neuroscience collaboration with John Allman. I figured it would be most convenient if I could join her travels and to take a month of focus to finish writing my thesis. Jim got in contact with professor Pietro Perona and he was so kind to offer me a desk. Additionally, John was so generous to arrange housing for us only one block away from campus. It has been an honour to stay within the history of the house of Max Delbrück, a CalTech Nobel laureate.

Gratitudes

Although uncustomary, I would like to start chronologically by thanking Silke Hemming, for she was the first to facilitate the idea of doctoral graduation and was always open for ways to keep developing myself. She trusted me that I could bring this project to a good, timely end, concurrently with my other responsibilities as a junior researcher. Silke, without your open support this would all have ground to an early halt and I am happy we could make this work together. To date, as I said before, you are one of the finest managers I have met.

Eldert van Henten, as soon as Silke and I agreed upon this adventure, you were there for us. Thank you very much sincerely for promoting me, as I could not have wished for a better match. Your background in agricultural robotics, most notably the cucumber harvesting robot, your experience with *Crops* and many other projects have had a great positive impact on the direction of my research. I am especially very grateful for all your revisions on my papers, as I was always amazed (and sometimes shocked) on the level of further refinement that could still be achieved. Together we have written some fine paragraphs and made sure every word, and every sentence was correct. I am also happy you gave me the independence I needed and the trust that my choices would work out. I hope we can continue working together, in whatever interesting construction comes next.

Joris IJsselmuiden, also you have my sincere gratitude. As my day-to-day supervisor you have been an important person to reflect upon ideas, results and writing. You were also very kind to let me work very independently and for that freedom I am grateful. When I needed your vision, I could rely on you and exchange thoughts. Above all other things, you valued that science should be fun and that we should never stress about results or timing things (although the latter sometimes conflicted with my rigorous, perhaps neurotic, desire of planning). Your comments on my work through the many revisions we worked on were always extended and thoughtful. It always took me quite some time to process your extensive input and I am certain this brought all chapters to a new level. Together we made sure the work was of the highest quality, before sending it to Eldert for his final verdict. Unfortunately you decided to leave the group just before I graduated, but nevertheless for me it feels you completed the journey with me. I am sure you will find your way, as a man with your talents are always in high demand.

I would also like to thank my other supervisor, Jochen Hemming. You have been a valuable colleague to me from the start. Regarding my Ph.D. project you were intentionally put a bit on the background, as you were already very busy running many projects concurrently. However, we must not forget what your supervision still actually implied; a lot of extra work came on your plate. For example, when I decided 'to go on sabbatical for 3 months', *again*. Or all the other tasks should have been delegated to me, but I could not do because I was busy writing. Regarding the latter, you always helped reviewing too; a very time-consuming task, paper after paper. I am grateful for all the extra energy you had to spend to help me graduate. I know those hours were often off the record.

Bart van Tuijl, yes you come next as one of my favourite day-to-day colleagues. How good is it to be befriended with a mechanical engineer for all those times that my software magic could not get things physically moving. With you I could complain about all the things wrong with our institute, life, the universe and everything. Together we made countless field tests work and gathered invaluable data for this thesis.

When I just started working for WUR, it was Erik Pekkeriet who said that I should pick up the work of the cucumber harvesting robot and bring it towards the market. In his view, that would be the highest goal we could achieve as an institute and I still agree. I hope we can make this work, as I am certain we now have all to good cards in our hands to play it well. Rick van de Zedde, Janneke de Kramer, Frans Kampers, Ruud van de Bulk and Bastiaan Berendse, thank you all for supporting the spin-off.

For there are so many other colleagues at Wageningen, I must apologise I cannot mention you all personally. Gerrit Polder, Jos Balendock, Jos Ruizendaal, Angelo Mencarelli, Toon Tielen, Pieter Blok, Hyun Suh, Manya Afonso, Ron Wehrens, Bram van Breugel, Hendrik de Villiers, Gert Kootstra, thank you for being supportive in this journey as I am sure each of you also had to be sometimes flexible to make things work. I also want to welcome our new colleagues in our new group Agro Food Robotics, together we can become a stronghold in our domain.

I would also like to thank the thesis committee for taking the time in travelling all the way to Wageningen to my defence and reading and contemplating about my work. I look forward to our discussion and reflection on the matter.

To my European colleagues in the projects *Crops* and *Sweeper*, I would like give praise to their efforts of the research and development that surrounds this thesis. Without the 7 years of work you have put into this idea, we could not have achieved such a new level of robotisation.

Broadening the scope globally, I first would like to thank Rob for being so kind that I could temporarily be a part of your group. I really enjoyed my times being there and I am sure it was exactly what I needed to focus and get things done. To Alperen Degirmenci, Yashraj Narang, Qian Wan, Paul Loschak, Peter Hammer, Pierre-Frederic Villard and Mohsen Moradi Dalvand, I would like to say thank you for adopting me and showing me around. It was very nice to lunch and party with you guys and I hope we meet again. Leif Jentoft, Lael Odhner and Yaro Tenzer, it was very nice to see your transition from grad students to your spin-off from up close. You have been very busy and I wish you all the success with RightHand Robotics.

Doug, Azu, you two are just a bunch of lovely and fun people. I would never have dared to imagine we could reach this level of friendship. Thank you for being so supportive during our visits and showing us around in Cambridge. I have enjoyed the many culinary explorations, like the Szechuan kitchen or the authentic dumplings we never had before.

Jim Ostrowski, thank you for bringing me in contact with Pietro Perona. I am sure our paths cross again and I wish you all the best with Blue River. Pietro, also you were so kind to offer me a desk in your laboratory and I would like to thank your for your hospitality and generosity. It has been a most fulfilling and productive period in Pasadena. Sara Beery, thank you for showing us around. I really enjoyed touring with you. Oisin Mac Aodha, Joe Marino, Cristina Segalin, Grant Van Horn, Tony Zhang, Matteo Ronchi, Mason McGill and Eli Cole, you were great and relaxed lab mates and I admire your fine working culture of lunching together, going to talks and walking in the afternoon.

To Louis Vuurpijl, who unfortunately left us too early and is not amongst us anymore, thank you for the support and encouragements during my times at Radboud. I am forever grateful that you nominated me for the award I received.

My dear family, Jackie, Toin and Lieve, my apologies for being so involved with working at times. You could have used some extra hands building your own eco-friendly passive house, which is an astonishing achievement you can be proud of. I am thankful for providing a place where I could rest from it all. You have been most supportive and understanding. Jan, thank you for giving me this pair of creative interconnected brains. Our shared interest in photography was one of the reasons why I chose the direction of computer vision. Rob, Ingrid and Karlijn, also thank you for being there when Lianne and I were once again too worked up. You always managed to put things in the right perspective.

Sjors, as one of my best friends I should mention you as well. Without a doubt it was invaluable that I could always crash at your place for a beer and fine food, with a free Hugo included. Unfortunately, due to all of this, Lianne and I were often the first to leave all the lovely Friday and Saturday evenings, our brains already fried from the week. Thank you for understanding, we had some great quality time the past years. Jan and Simone, yes also you I will mention. Simone, you have shown us why it is worthwhile to put a lavish meal on the table and Lianne and I followed this tradition in our home. Without good food, we could not have made it this far. Jan, it was your crazy idea to go to Stanford, saying *"You should go there too, it is amazing!"*. Well, I listened and it was amazing. Thank you for planting this idea in my head.

Sem and Inge, I really love the way you are always up for everything and embrace life as it is whilst not putting pressure on anything. We had some great dinners the past years and we should keep this up! Our travels to Cap d'Antibes will always be in my heart.

To all my other friends from Uden, it has been remarkable that our group is still so close after all these years. Again, Lianne and I were not the most vibrant and active participants due to our commitments, but that does not mean we did not value or enjoyed every bit of our collective leisure. Caroline, thank you for all the afternoon coffee times.

Andrei, thank you for following my suggestion to *choose* Lianne. Life would have been totally different otherwise.

Last but not least, Lianne. I would like to thank you for not only the past 4 years, but all 10 of them. You have been astonishing. Together we have synced into a rhythm of shared interests, science and leisure. We have always joked that, although I gave you a year of a head start, I was going to catch-up in finishing writing the thesis. Well, *technically* by writing these last words that has happened, although I still probably have to do half a year of revisions before Eldert is happy. However, we both know your work and efforts outshine mine regardless. You have so much more academic potential and I am sure you will continue to research what you'll love most. The brain is a wonderful subject and it is funny how our work on biological and artificial neural networks sometimes connects. Thank you for sharing the academic ride with me including all its ups and downs, and in general, this fantastic journey of life.

"We really walked a lot today", said Lianne in the afternoon. *"Well, you have one of these step counters on your wrist, right? Probably we have walked 17962 steps"* said Ruud. *"Haha, yeah right"* she said, meanwhile looking up the number of steps on her wristband, pausing her walk in front of the Einstein papers project house. *"Damn you, 17997 steps."* To which Ruud replied: *"Hmm, still about 30 off"*.

Pasadena, 2018



Model styled graphically to *Brassica oleracea*, also known as Romanesque cauliflower.

Curriculum Vitae

Ruud Barth, born in Uden, the Netherlands in 1988, started his academic career at the Radboud University of Nijmegen. In 2012, he obtained his master's degree in Artificial Intelligence (cum laude) from the Radboud University Nijmegen, including an additional Honours diploma with the highest possible grade (10/10). His master thesis titled *"Hand Gestural Control of Sound"* was partially performed at Stanford University and was awarded with Radboud's best thesis prize, granted annually per faculty. After his academic education, he joined the Greenhouse Horticulture Technology group at Wageningen University and Research, where he focussed on computer vision and robotics research in the domain of agriculture and food. In 2013, Ruud received the *"Young Professional Award"* from the European Machine Vision Association for his work on a selective broccoli harvesting robot. During his time at Wageningen, he was one of the main authors for the granted €4M EU project *Sweeper* and was a work package leader of integration of the €11M project *Pick&Pack*. In parallel to his project management duties and other, more business-to-business research, Ruud initiated pursuing a PhD degree in the summer of 2014. His research for this project was partially performed at Harvard University and the doctoral thesis was written during a sabbatical at the California Institute of Technology. During the end of his PhD track, Ruud started exploring entrepreneurship by founding Saia Agrobotics.



Model styled graphically to a collection of books.

List of Publications

Accepted Journal Publications

- R. Barth, J. IJsselmuiden, J. Hemming, and E.J. Van Henten. Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Computers and Electronics in Agriculture*, 144:284 – 296, 2018c. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2017.12.001. URL `http://www.sciencedirect.com/science/article/pii/S0168169917305689`

- R. Barth, J. IJsselmuiden, J. Hemming, and E.J. Van Henten. Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture*, 2017a. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2017.11.040. URL `http://www.sciencedirect.com/science/article/pii/S0168169917307664`

- R. Barth, J. IJsselmuiden, J. Hemming, and E.J. Van Henten. Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, 146:71 – 84, 2016. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2015.12.001. URL `http://www.sciencedirect.com/science/article/pii/S1537511015001816`. Special Issue: Advances in Robotic Agriculture for Crops

- W. Bac, J. Hemming, B. van Tuijl, R. Barth, E. Wais, and E.J. Van Henten. Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, 34(6):1123–1139. doi: 10.1002/rob.21709. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21709`

Submitted Journal Publications

- R. Barth, J. Hemming, and E.J. Van Henten. Improved Part Segmentation Performance by Optimising Realism of Synthetic Images using Cycle Generative Adversarial Networks. *ArXiv e-prints, submitted to International Journal of Computer Vision: Special Issue on Deep Learning for Robotics*, Mar. 2018a

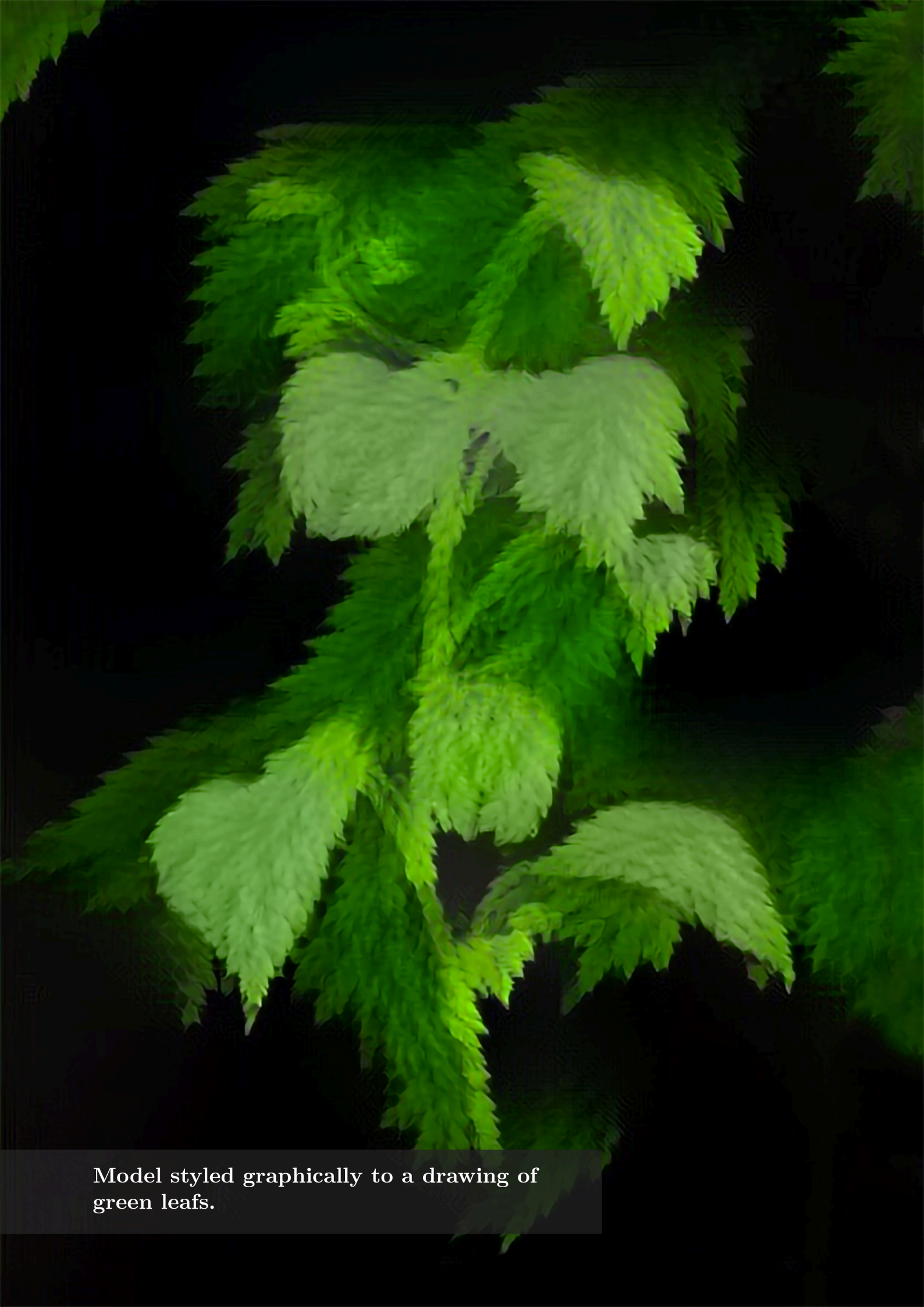
- R. Barth, J. Hemming, and E.J. Van Henten. Estimating angles between fruit and stems to support grasping in a sweet-pepper harvesting. *submitted the Journal of Biosystems Engineering*, Mar. 2018b

Conference Proceedings

- R. Barth, J. IJsselmuiden, J. Hemming, and E.J. Van Henten. *Optimising Realism of Synthetic Agricultural Images using Cycle Generative Adversarial Networks*. Wageningen University & Research, Wageningen, 2017b. URL `http://edepot.wur.nl/434834`
- R. Barth, J. Baur, T. Buschmann, Y. Edan, T. Hellstrom, T. Nguyen, O. Ringdahl, W. Saeys, C. Salinas, and E. Vitzrabin. Using ros for agricultural robotics : design considerations and experiences. In *Proceeding of the International Conference on Robotics and associated High-technologies and Equipment for Agriculture and forestry (RHEA)*, pages 509–518, 2014
- P. M. Blok, R. Barth, and W. van den Berg. Machine vision for a selective broccoli harvesting robot. *IFAC-PapersOnLine*, 49(16):66 – 71, 2016. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2016.10.013. URL `http://www.sciencedirect.com/science/article/pii/S2405896316315749`. 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture 2016
- O. Ringdahl, P. Kurtser, R. Barth, and Y. Edan. Operational flow of an autonomous sweetpepper harvesting robot. 2016. BO-25.06-002-003-PPO/PRI, EU-2015-03, 1409-035 EU. http://edepot.wur.nl/401245
- J. Hemming, C. W. Bac, B. Van Tuijl, R. Barth, J. Bontsema, and E. Pekkeriet. A robot for harvesting sweet-pepper in greenhouses. In *Proceedings of the International Conference of Agricultural Engineering*, 2014

Patent

- B. Van Tuijl, R. Barth, T. Tielen and E. Karruppannan, P6075740NL, Harvesting Device.



Model styled graphically to a drawing of green leafs.

PE&RC Training and Education Statement

With the training and education activities listed below the PhD candidate has complied with the requirements set by the C.T. de Wit Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises of a minimum total of 32 ECTS (= 22 weeks of activities)



Review of literature (6 ECTS)

- Greenhouse horticulture technology (2016)
- Deep learning for agriculture, greenvision (2017)

Writing of project proposal (4.5 ECTS)

- Eye-in-hand Sensing and Visual Servo Control for a Sweet-Pepper Harvesting Robot.

Post-graduate courses (5.5 ECTS)

- ICVSS - International Computer Vision Summer School,
  University of Cambridge, University of Catania (2013)
- Neural Networks for Machine Learning,
  Coursera, Geoffrey Hinton (2017)
  Grade 9.48/10

Laboratory training and working visits (35 ECTS)

- *Harvard University* (2014)
  Visual servo control, simultaneous localisations and mapping.
- *Harvard University* (2016)
  Synthetic image generation for deep learning.
- *California Institute of Technology* (2018)
  Background and scientific context of research.

Invited review of (unpublished) journal manuscript (2 ECTS)

- Biosystems Engineering: agricultural robotics (2016)
- ICRA - IEEE International Conference on Robotics and Automation: deep learning (2017)

Competence strengthening / skills courses (2 ECTS)

- Project Management (2013)
- Ethics in Science; PE&RC (2016)

PE&RC Annual meetings (0.9 ECTS)

- PE&RC Day (2017)
- PE&RC PhD Weekend (2017)

Discussion groups / local seminars / other scientific meetings (9.2 ECTS)

- Seminar: Tegenlicht Meet-up Robotica (2015)
- Seminar: The robots are coming (2015)
- SWEEPER & Pick&Pack Project Meeting (2015)
- SWEEPER & Pick&Pack Project Meeting (2016)
- SWEEPER & Pick&Pack Project Meeting (2017)

International symposia, workshops and conferences (4.1 ECTS)

- European Machine Vision Association conference (2014)
- IROS - International Conference on Intelligent Robots (2017)

Lecturing / supervision of practical's / tutorials (0.6 ECTS)

- ROS-Halcon Workshop (2015)
- Deep Learning Workshop (2016)

Supervision of 2 MSc students (3.5 ECTS)

- Greenhouse robotics systems, the past and future.
- Semantic segmentation of plant images.

Total: 73.3 ECTS


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Colophon

Acknowledgement of Funding
This research was partially funded by the European Commission in the 7th Framework Programme (CROPS GA no. 246252), in the Horizon2020 Programme (SWEEPER GA no. 644313) and by the Dutch horticultural product board (PT no. 14555) and the Dutch Topsector *'Tuinbouw en Uitgangsmaterialen'* (TKI EU-2015-03).

Cover Design
Ruud Barth

Cover Description
"The Green Monster", *a synthetic image of a Capsicum annuum greenhouse scene, with an overlay of an artificial neural network interpretation that was fitted to enclose the fruit by using a neural algorithm of artistic style transfer (https://arxiv.org/abs/1508.06576). Rendering of the cover took 39 hours and 51 minutes on a single Intel i7 CPU.*

Printed by
ProefschriftMaken —— DigiForce

© Ruud Barth, 2018