Automatic imaging system for tulip bulb detection and sorting using deep learning

Dan Rustia, Jos Ruizendaal

Report WPR-1342

WAGENINGEN
UNIVERSITY & RESEARCH

Referaat
Bloembollen worden na de oogst op een sorteerlijn gereinigd en in verschillende ziftmaten gesorteerd. Om de kwaliteit van de bollen uit de sorteerlijn te verbeteren is in dit onderzoek een camera systeem ontwikkeld en getest voor automatische detectie van de pel kwaliteit en bol grootte. Met een camera zijn lijnscanbeelden van bollen verzameld op de transportbanden met behulp van een op maat gemaakte cameraboxen. Elke camerabox was verbonden met een high-performance computing (HPC)-server die de verkregen beelden verwerkt met behulp van een instance-segmentatie-algoritme. Het ontwikkelde algoritme kan succesvol bollen detecteren en classificeren op basis van reinigingskwaliteit en grootte met een gemiddelde F1-score van 0,87 op een commerciële bollen pellijn en een gemiddelde F1-score van 0,94 in een gecontroleerde opstelling.
Er zijn ook tests uitgevoerd om te verifiëren of het beeldvormingssysteem kan worden gekoppeld aan een bollensorteersysteem. De resultaten laten zien dat de opstelling nog verder ontwikkeld moet worden om een betere positiebepaling van de bollen te krijgen, voor een effectieve sortering.

Abstract
This work aims to develop and test an imaging system for automatic detection of bulb cleaning quality and potential sorting of bulbs according to size and cleaning quality. Line scan images of bulbs were collected from conveyor belts using a custom-built camera box. Each camera box was connected to a high-performance computing (HPC) server which processes the acquired images using an instance segmentation algorithm. The developed algorithm can successfully detect and classify bulbs according to cleaning quality and size with an average F1-score of 0.87 and on a commercial bulb sorting setup and an average F1-score of 0.94 in a controlled setup. Tests were also performed to verify if the imaging system can be coupled with a bulb sorting system however results show that further work has to be done in order to achieve better positioning for effective sorting.

Reportinfo

Report WPR-1342
Project number: 3742275802-Bollenrevolutie 4.0 Verwerking
DOI: https://doi.org/10.18174/669966

This project is part of the PPS Bollenrevolutie 4.0 and had as goal to map the quality of the bulbs in the bulb cleaning line. Bollenrevolutie 4.0 is a four-year research program of the 'topsector Tuinbouw & Uitgangsmaterialen van het ministerie van LNV'. This Public-private partnership includes seven partners; KAVB, Anthos, Wageningen University & Research (WUR), Cremer Speciaalmachines B.V., Machinefabriek Steketee B.V., Agrisim B.V., BKD en TechnNature. Economic Board Greenport Duin & Bollenstreek and Rabobank Bollenstreek contribute to the financing from their Innovation funds. A brief explanaition of this PPP can be found on the website www.vitaleteelt.nl

Disclaimer

© 2024 Wageningen, Stichting Wageningen Research, Wageningen Plant Research, Business Unit Greenhouse Horticulture, P.O. Box 20, 2665 MV Bleiswijk, The Netherlands; T +31 (0)317 48 56 06; www.wur.eu/plant-research
Chamber of Commerce no. 09098104 at Arnhem
VAT no. NL 8065.11.618.B01
Stichting Wageningen Research. All rights reserved. No part of this publication may be reproduced, stored in an automated database, or transmitted, in any form or by any means, whether electronically, mechanically, through photocopying, recording or otherwise, without the prior written consent of the Stichting Wageningen Research. Stichting Wageningen Research is not liable for any adverse consequences resulting from the use of data from this publication.

Address

Wageningen University & Research, BU Greenhouse Horticulture
Violierenweg 1, 2665 MV Bleiswijk
P.O. Box 20, 2665 ZG Bleiswijk
The Netherlands
+31 (0) 317 - 48 56 06
+31 (0) 10 - 522 51 93
glastuinbouw@wur.nl
www.wur.eu/greenhousehorticulture

Table of contents

1 Introduction

In the tulip bulb industry, bulb cleaning is a very important task to remove the small newly developed bulbs and old plant residues. One part of bulb cleaning is to sort each bulb according to different conditions. First, bulbs that are too small are usually not ready for commercial flower production, so they are planted back to the ground. Some bulbs may have also been rotten due to root or soil fungi and therefore cannot be sold. Based on this process, several machines in the peeling line are designed to remove excessive pieces of bulb skin and roots to ensure the cleanliness of each bulb. After the mechanical peeling line, a manual labor on a conveyor belt is used to check the quality of the peeling and cleaning of the bulbs. As the settings of the mechanical peeling line are subject to change, due to differences in material entering the line, the settings should be checked regularly. If the peeling is too rough, it will lead to damage to the bulb which can cause potential spreading of fusarium. But if the peeling is too 'soft', it will lead to a lot of manual peeling. As there are differences in the bulbs entering the system, the optimal setting is subject to change. A quality control of the bulbs coming from the mechanical peeling line will enable the possibility to either change the settings of the peeling machine or make a sorting of bulbs in different classes to have manual peeling or not. Therefore, there is a need to design an automated system that can detect several bulb traits including size and presence of skin or roots.

This report discusses about the development of an automated imaging system for detecting and classifying tulip bulbs based on their peeling conditions and size. The specific objectives include the following:

1. To design an imaging system for fast and reliable image acquisition of tulip bulbs going through conveyor belts, taking into account the hardware costs for the setup.
2. To develop an algorithm that can both detect tulip bulbs based on their peeling condition but also measure volume through stereovision (Refer to the related report found at https://edepot.wur.nl/653183 for more information about volume measurements).

2 Materials and methods

2.1 System architecture

The developed system had two versions, as shown in Figure 1. The first version of the system is mainly for automatic detection and classification of tulip bulbs. It includes several camera boxes which are all locally connected with a high-performance computing (HPC) server PC via Ethernet. The HPC server PC has a graphical user interface (GUI) which receives and processes data from the camera boxes. Meanwhile, the second version of the system is designed to be connected to a programmable logic controller (PLC) for potential sorting of tulip bulbs. Further information about the designed systems is discussed in the succeeding sections.

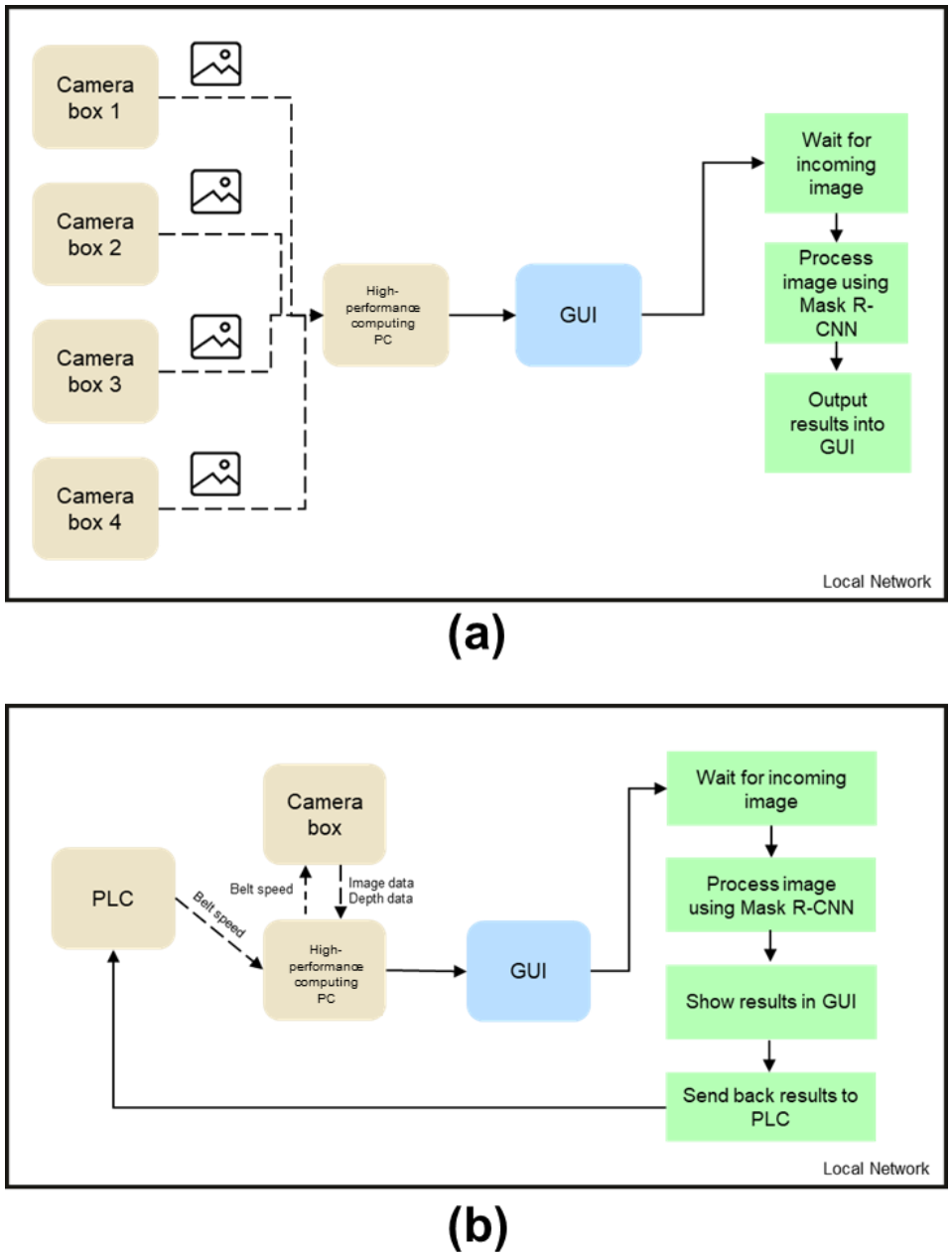

Figure 1 *System architecture: (a) Automatic detection system; and (b) automatic detection system with sorting mechanism.*

2.2 Hardware development

2.2.1 Automatic detection system

The automatic detection system, which is composed of camera boxes and an HPC server PC, was installed in a plant nursery of VaVi Tulips B.V. Each camera box includes an industrial portable PC, white LED lights, and a RealSense D435 camera. Actual images of the camera boxes and its installation setup is shown in **Figure 2**.

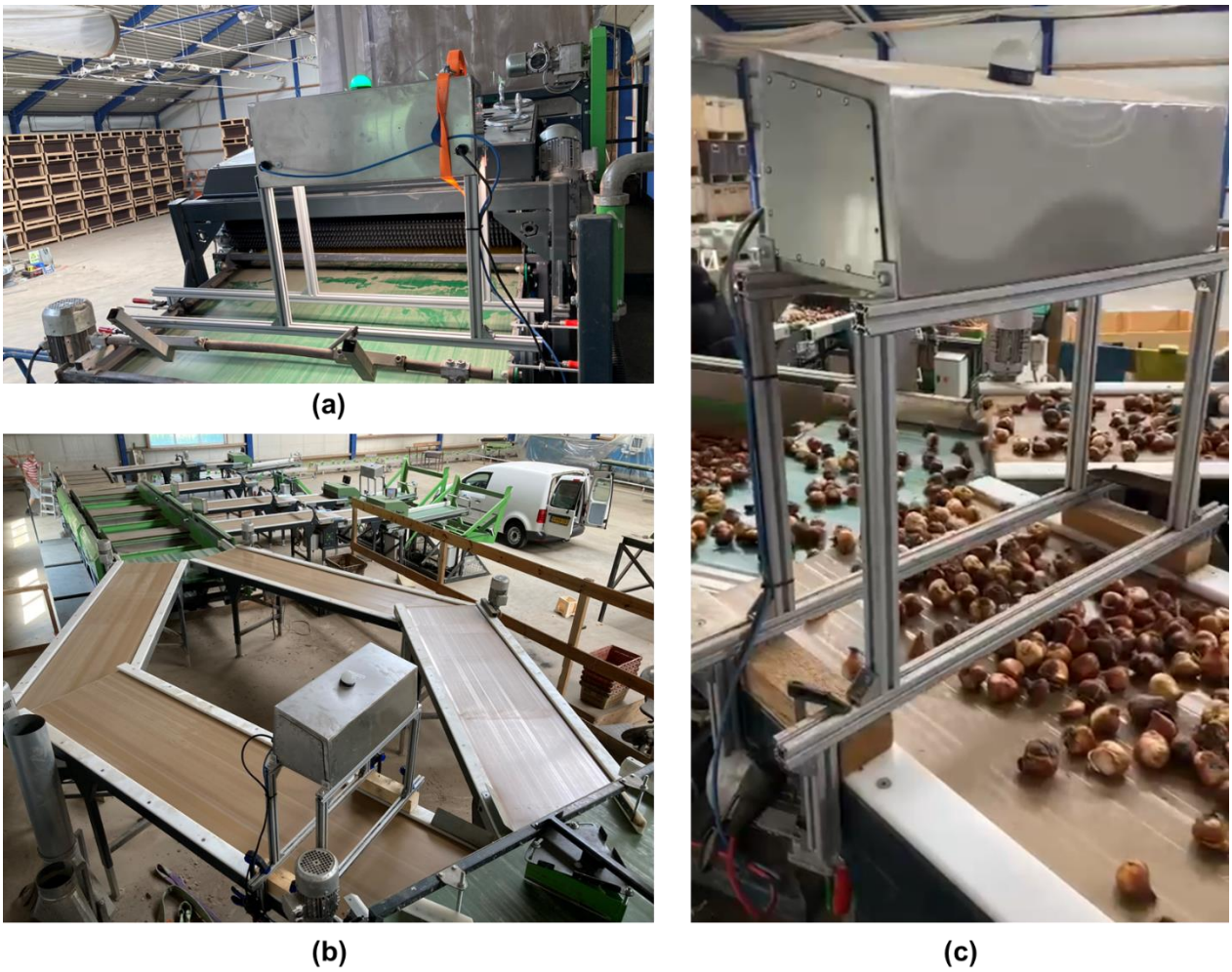

Figure 2 *Automatic tulip bulb detection system setup in VaVi. (a) Camera box for detecting freshly harvested bulbs; (b) Camera boxes for detecting peeled bulbs; and (c) Zoomed in system setup with actual bulbs.*

The first camera box (Figure 1a) detects the number of unpeeled tulip bulbs entering the peeling line. Meanwhile, another camera box (Figure 1b) detects the effectiveness of a root removal system while the last camera box detects all the manually peeled bulbs. The goal of the system in this setup is to detect all the badly peeled bulbs for possibly sorting them and returning the badly peeled bulbs for peeling again.

Communication between the HPC server PC and the camera boxes were performed using ZeroMQ. The HPC server PC includes an Intel Xeon E5-1650 CPU and an NVIDIA GTX Titan GPU.

2.2.2 Automatic detection system with sorting mechanism

The automatic detection system with sorting mechanism is composed of a camera box, HPC server PC, and a programmable logic controller (PLC), as shown in **Figure 3**.

Figure 3 *Automatic tulip bulb detection and sorting system setup in Cremer.*

The main goal of the system is to detect the peeling condition of the bulbs while sorting them. In so doing, the PLC is responsible for measuring the conveyor speed through an encoder and controlling a rejection valve. The PLC transmits the conveyor speed to the HPC server PC while the HPC server PC forwards the speed to the camera box. Detailed information about the communication between the system components are discussed in the later sections.

To enable faster processing, the HPC server PC is equipped with an AMD Ryzen-9 CPU and an NVIDIA GeForce RTX 4080 GPU.

2.3 Software development

2.3.1 Automatic tulip bulb detection algorithm

This section discusses the pre-processing, collection, and analysis of tulip bulb conveyor images.

2.3.1.1 Image pre-processing and data collection

There are two ways in which the images were stitched to generate line scan images. In both ways, the conveyor belt speed was obtained and used to determine how frequently should one row of an incoming image be added to a buffer of a stitched line scan image.

In the automatic detection system without sorting mechanism, the conveyor belt speed was measured by calculating the percent difference between a region of interest of both the current image and an incoming image. The coordinates and sizes of the region of interest was pre-defined arbitrarily per camera box based on the lighting condition and appearance of the conveyor belt, which eventually controls the sensitivity of stitching. This process produces 1920 x 1920 pixels line scan images. To resolve potential detection loss due to stitching, an overlap area of 180 pixels was set at the bottom of the 1920 x 1920 pixels line scan image.

On the other hand, in the automatic detection system with sorting mechanism, the conveyor belt speed was obtained from a PLC via EtherCAT protocol which measures the conveyor belt speed through an encoder. This dynamically adjusts the stitching speed based on the current conveyor belt speed.

2.3.1.2 Data preparation

Two datasets were prepared in this project. The VaVi dataset is composed of line scan images, with an image resolution of 1920 x 1920 pixels, that were obtained from VaVi. On the other hand, the Cremer dataset is composed of line scan images with half of the original line scan image resolution. The line scan images were stitched only until 960 pixels in length since it was necessary to optimize the inference time for each line scan image. The statistical summary of the datasets is shown in **Table 1**, while sample images of each dataset are shown in **Figure 4**.

Table 1 *Tulip bulb conveyor line scan image statistical summary.*

Dataset name	Number of line scan images	Line scan image size (L x W)	Class	Number of images
VaVi 1	77	1920 x 1920	*Controle* (Clean)	1618
			Slechtgepeld (Badly peeled)	1261
			Kaal (Bald)	59
			Peer	94
VaVi 2	102	1920 x 1920	*Controle*	4246
			Slechtgepeld	5547
			Kaal	160
			Kleintje (Small)	333
			Velletje (Skin)	148
VaVi-aug	102	1920 x 1920	*Controle*	4200
			Slechtgepeld	5433
			Kaal	1933
			Kleintje	1309
			Velletje	1177
VaVi-test	40	1920 x 1920	*Controle*	1410
			Slechtgepeld	1183
			Kaal	48
			Kleintje	99
			Velletje	233
Cremer	305	960 x 1920	*Controle*	625
			Slechtgepeld	650
			Kaal	425
			Kleintje	398
			Velletje	334
Cremer-test	31	960 x 1920	*Controle*	95
			Slechtgepeld	177
			Kaal	95
			Kleintje	72
			Velletje	23

VaVi 1 is a dataset obtained for initially testing the functionality of the system, with its sample images in Figure 4a. After proving that the system worked using the VaVi 1 dataset, the VaVi 2 dataset was collected for fully running the system and letting the managers in VaVi to test the system. The Vavi-test dataset is an independent testing dataset collected from VaVi which does not contain any image from VaVi 1 and VaVi 2.

But as it can be seen from Table 1, there is a huge imbalance in the number of objects for each class in the VaVi 2 dataset. Therefore, a variation of the VaVi 2 dataset was prepared, namely the VaVi-aug dataset, as shown in Figure 4b. The VaVi-aug dataset was prepared by applying foreground inpainting (Xiong et al., 2019) as illustrated in **Figure 5**. The aim of foreground inpainting was to reduce the effects of class imbalance, which eventually affects detection and classification performance. In so doing, each object from the annotated VaVi dataset was used as an object dictionary for overlaying. Specifically, only the *Kaal*, *Kleintje*, and *Velletje* classes were selected since there were too few instances of the three classes. Then, the presence of objects in a pre-defined area covering the conveyor belt (not including the rails and bars), illustrated in Figure 5 as a dashed line, was checked. Next, an object for the dictionary was randomly selected and overlaid to a specific location without a foreground object, as illustrated using yellow dotted lines in Figure 5. This process was done for 10 times on each line scan image.

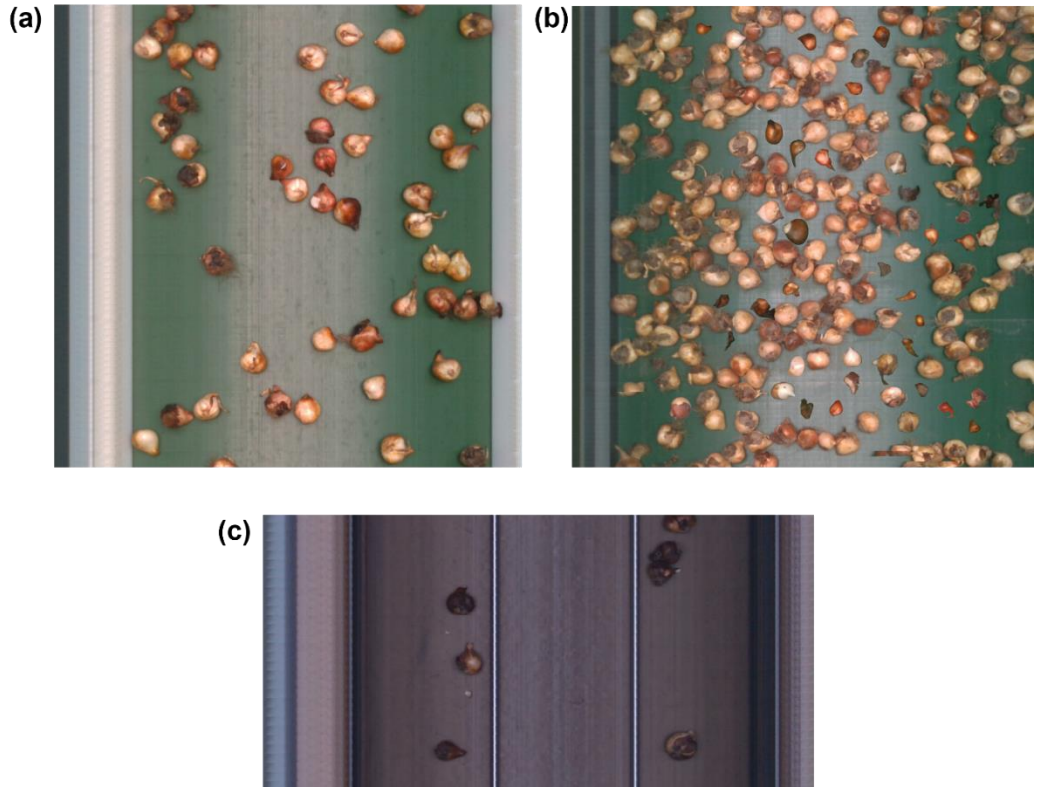

Figure 4 *Sample line scan images from each dataset. (a) VaVi; (b) VaVi-aug; and (c) Cremer.*

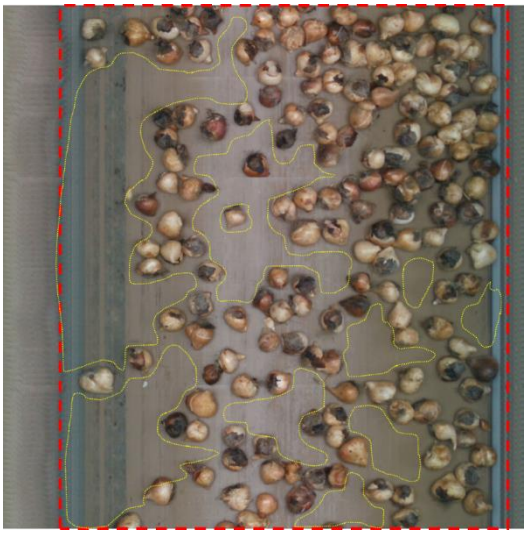

Figure 5 *Sample image for inpainting. The red dashed line represents the pre-defined area for inpainting while the yellow dotted lines represent the potential areas for overlaying new objects.*

2.3.1.3 Instance segmentation algorithm

Each line scan image was processed using an instance segmentation algorithm through deep learning. The instance segmentation model used in this project is Mask Region-based Convolutional Neural Network (R-CNN) (He et al., 2017) with a Point-based Rendering (PointRend) (Kirillov et al., 2019) module which can extract finer segmentation predictions compared to a standard Mask R-CNN model. Furthermore, PointRend was employed since the instance segmentation results shall be used for volume estimation. The instance segmentation algorithm outputs the class label, bounding boxes, and mask boundaries of each object.

In training the model, both Mask R-CNN and Mask R-CNN with PointRend module was tested for comparison with similar training configurations including 800 x 800 pixels model input size, 1500 training iterations, and batch size of 2 using Detectron2 deep learning framework. Learning rate was iteratively tested with values of 0.005, 0.001, and 0.0005. All training and testing processes were performed using the same specifications as the HPC server PC.

2.3.1.4 Post-processing

One of the main caveats of stitching line scan images is resolving the redundant presence of an object in two succeeding images. To resolve this issue, an overlapping object post processing solution was done, as illustrated in Figure 6. First, the bottom 90 pixels of the line scan image (illustrated as a yellow solid box in Figure 6) at t=0 was copied as the first 90 pixels of the next image at t=1, where t represents time. The 90 pixels was determined from the average size of a bulb, as measured from previous trials. Each object inside the 90 pixels area of the image at t=0 and the current line scan image at t=1 was matched using intersection over union (IoU), as computed using Equation 1. *IoU* is measured from 0 to 1.0 where values closer to 1.0 has better overlap. If the *IoU* between two objects were more than 0.5, then it was considered as a match. For the first case, if the object is detected in both images, then the object at t=0 was ignored while the object at t=1 was counted as a detection. For the second case, if the object is only detected at t=1, then the object detected at t=1 was counted as a detection. For the third, if the object is only detected at t=0, then the object at t=0 was considered as a detection. For the final case, if an object is detected at t=1 between pixels 85 to 95 of the overlap area, then a match is checked at the previous image at t=0 and ignored if a match is detected,

Figure 6 *Overlapping object post processing for stitched line scan images. (a) Line scan image at t=0, and (b) line scan image at t=1.*

$$IoU = \frac{A \cap B}{A \cup B}$$

Equation 1 *Intersection over union.*

2.3.1.5 Evaluation metrics

The instance segmentation algorithm was evaluated using two metrices: F1-score and *IoU*. F1-score is measured as the harmonic mean of precision and recall (Rustia et al., 2021), and computed as shown in Equation 2, where *TP* = true positive, *FP* = false positive, and *FN*= false negative. In this work, a detection is considered a *TP* if it has a 0.5 *IoU* match with the bounding box annotations. Meanwhile, *IoU* for evaluation follows the same definition as Equation 1 but instead of using bounding boxes, the individual points were matched between the mask annotations and mask predictions.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 - score = \frac{2 * precision * recall}{precision + recall}$$

Equation 2 *Precision, recall, and F1-score.*

2.3.2 User interface

Two versions of a user interface were designed for two testing sites. Both user interfaces were designed using PyQt5. On both interfaces, a user can click the *Start collection* button to initiate communication between the camera box and the HPC server PC, while turning on the LEDs of the camera box. Each image sent by the camera box is then processed by the HPC server PC using the instance segmentation algorithm. After each collection, such as after filling up a bulb crate, a user can click *Stop collection* then it will reset the counts for each camera box. The user can also opt to save the .csv files which contain the number of objects detects, along with the date and time of the collection. The raw and processed .jpg files can also be saved for future reference.

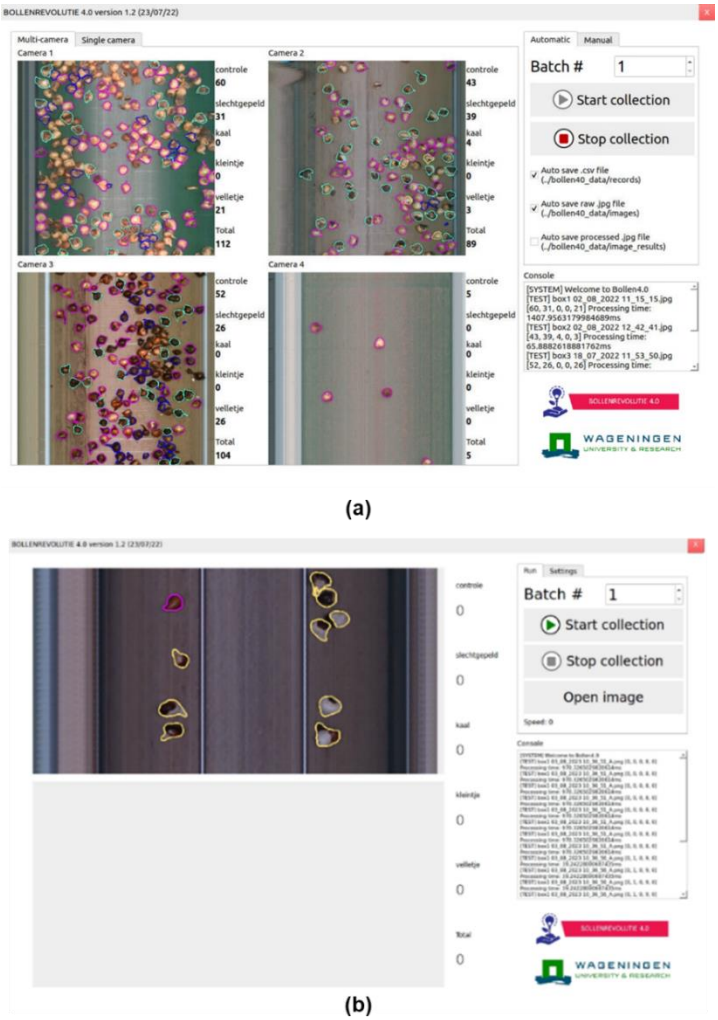

Figure 7 *User interface for automatic tulip bulb detection and sorting in (a) VaVi and (b) Cremer.*

3 Results and discussion

3.1 Instance segmentation algorithm evaluation

3.1.1 VaVi model without inpainting

The instance segmentation model used for the VaVi dataset was optimized and evaluated as shown in **Figure 8**. It can be clearly seen that, without data augmentation such as inpainting, the model performed satisfactorily with a highest F1-score of 0.868 using the Mask R-CNN model with PointRend module and Resnet101 backbone at a learning rate of 0.001. Although different from the results with inpainting augmentation, it shows that the Mask R-CNN model with a learning rate of 0.005 and Resnet50 backbone had the highest average *IoU*. However, it was observed in all models that the classification performance was very poor, even for the model with best average F1-score, as shown in **Figure 9**. This means that augmentation was necessary since the model was unable to properly classify the minority classes.

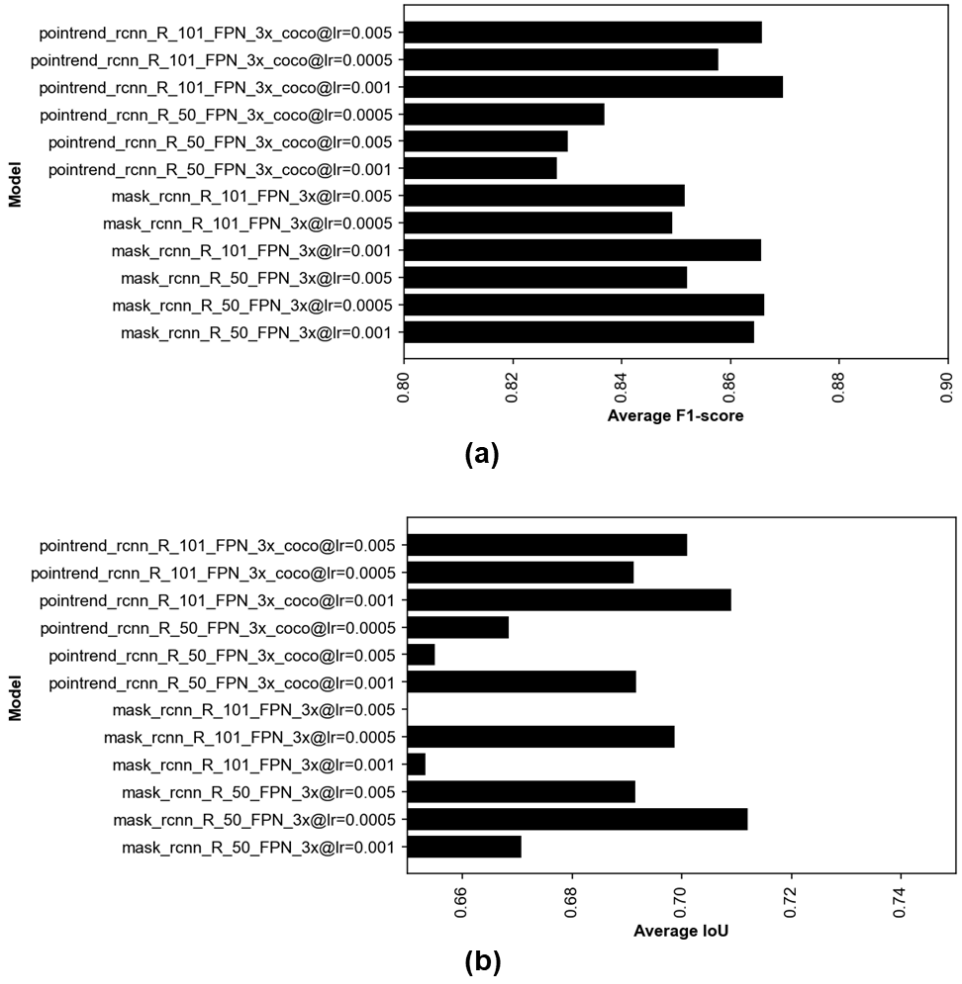

Figure 8 *Instance segmentation model optimization results for the VaVi dataset.*

3.1.2 VaVi model with inpainting

To improve classification performance, the foreground inpainting was applied to the training images of the VaVi dataset, with results shown in Figure 10. It can be easily noticed that there was a boost in performance performance among all the models in terms of F1-score, which can be observed most especially in one of the confusion matrices, as shown in Figure 9. Moreover, the average *IoU* of the models also increased while having the Mask R-CNN model with PointRend module, Resnet101 backbone, and learning rate of 0.001, had an average F1-score and average IoU of 0.882 and 0.745, respectively; the model was therefore used for final deployment of the system.

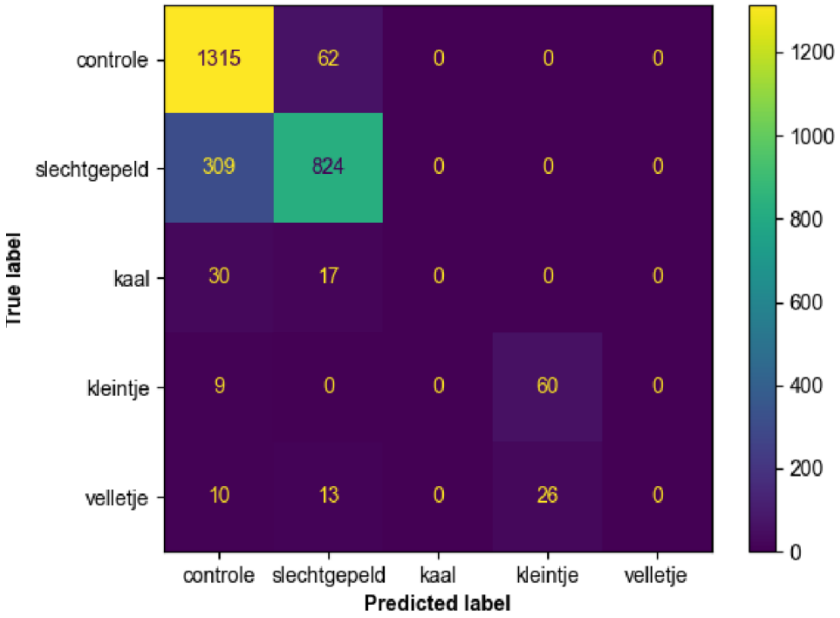

Figure 9 *Confusion matrix of the pointrend_rcnn_R_101_FPN_3x_coco@lr=0.001 model trained on the VaVi dataset.*

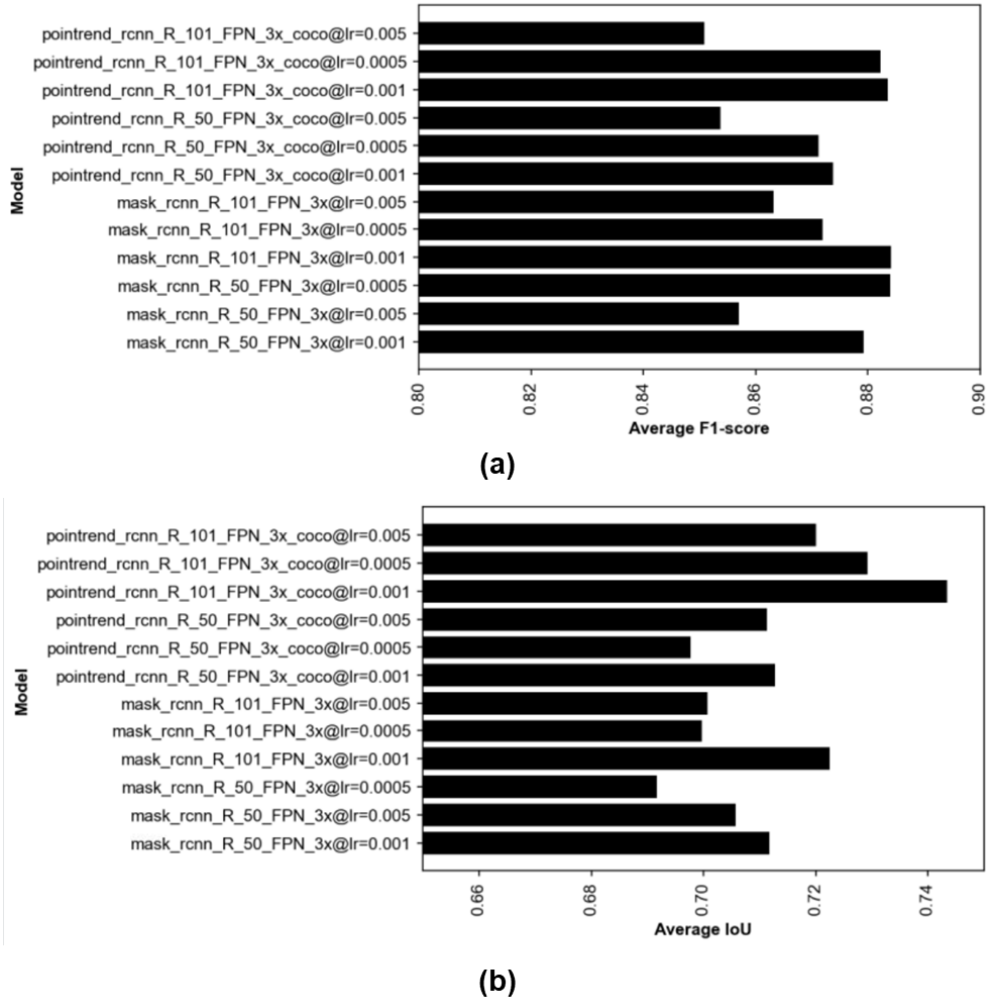

Figure 10 *Instance segmentation model optimization results for the VaVi-aug dataset.*

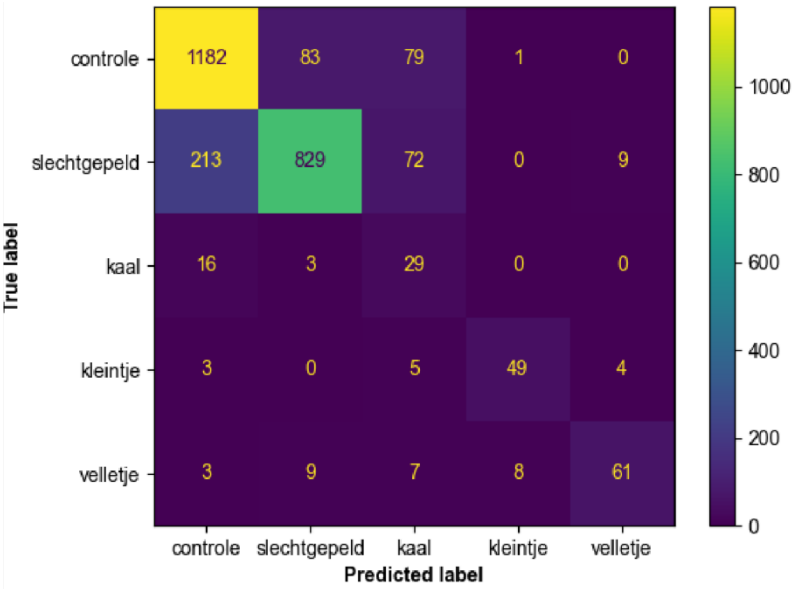

Figure 11 *Confusion matrix of the pointrend_rcnn_R_101_FPN_3x_coco@lr=0.001 model trained on the VaVi-aug dataset.*

3.1.3 Cremer model

Model performance on the Cremer dataset was evaluated as shown in Figure 12. It can be evidently seen that the performance of the best model (F1-score=0.94) was better than the model trained on the VaVi dataset. This was mainly because the images from the Cremer dataset were simpler and had less objects on each image. Among the trained models, the Mask-RCNN with Resnet50 backbone and learning rate of 0.005 had the best F1-score and average IoU of 0.98 and was therefore used as the model for deployment. It can also be seen in Figure 13 that the model had a good classification performance since the Cremer dataset had a better class distribution.

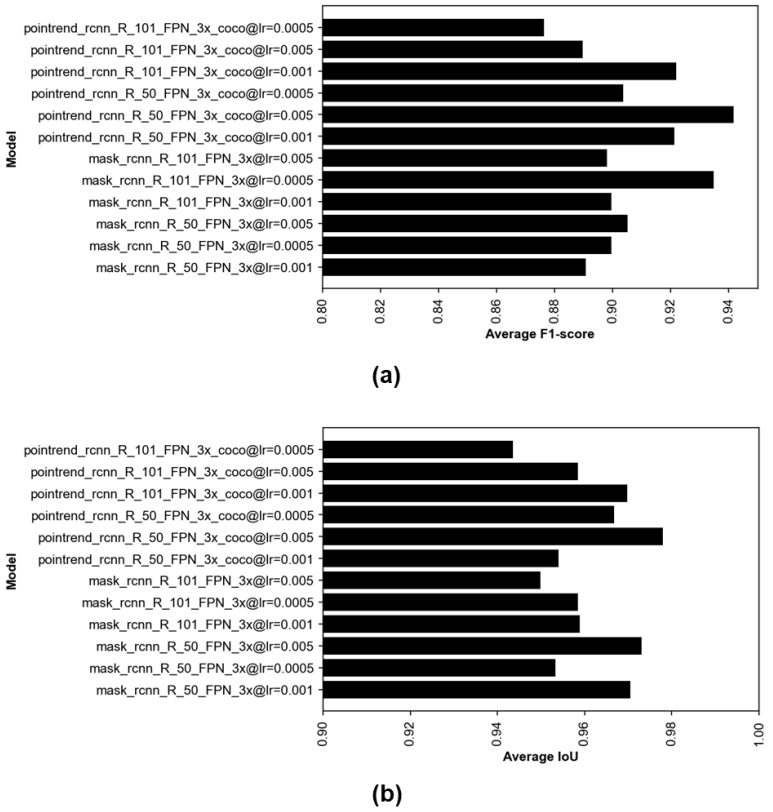

Figure 12 *Instance segmentation model optimization results for the Cremer dataset.*

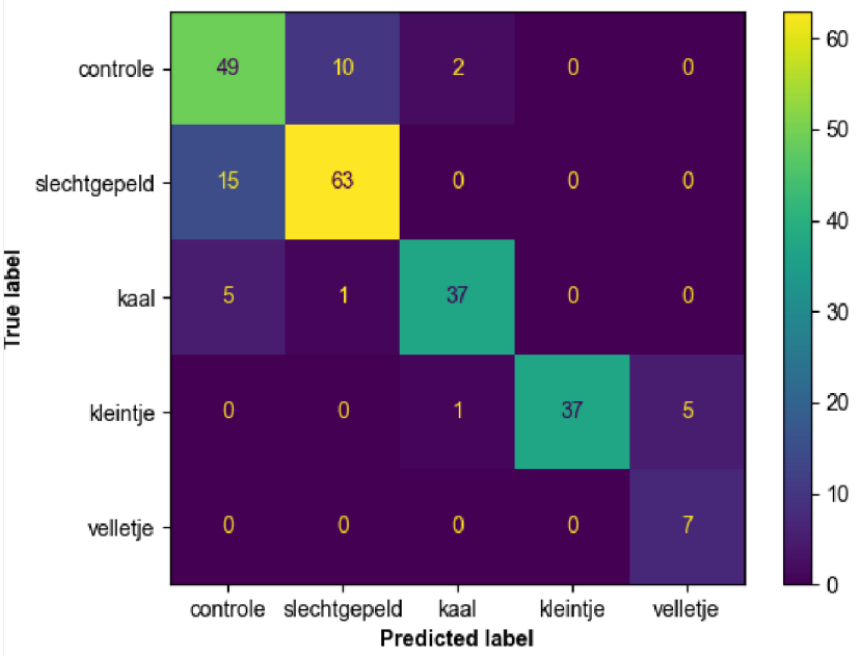

Figure 13 *Confusion matrix of the pointrend_rcnn_R_50_FPN_3x_coco@lr=0.005 model trained on the Cremer dataset.*

3.1.4 Qualitative model testing

Sample segmentation results done by the instance segmentation model are shown in Figure 14. It can be clearly seen that the mask output results were very sharp and matched well with the depth image. However, it can be seen that there were a few pixels outside of each bulb that were considered as part of the individual masks, as shown in Figure 14c. Even though unseen in these results, stitching issues were also observed in which same pixel lines were copied to the image due to delays in acquiring an image from the camera.

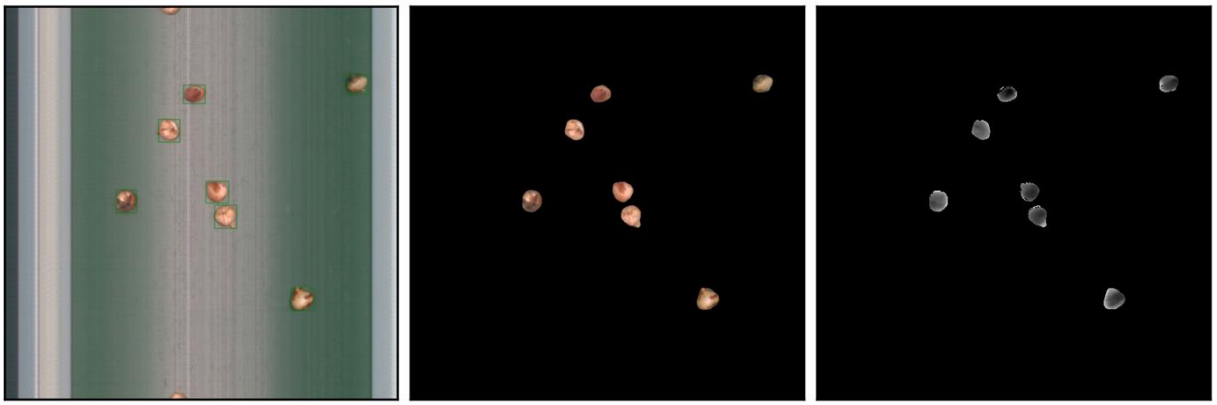

Figure 14 *Sample inference output results of the instance segmentation model: (a) Output image; (b) segmentation results without the background; and (c) segmentation results mapped to the depth image.*

3.1.5 System integration test for bulb sorting

The Cremer instance segmentation model was deployed in an actual system connected to a sorting component, as described in Figure 3. To verify if sorting can be done, latency tests were performed, as shown in **Table 2**.

Table 2 *Communication latency test.*

Operation	Average delay (ms)
Receiving data from camera box and getting position data from PLC	29ms
Image processing/analysis with objects detected	98ms
PC to PLC transmission	600ms
Total	727ms

The latency test showed that image processing was relatively fast with only about 98ms delay. It should be noted that multi-threading was applied in order to achieve this processing speed. It can also be seen that there was a considerable delay of around 727ms in order to push the bulbs for sorting. Based on these results, the average positioning error of the bulbs was about +/-3mm at 50mm/s belt speed and +/-10mm at 310mm/s. This means that the error increased as the belt speed increased. Unfortunately, using a belt speed of 50mm/s was not fast enough for bulb sorting. Therefore, it is highly recommended to resolve this issue before integrating the whole system. Another recommendation is to reduce the communication channels such as by doing all operations such as sorting and image analysis in the server PC alone.

References

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017, 22-29 Oct. 2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV).

Kirillov, A., Wu, Y., He, K., & Girshick, R. B. (2019). PointRend: Image Segmentation As Rendering. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9796-9805.

Rustia, D. J. A., Chao, J.-J., Chiu, L.-Y., Wu, Y.-F., Chung, J.-Y., Hsu, J.-C., & Lin, T.-T. (2021). Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. *Journal of Applied Entomology*, *145*(3), 206-222. https://doi.org/https://doi.org/10.1111/jen.12834.

Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., & Luo, J. (2019, 15-20 June 2019). Foreground-Aware Image Inpainting. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

To explore the potential of nature to improve the quality of life

Wageningen University & Research,
BU Greenhouse Horticulture
P.O. Box 20
2665 ZG Bleiswijk
Violierenweg 1
2665 MV Bleiswijk
The Netherlands
T +31 (0)317 48 56 06
F +31 (0)10 522 51 93
www.wur.nl/glastuinbouw

Greenhouse Horticulture Report WPR-1342

The mission of Wageningen University & Research is "To explore the potential of nature to improve the quality of life". Under the banner Wageningen University & Research, Wageningen University and the specialised research institutes of the Wageningen Research Foundation have joined forces in contributing to finding solutions to important questions in the domain of healthy food and living environment. With its roughly 30 branches, 7,700 employees (7,000 fte), 2,500 PhD and EngD candidates, 13,100 students and over 150,000 participants to WUR's Life Long Learning, Wageningen University & Research is one of the leading organisations in its domain. The unique Wageningen approach lies in its integrated approach to issues and the collaboration between different disciplines.