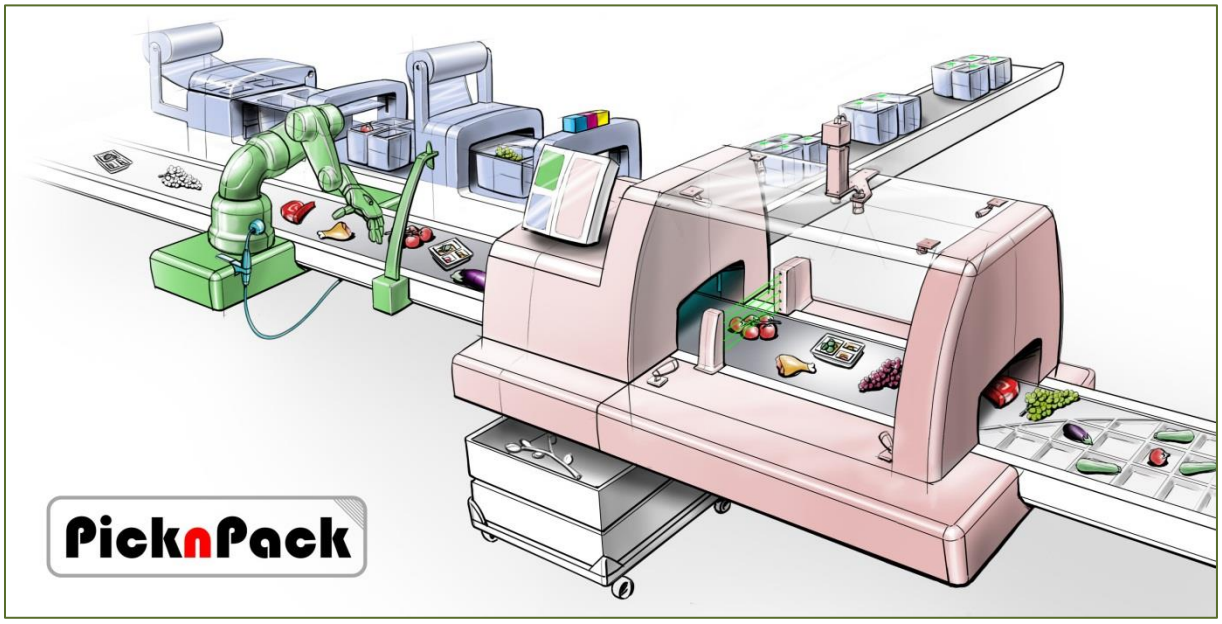
Report on the performance of the QAS module

Detecting quality deficient products and shape and position assessment on irregular products

Jeroen van Roy (KUL), Niels Wouters (KUL), Gert Kootstra (DLO), Zhaozong Meng (UM), Zhipeng Wu (UM), Mattias Van Dael (KUL), Pieter Verboven (KUL), Wouter Saeys (KUL)

9/20/2016



Flexible robotic systems for automated adaptive packaging of fresh and processed food products

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n° 311987.

Table of Contents

Dissemination level		
PU	Public	X
PR	Restricted to other programme participants (including the EC Services)	
RE	Restricted to a group specified by the consortium (including the EC Services)	
CO	Confidential, only for members of the consortium (including the EC Services)	

1 Introduction

This deliverable report gives an overview of the performance of the Quality Assessment and Sensing (QAS) module as was achieved after the full integration of the line in Wageningen (WUR). The functionality and capabilities discussed in this report were demonstrated during the demonstration days in Wageningen at the end of May 2016.

This deliverable is a result from Tasks 4.4 (Development of a quality sensing platform) and 4.5 (Evaluation of the quality assessment method on relevant food products).

2 Functionality of the QAS module

The main functionality of the QAS module concerns the measurement and rating of the quality of *all* products that pass through the module. The term quality should be interpreted in a broad sense: It can refer to the assignment of a product to a certain quality class, but it can also concern the measurement of more specific product properties e.g. damage detection or brix estimation.

This functionality is realized by applying a series of sensors which each measure different properties of the products. The sensors are divided over two submodules termed submodules “A” and “B” (Figures 1, 2 and 3). Submodule A is constructed in a collaboration between KUL, WUR and MU and contains an RGB camera, a 3D laser scanner, a microwave scanner and a hyperspectral camera (HSI, 600 – 1000 nm). Submodule B contains an X-ray unit and is constructed by InnoS.

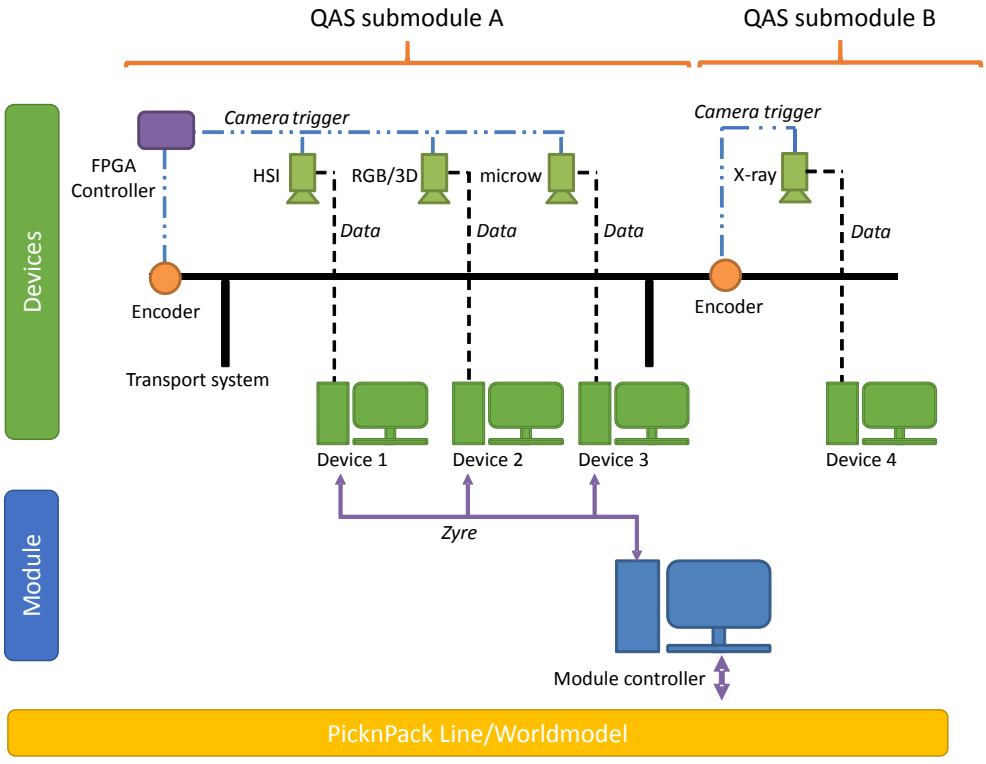


Figure 1. Schematic overview of the PicknPack QAS module.

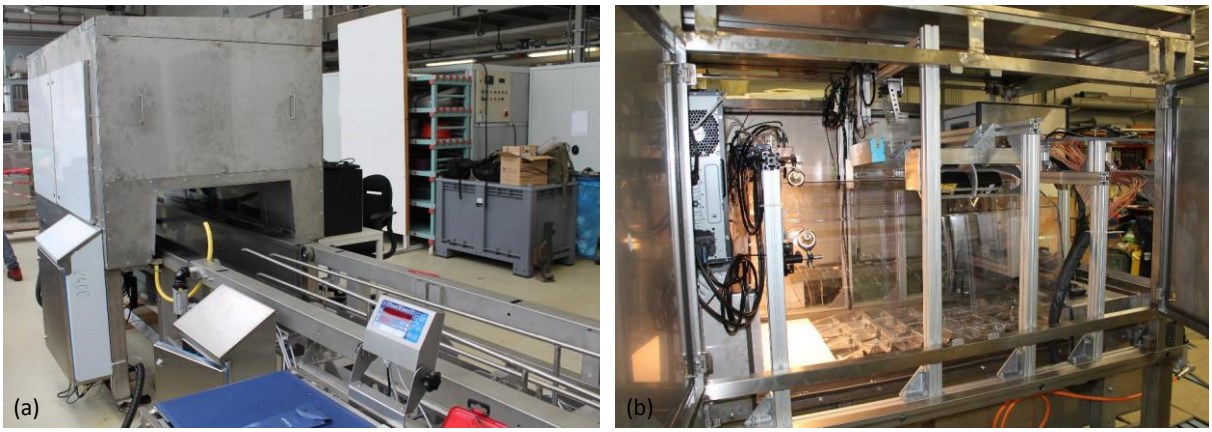


Figure 2. (a) Montage of submodule A in the PicknPack Line. (b) View of the interior of the QAS submodule A. Each of the sensors is mounted on a bridge over the web of packages.

Figure 3. Montage of the QAS submodule B (X-ray) in the PicknPack Line.

As illustrated in Figure 1, each QAS sensor is connected to a dedicated computer unit (called a "device") which collects and processes incoming data. Data acquisition is controlled by an FPGA (field programmable gate array) controller which translates the speed of the line as captured by encoders into a trigger for the sensors.

The devices are connected to a central module controller which also interacts with the PicknPack line as a whole. Communication is done according to the protocol developed in the PicknPack project (based on the zyre protocol).

More information on each of the sensors and the assembly of the (sub-) modules can be found in deliverable reports D4.5 and D4.6 and D4.7.

The QAS module is equipped with flexible self-learning software tools which should enable operators to quickly and easily train the algorithm to deal with new products or product features. These software tools are described in more detail in deliverable reports D4.8. and D4.9.

The remainder of this report provides an overview of the realized functionality and performance of the QAS module. The performance of the module is judged by means of two types of performance indicators:

Performance of the module as a measuring system: This pertains to the more fundamental capabilities of the system needed to perform quality inspection in an industrial environment. These include:

- The speed and accuracy at which all sensors can acquire incoming data
- Spatial calibration related aspects which are important for sensor fusion
- The correct behaviour of the QAS module in the PicknPack communication network
- Safety aspects
- Hygienic design

Performance of the module as a processing system: This pertains to the more advanced capabilities of the system in that it can generate useful product information that can be used in the remainder of the line (printing, sorting,…). More specifically, this report discusses:

- The product properties that can be measured and the correctness and accuracy of those measurements.
- The ease with which the module can be operated/trained (flexibility).

2.1 Performance of the module as a measuring system:

2.1.1 Communication network

The communication protocol developed in the PicknPack project has been successfully implemented and tested in the QAS module. The network topology used is shown in Figure 1: The PicknPack line communicates with the module controller which in turn communicates with the QAS devices.

A more detailed overview of the most important capabilities of the module controller and the devices is listed here:

QAS module controller

- Monitors the presence and activity of QAS devices and acts appropriately
- Sends/Receives information from the line/world model and redistributes as needed
- Collects/stores processed data
- Calculates new features by combining data from the devices (sensor fusion)

Devices

- Exchange information/data with the module controller
- Acquisition and processing of (raw) data

The communication protocol was implemented for the module controller (LabVIEW), HSI camera (LabVIEW), RGB camera (C++), 3D scanner (C++) and microwave scanner (C#). The communication protocol could not be fully implemented on the X-ray unit due to the late stage at which InnoS joined

the PicknPack project. However, given that the communication protocol works well on the other devices, it is safe to assume that the protocol can be successfully implemented on the X-ray unit as well if more time can be spent.

2.1.2 Speed and accuracy of data acquisition

All sensor systems were able to deal with the stop and creep regime as dictated by the PicknPack line. More specifically this pertains to a stop phase of about 10 s and a creep phase under 2 s long, during which a maximum speed of about 16 cm/s is reached and a total distance of 250 cm is traversed.

During acquisition the FPGA controller (Figure 1) monitored the movement of the line and produced correct custom trigger signals for all sensors. No data lines were missed and images of products/trays/punnets could be correctly reconstructed from the raw data feed.

The sensors achieved the following resolutions:

- HSI: 0.4 mm / pixel
- RGB: 0.25 mm /pixel
- 3D: 1 mm / pixel
- Microwave: 2.1 mm / pixel in direction of motion, 21 mm / pixel in perpendicular direction
- X-ray: 0.135 mm / pixel (= 0.027 mm / pixel @ 5x binning)

2.1.3 Spatial calibration

All sensors were calibrated spatially, i.e. the relation between the local coordinate system of each sensor was determined relatively to the coordinate system of the line as a whole. Using this calibration, the sensors were able to correctly extract regions of interest (ROI's, being trays/punnets) from the recorded data at runtime, based on the tray information provided by the world model/line controller. In Figure 4, three examples of the ROI extraction are shown.

(a) (b) (c)

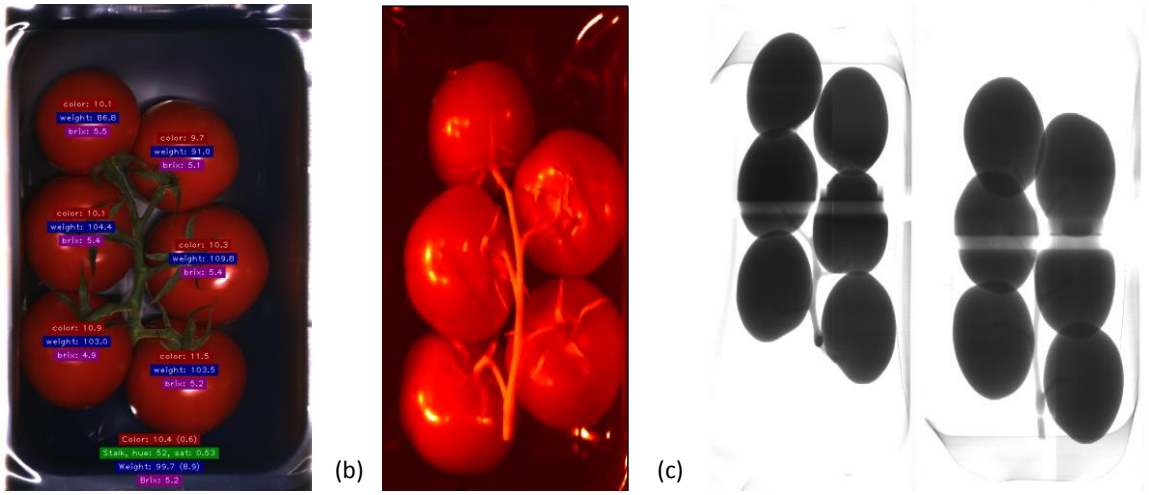


Figure 4. (a). ROI as extracted from the RGB data. Quality features are printed on top of each tomato. (b) ROI as extracted from the HSI data (fake colour image, only a single waveband is shown). (c) ROI as extracted from the dual X-ray data.

2.1.4 Safety related aspects

Safety was taken into account during the operation of the module. The most important safety measures were:

- Installation of emergency stops
- Installation of door sensors that shut down the laser system of the 3D scanner as soon as the doors of the module are opened
- Appropriate shielding of the X-ray and microwave scanners
- Examination and approval of the QAS module by a designated safety inspector

During operation of the line the safety systems performed as expected.

2.1.5 Hygienic design

Both QAS submodules were designed with hygienic criteria in mind, e.g. by using stainless steel as construction material or by making all surfaces slanted to avoid build-up of contamination.

Submodule A of the QAS module (see Figure 2(b)) is equipped with a tunnel which was designed according to industrial design criteria. The sensors are installed above the tunnel so the part of the module that comes into contact with food products can be cleaned properly and safely.

The X-ray unit (submodule B) was also designed according to industry standards and is fully cleanable.

2.2 Performance of the module as a processing system

The capabilities of the QAS module described in the previous paragraphs serve as a basis to be able to correctly process the incoming data. More specifically, the QAS module and its devices acquire raw data and extract from it ROI's per tray/punnet which are linked to information obtained from the worldmodel (position, content). This data is then processed using the various data models developed during the PicknPack project. Once the processing is completed, the features obtained per tray/punnet are uploaded to the worldmodel for storage or further use down the line (e.g. printing of custom labels).

2.2.1 Measured product properties

In this paragraph, the results obtained are discussed for each of the sensors separately. Generally speaking for each sensor, first a segmentation step is performed to separate the plastic tray background from the objects of interest and to split an object (e.g. tomato truss) into its components (e.g. individual tomatoes and stalk). Further processing is then conducted on the segmented objects.

Hyperspectral Imaging Sensor (KUL)

Tomato trusses

Tomato trusses are segmented by applying a PCA (principal component analysis) model that differentiates between stalks and tomato flesh. Subsequently, individual tomatoes are segmented by spatial processing (based on the Hough transformation).

The HSI setup was used to create models to determine:

- the **brix-value** (sugar content) of individual tomatoes
- The **ripeness** of individual tomatoes
- the presence of **skin damage**

Figure 5 shows the results achieved by the model that determines the Soluble Solid content (SSC) (expressed as Brix-degrees). An R^2-value of 0.53 was achieved (Table QAS1). This low value is explained by the fact that for tomatoes the variation in sugar-content between tomatoes is generally low.

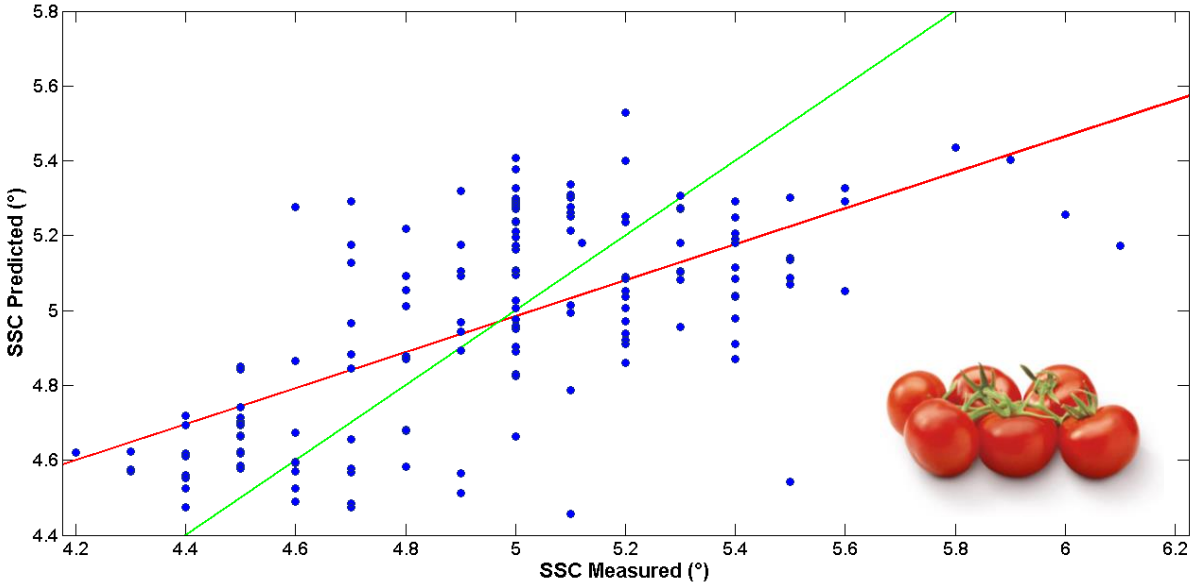


Figure 5. Model that predicts the Brix-value in tomatoes. Measured soluble solid content versus predicted.

Table 1. Overview of the results obtained by the data models for determining tomato brix values (SSC) and colour (Hue H).*

	R^2	R^2_p	RMSEC	RMSEP
L*	0,86	0,86	1,56	1,57
a*	0,95	0,93	1,86	1,89
b*	0,55	0,42	1,92	2,29
H*	0,97	0,95	3,01	3,35
SSC	0,53	0,23	0,25	0,31

The colour of the tomatoes is predicted by estimating the L-, a- end b-values of the tomatoes in the Lab colour space. The predicted a- and b-values are then used to estimate the Hue value for each

tomato (Figure 6 and Table 1). A good correlation (r^2 of 0.97) was obtained. This hue-value can then be linked to the ripeness class of which the tomato is a member.

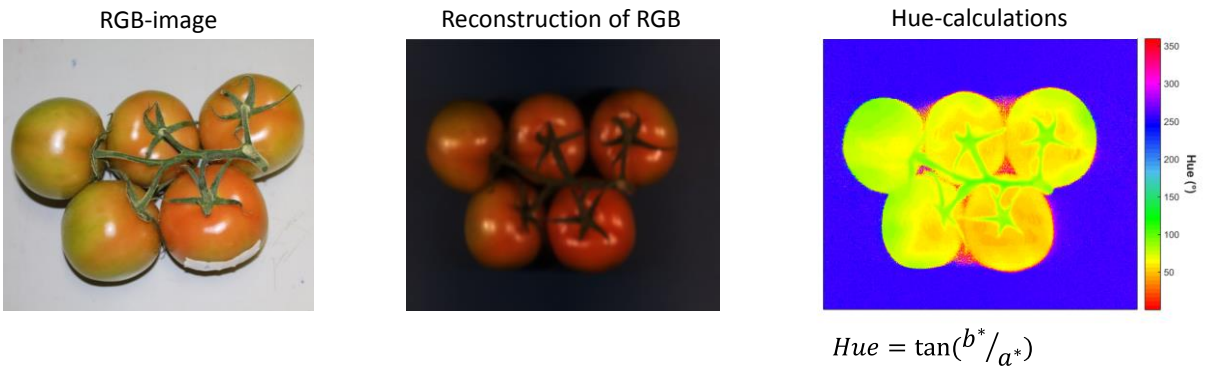


Figure 6. Hue prediction for tomatoes. The hue-value is calculated as the tangent of the b- and a-value of the Lab-colour space.

Finally, figure 7 shows an example of damage detection in tomato trusses. These results were obtained offline, but can be implemented online according to the same methodology used for the ripeness and brix-values.

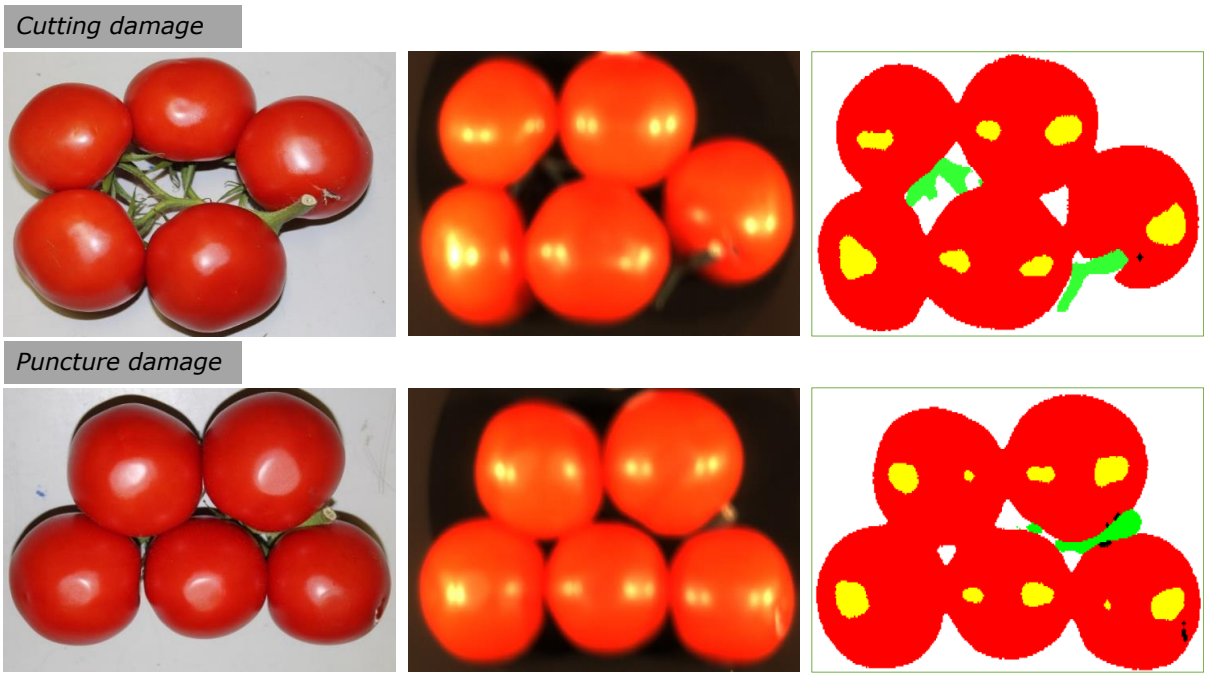


Figure 7. Determination of cutting and puncture damage in tomatoes. The left images show an RGB image of a truss. The middle images are a fake colour reconstruction of the trusses. The right side images show the results obtained (red= tomato flesh, yellow= gloss, greed = stalk, black =damage).

Grapes

Similar to the methodology described above for tomatoes, a model for the **sugar content (brix)** of grapes was developed. In Figure 8 and Table 2, the results obtained are shown for white grapes. A good

correlation (r^2 = 0.95) between measured and estimated SSC (soluble solid content) was obtained. For red grapes (r^2 = 0.35, see Table 2), no good predictive model could be fit.

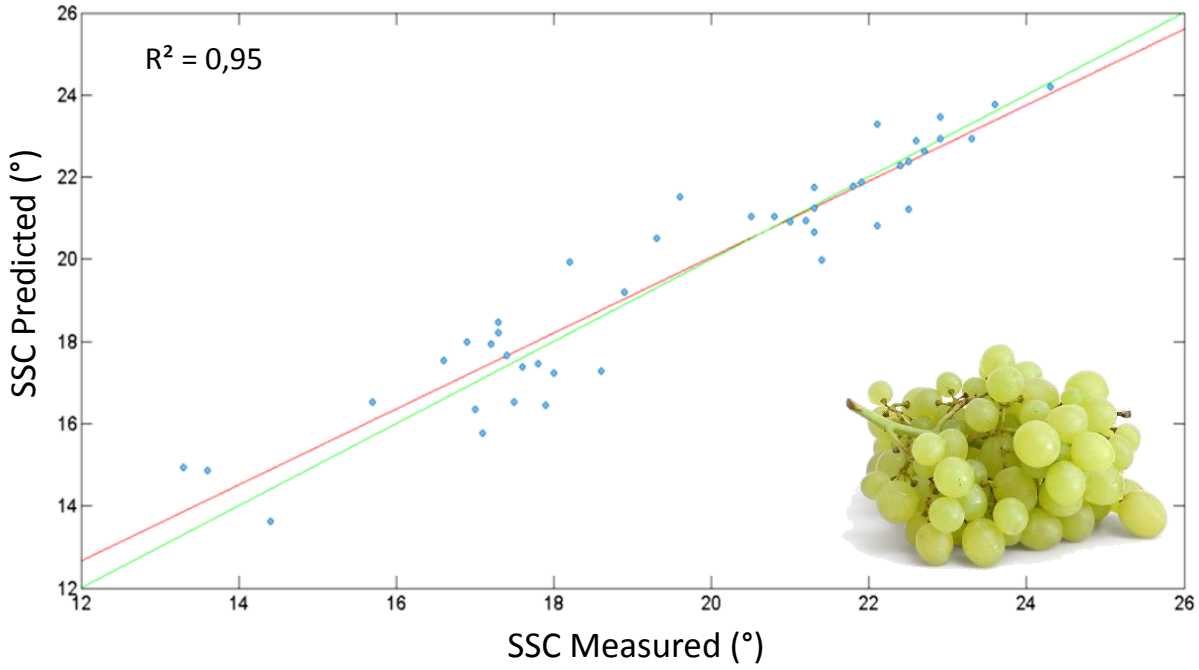


Figure 8. Measured versus predicted soluble solid content (SSC, Brix) for white grapes.

Table 2. Overview of the results obtained by the data models for determining grape brix values (SSC).

	R^2	R^2_p	RMSEC	RMSEP
SSC (white)	0,95	0,73	0,64	1,33
SSC (red)	0,35	0,31	1,36	1,72

Chicken and turkey

A classification model was built that could discriminate between **uncooked/cooked** turkey and chicken. The model was built by means of partial least squares discriminant analysis (PLSDA). Figure 9 and Table 3 present an overview of the results obtained. Overall, about 90 % of the objects can be assigned to the correct class.

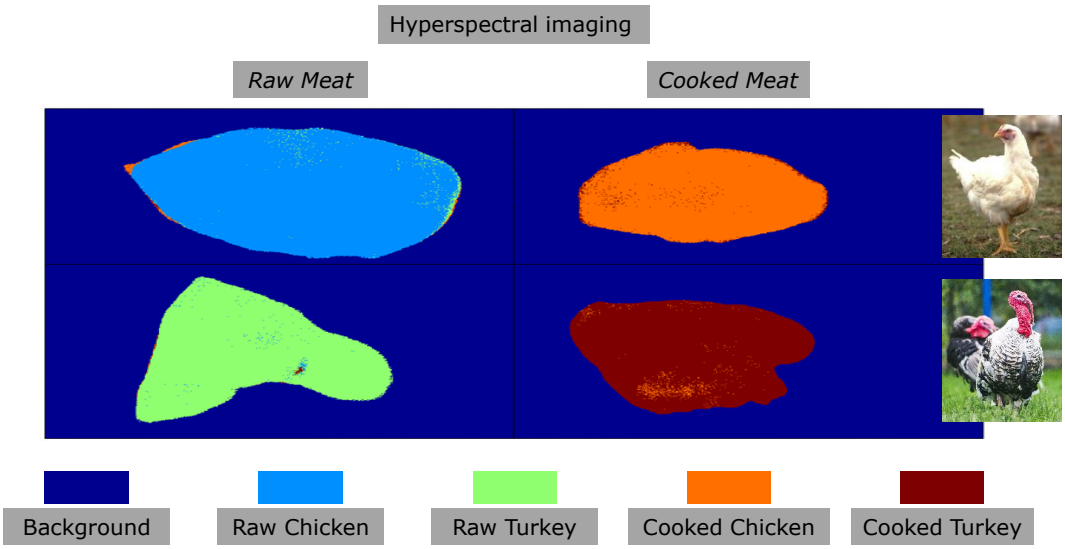


Figure 9. Example of the classification between raw or cooked chicken and turkey. Each pixel is assigned a colour to mark to which class it was appointed by the model.

Table 3. Summary of the results obtained for the classification model for cooked/raw chicken and turkey.

Ground truth	Raw Chicken	Cooked Chicken	Raw Turkey	Cooked Turkey
Raw Chicken	88	0	5	0
Cooked Chicken	0	93	0	9,2
Raw Turkey	12	0	95	0
Cooked Turkey	0	7	0	90,8

RGB sensor (WUR)

Tomato trusses

Based on a small number of examples, a segmentation method is trained, which segments the image into background, tomato and stalk/calyx based on colour information stored in Hue-Saturation histograms. As both stalk/calyx and tomatoes can be green, a further refinement of the stalk/calyx segmentation is done using shape information, looking for elongated structures in the image. Furthermore, individual tomato segmentation is achieved by detecting circular/elliptical shapes in the image. Segmentation of individual tomatoes and stalk works robustly. Some examples of data recorded on the line are shown in Figure 10.

Figure 10: Examples of segmentation of the raw image data to the separate packages (orange boxes), identification of the individual tomatoes (blue ellipses), and stalk/calyx (blue dotted regions).

Using the segmentation of the individual tomatoes, the **colour** of each tomato is determined by taking the median Hue value over all pixels of the tomato area. This median Hue value is then used to estimate the colour class of each tomato on a human-readable colour scale using a polynomial regression model. The regression model was trained and tested on ground-truth data acquired by two human experts that established the colour class of 214 tomatoes on a scale from 1-12.

Figure 11 shows the results of estimation of colour class per tomato on the vine. The estimations made by our method correlates well with the ground truth measurements, with R^2=0.95. The average prediction error was 0.49. The ground-truth measurements were done by two experts and the predictions are compared to the average of the two. It must be noted that the human judgement of colour class was not fully consistent over the two experts, with an average difference of 0.56 colour class. It can thus be noted that the prediction error is smaller than the disagreement between the experts. It is difficult to say which part of the prediction error is due to human error and which part to machine error.

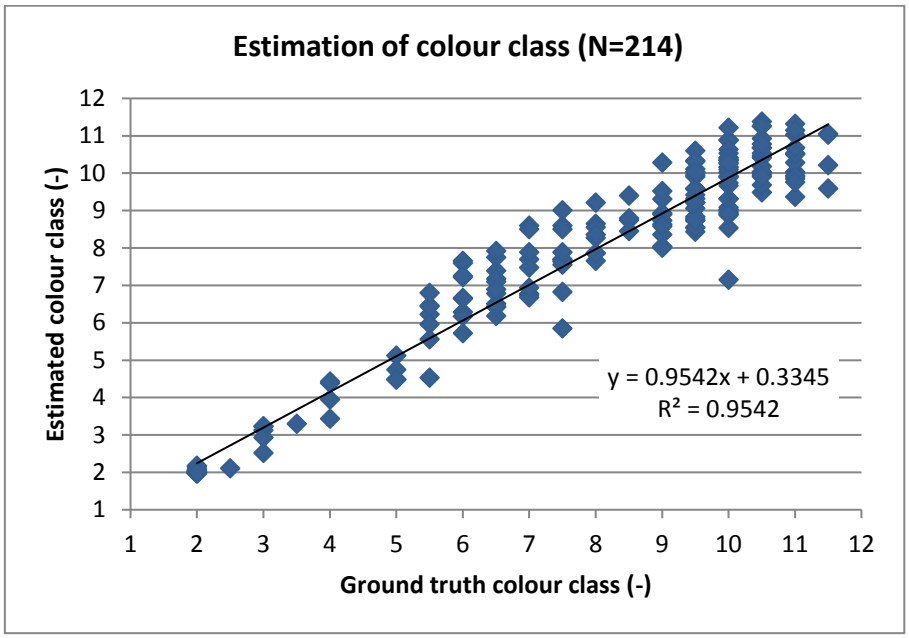


Figure 11: Estimation of colour class on a scale from 1-12. Average error is 0.49.

3D sensor (WUR)

Tomato trusses

An example of the data recorded by the 3D sensors can be seen in Figure 12. Using the transformation matrices obtained during the calibration of the sensors, the package ROIs are translated to the coordinates of the 3D sensor, which is used to get separate 3D point clouds for each package.

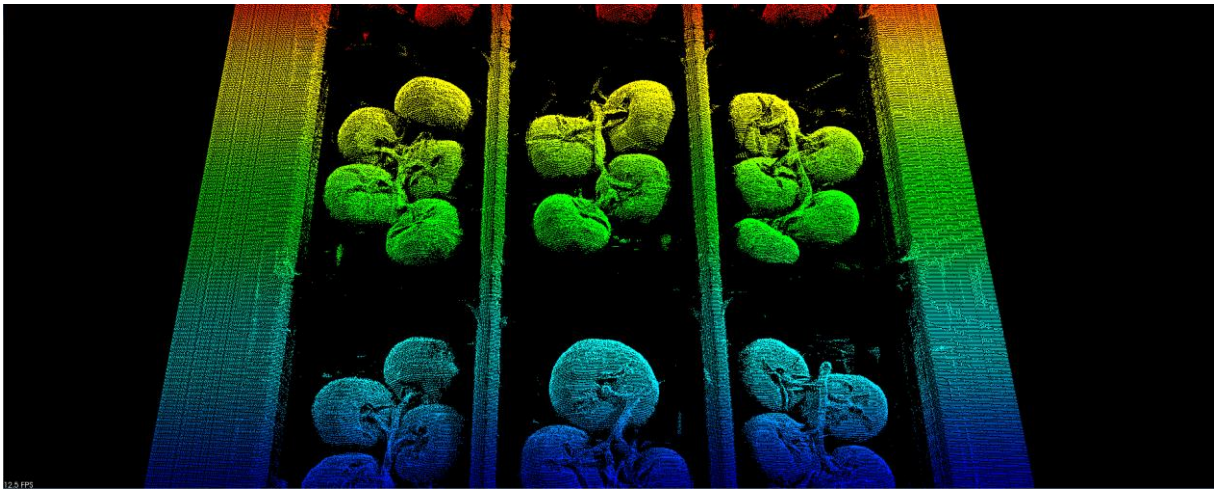


Figure 12: Example of 3D data recorded by the QAS module.

Based on the segmentation of the individual tomatoes, some parameters of the shape and size of the tomatoes are determined. Specifically, we take the length of the major and minor axes of the ellipse fitted to the contour of each tomato. Based on this information, a regression method is trained to predict the **weight**. The learning and testing is performed on ground-truth data, which is collected by measuring the weight of each tomato with a scale.

Figure 13 shows the results of weight estimation in the PicknPack line. The results show an average prediction error of 5.0 g, with a R^2 of 0.53. Results of an earlier off-line experiment with more elliptic tomatoes are shown in Figure 14, indicating an average prediction error of 3.9 g.

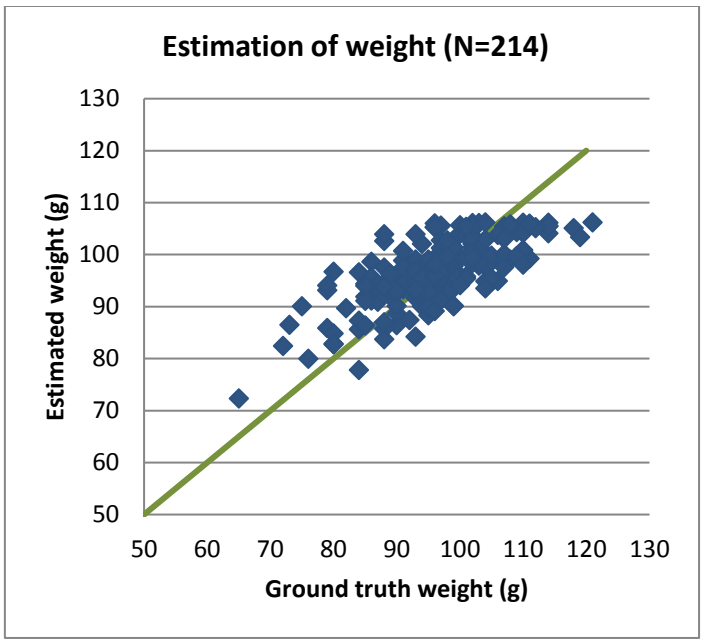


Figure 13: Estimation of weight of individual tomatoes. R^2=0.53. Average prediction error is 5.0 g.

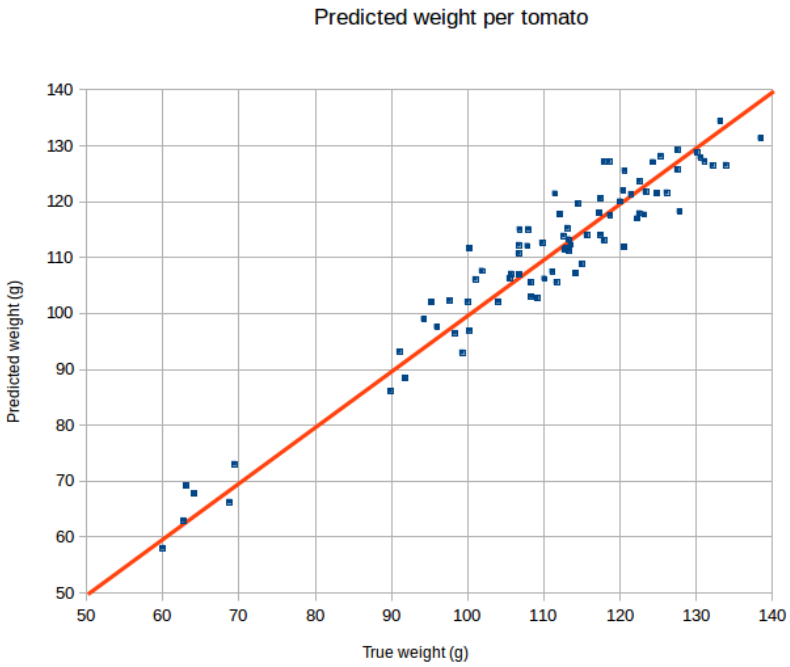


Figure 14: Weight estimation in an earlier off-line experiment, with an average prediction error of 3.9 gram.

Microwave scanner

The microwave sensor scans food products to obtain its dielectric properties and evaluate the food quality by means of the associated Equivalent Dielectric Value (EDV). Both the mean value and maximum EDV values are calculated for analysis and evaluation.

The microwave sensor can discriminate between food samples which differ in moisture content, which can for instance be used to evaluate the freshness and predict the product shelf life.

Chicken

The microwave was used to determine the cooking stages of chicken filet. In Figure 15, chicken breasts of different cooking stages are shown, being: raw chicken and chicken cooked with a microwave oven for 5, 10, 15, 20, 25, 30, 35, and 40 minutes. In these samples, sample 5 (cooked for 20 minutes) is properly cooked, sample 1 is uncooked (raw), 2-3 are insufficiently cooked, and 6 to 8 are over cooked.

Sample 1 Raw Chicken breast

Sample 2 Cooked 5 minutes

Sample 3 Cooked 10 minutes

Sample 4 Cooked 15 minutes

Sample 5 Cooked 20 minutes

Sample 6 Cooked 25 minutes

Sample 7 Cooked 30 minutes

Sample 8 Cooked 35 minutes

Sample 9 Cooked 40 minutes

Figure 15. Chicken breast samples of different cooking stages

The test results of the samples in Figure 15 are given in Figure 16.

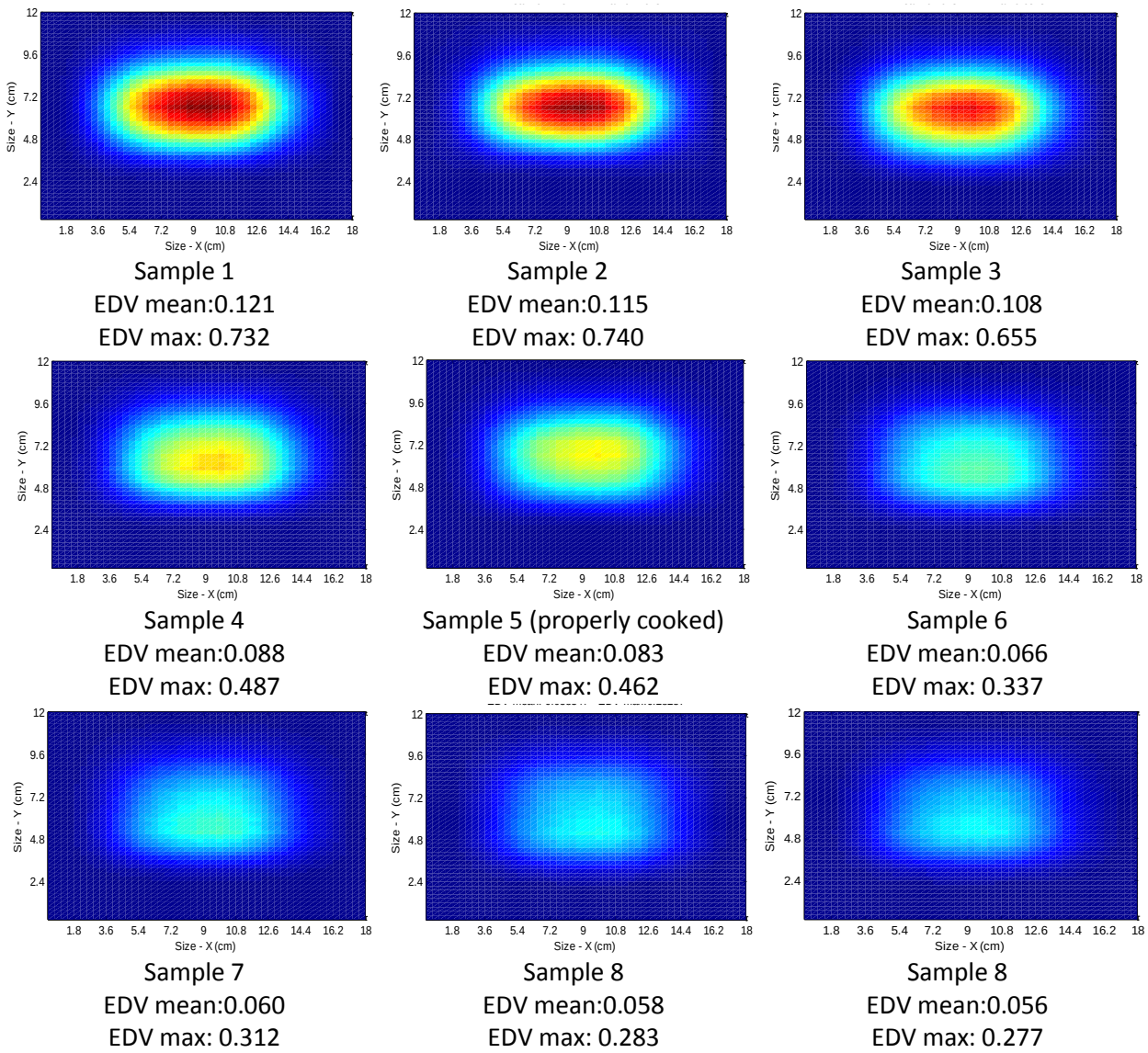


Figure 16. EDV measurement of chicken breast with microwave sensor

The trends of EDV mean and EDV max with cooking time are given in Figure 17, and Figure 18.

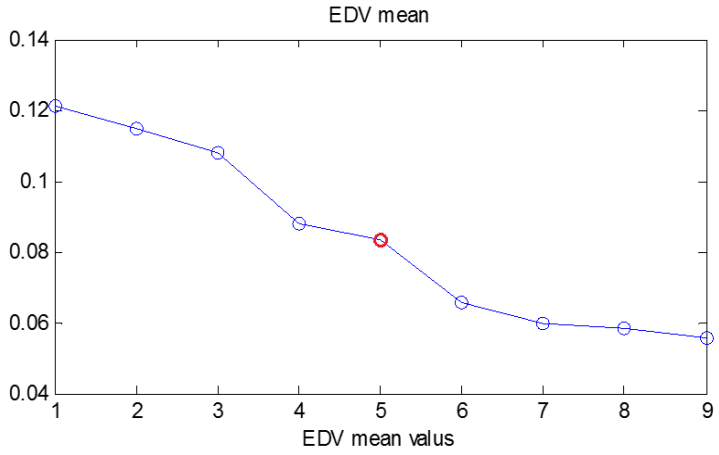


Figure 17. Trend of EDV mean values of chicken breast

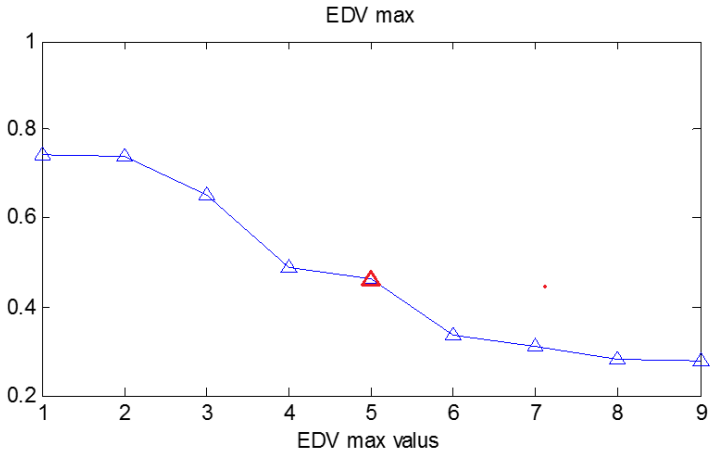


Figure 18. Trend of EDV maximum values of chicken breast

From Figure 17 and Figure 18, it can be seen that there is a good agreement in EDV mean and EDV max with the cooking time. The developed model can thus be used to evaluate the degree of cooking of chicken breast. The same methodology is expected to be applicable for other food products.

X-ray sensor (KUL and InnoS)

Tomato trusses

Considerable effort has been spent to allow accurate imaging in a stop-and-creep environment as implemented in the PicknPack system. The sensors were appropriately triggered by an encoder in direct contact with the web of trays. The resulting images needed various types of calibration and normalization, as shown in figure 19(a). The wavy patterns in the vertical direction are caused by small variations in exposure due to the stop and creep motion. The horizontal binary bands are the result of the detectors being overexposed due to hardware constraints at very low speeds of the web of trays just before and after the web stops and starts respectively. The vertical black bars that are visible in the images are caused by the web of tray supports passing between source and detector and were simply cropped out. Figure 19(b) shows the corrected X-ray images. The wavy patters have

disappeared completely but some image degradation is still visible in the regions of low web speed. The cause has been identified and future detector firmware upgrades should solve this issue.

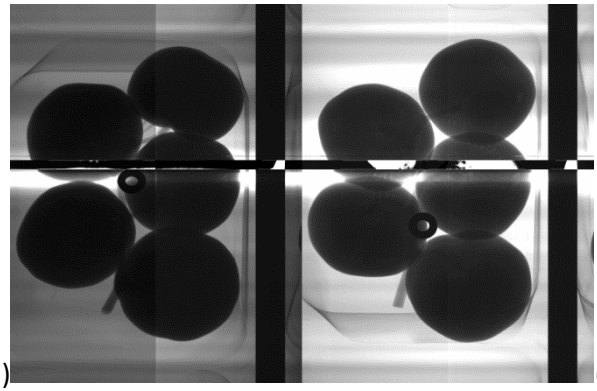
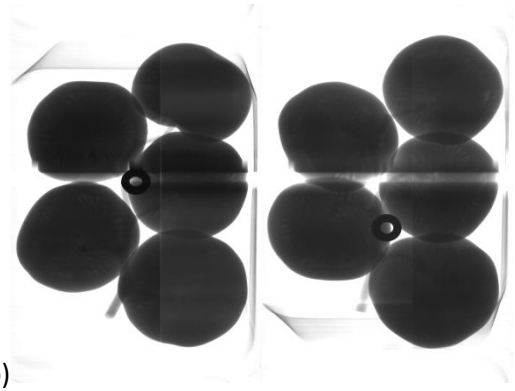
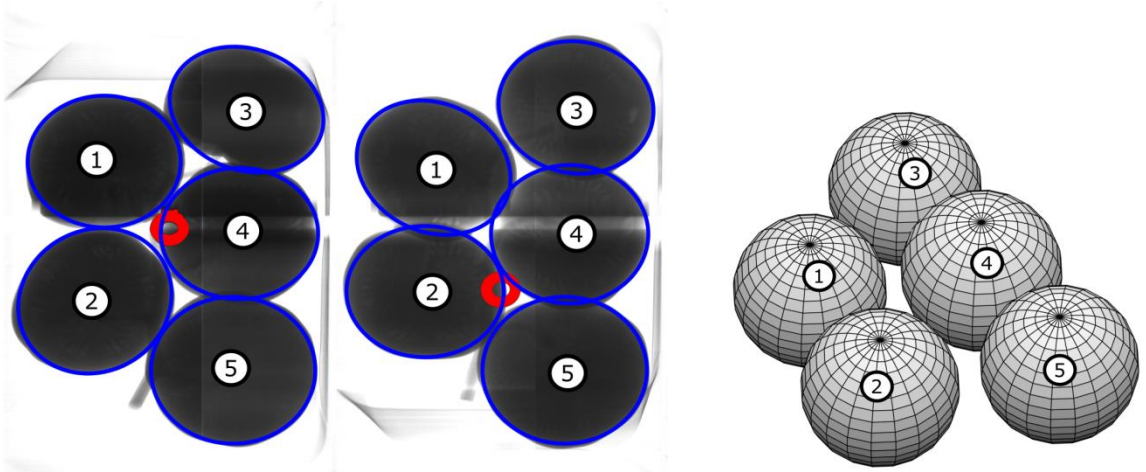
(a)   (b)

Figure 19: Uncorrected (a) and corrected (b) X-ray images.

The X-ray scanner is currently capable of detecting foreign bodies in trays filled with tomato trusses, grapes or chicken breast as shown in figure 20(a) in red. Based on the stereo-images, 3D position and size of individually detected objects (figure 20(a), blue) can be assessed. Note that size estimation does not encompass shape reconstruction.



(a) (b)

Figure 20: Detection of foreign bodies – in this case a metal washer - (in red, a) in trays filled with tomato trusses (detected contours in blue) and a 3D rendering of estimated tomato size and position (b).

Advanced algorithms for internal defect detection and shape reconstruction have been developed but still need integration with the 3D-vision system before online implementation is possible (also see Deliverable report D4.7). This integration relies on deploying the communication software as developed for the QAS module on the X-ray module which was delayed due to the postponed join date of InnospeXion in the PicknPack project.

Offline simulations for a system with a 0.5 mm, 1 mm and 2 mm detector pixel size (Figure 2) show that these methods can reliably detect small defects with limited variation in density as shown in Figure 21, left column. The right column shows classification results for manual inspection of radiographs.

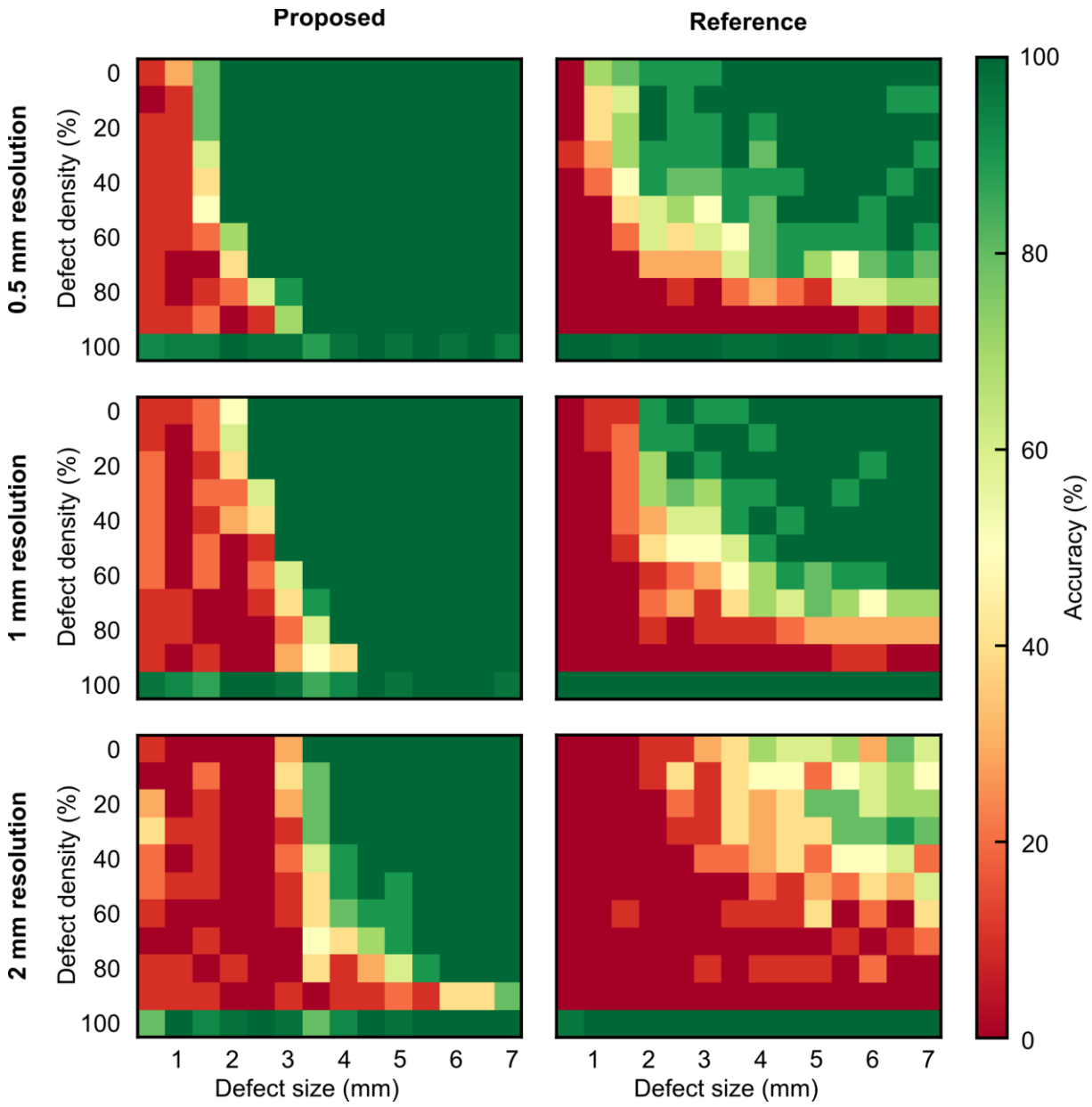


Figure 21: Detection accuracy (%) of defects for a detector with pixel sizes 0.5, 1 and 2 mm. Columns show the defect size in mm while rows show the defect density in relation to sample density (i.e. samples in the bottom row don't contain defects). As a reference, accuracy of visual classification of radiographs is shown on the right.

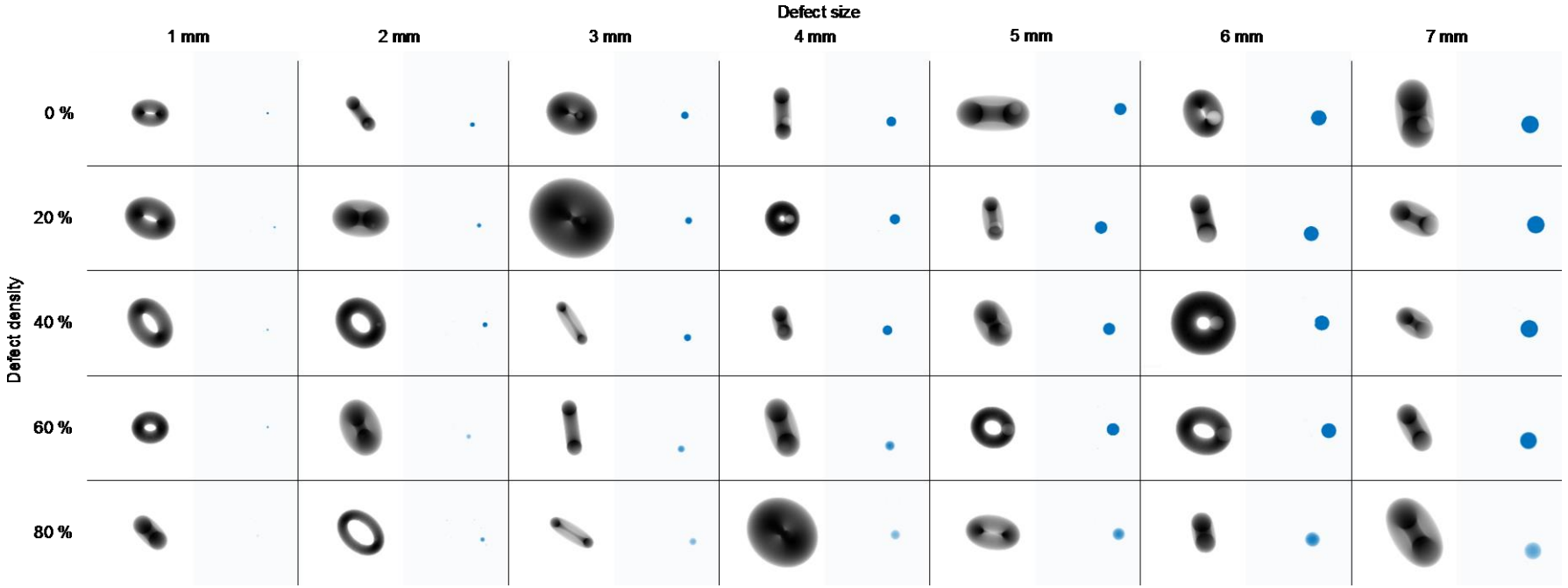


Figure 22: Radiograph and residual output of the in-silico simulations to evaluate multi-sensor inspection classification performance. The dataset consisted of randomly oriented tori of random size, with spherical defects with densities ranging from 0 to 100 % of sample density and radii of 0.5 up to 7 mm.

QAS module controller (KUL)

As described in the previous paragraphs, each of the sensor measures certain features per tray/punnet or per object and sends these to the module controller. The module controller can then be used to calculate new features based on the combination of the features received.

As an example of this approach, an algorithm was developed and implemented online that could assign products in a tray to a user specified quality class. An object is assigned to a certain quality class (in this example “premium quality”, “acceptable quality” and “not acceptable quality”) by checking for a number of pre-defined product properties if they fall within a desired range (selectable by the user of the system).

In Figure 23 an example is shown for tomato trusses. The brix value (HSI), colour (RGB) and weight distribution within the truss (3D) are measured by different sensor systems and sent to the QAS module controller. For each quality feature a desired range is defined (green = desired, orange = acceptable, red = not acceptable). If a product scores in the ‘green zone’ for all features, it is considered premium quality. If it score in the ‘red zone’ for a single feature, it is considered not acceptable. In the other scenarios (combination of green and orange ‘zones’) the product is considered of acceptable quality.

Figure 23. Screenshot of (a part of) the user interface to define quality classes for tomato trusses. The colours on the sliders indicate the desired values for each feature (green: premium quality, orange, acceptable, red: not acceptable).

The methodology described here is flexible, in that any of the features that are measured can be selected as part of the combination of features which are used for quality grading. Likewise, the ranges that determine each 'zone' on the quality scale (green/orange/red) can be quickly changed in function of the current needs of the production line.

2.2.2 The QAS module as a flexible food inspection system

Quality measurements as added value for consumers

The quality information gathered online in the QAS module about every product on the line is sent to the line controller. The line controller forwards this information to the printer module, which prints a QR code, including a link with which the consumer has access to all available quality information (see Figure 24).

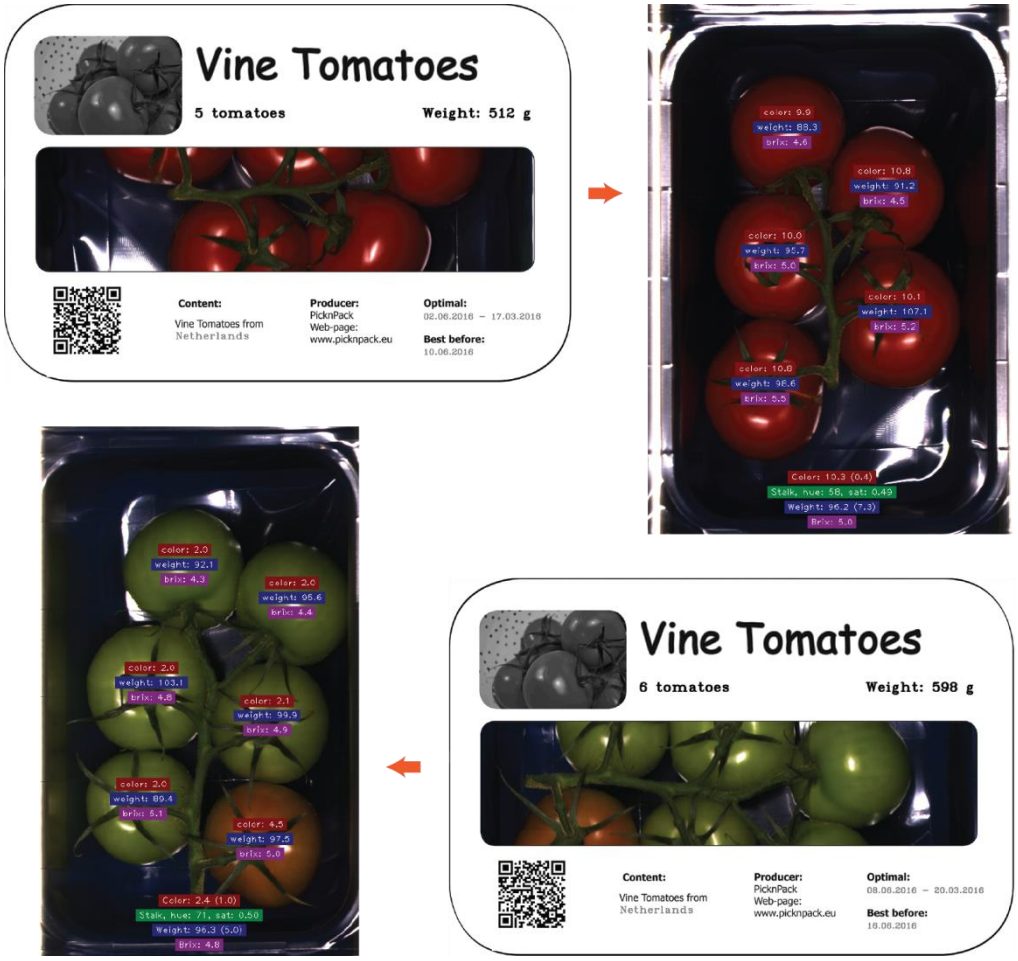


Figure 24. Examples of prints produced by the PicknPack line including a QR code, which contains a link to an image containing more information about the product as measured by the QAS module.

Semi-supervised learning (KUL)

To assist an operator at building new classification models for predicting (new) quality features for a (new) type of product, a semi-supervised segmentation algorithm was developed. This combines the discriminating power and ease of use of unsupervised segmentation algorithms, like k-means clustering, with the calculation speed of supervised classification algorithms, like PLSDA. In a first step,

an operator should decide on which algorithm gets the best unsupervised segmentation result. Based on this input, the algorithm will build a supervised model to classify the pixels. Then the operator has to decide on 10 new images if the segmentation, resulting from the supervised classification, of these images is correct. Otherwise, the training set should be extended using an unsupervised segmentation algorithm, like in the first step. This leads to an iterative procedure to build a robust classification model. In Figure 25, an example of this methodology is shown for tomatoes and grapes.

This algorithm was not yet implemented on the line but demonstrated stand alone. More detailed information can be found in deliverable report D4.8.

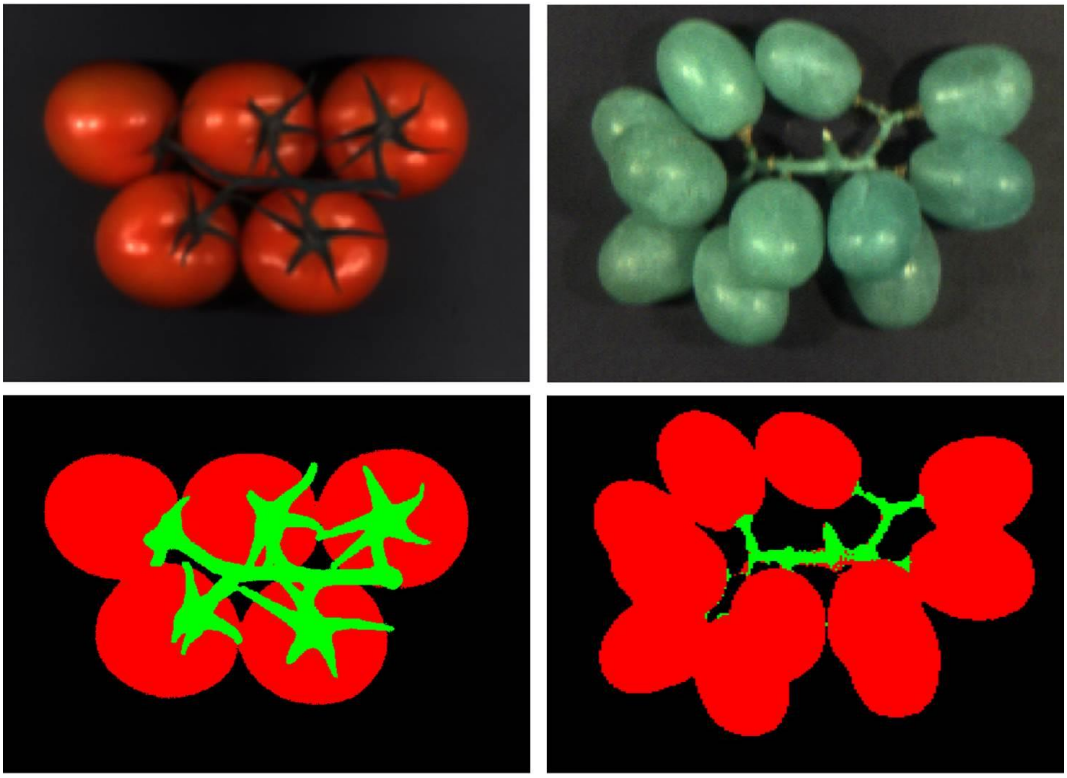


Figure 25. Example of the results obtained for the semi-supervised segmentation of tomatoes (left) and grapes (right). The top figures show a fake colour image of the measured products. The bottom figures show the results of the segmentation. The percentages represent the number of correctly classified pixels.

Deep learning (WUR)

Recently, methods inspired by the way in which the human brain's networks of neuronal cells process visual stimuli have led to great progress in the field of computer vision. State-of-the-art methods now attain remarkable performance in complex, general tasks such as identifying and tracking cars or people within images or videos.

Artificial neural networks accept images captured using, for example, hyperspectral cameras, and process the information in these images by passing them through layers of processing units of artificial

neurons. These neurons are modelled on the brain's neuronal cells, and each performs a simple computation. However, by combining many layers of these neurons, complex computations can be performed. By adjusting the connections between layers, a neural network can be adapted to perform a multitude of tasks, using the same general principle.

This technique was applied in the PicknPack Quality Assessment Module, where it employs neural networks to perform tasks such as identifying cuts or puncture damage in tomatoes. The process starts with a user annotating a set of training images by pointing out areas of the image containing the kind of imperfection that needs to be identified. In a process known as Deep Learning, the neural network is presented with this series of training images and user identified imperfections, and uses these to learn by example. After training, the neural network can be used in a production line to identify similar imperfections in images of produce passing through the quality assessment module.

It is important to note that the process of training the neural networks can be performed by non-experts. Users merely have to collect a representative set of training images, and point to defects in these images. This allows the deep-learning module to be adapted to a variety of produce types and kinds of imperfection. For example, a company may collect training images of chicken fillets, and identify areas of unacceptable discolouration on their surfaces. The QAS module will then learn, based on this training data, how to perform this novel task in an automated fashion.

The QAS module has been tested on a variety of tasks. These have included the identification of worm damage and bruises in colour images of apples, and cuts and puncture damage in hyperspectral images of tomatoes. The module has also been tested for potential use as part of the pick-and-place robot, where the system first identifies the stalks of blue, red or green grapes. In each case, however, the training process was identical. Images of either apples, tomatoes or grapes were captured, and areas of interest (for example, worm damage, cuts or stalks) highlighted by a user. Examples of the results are in Figure 16.

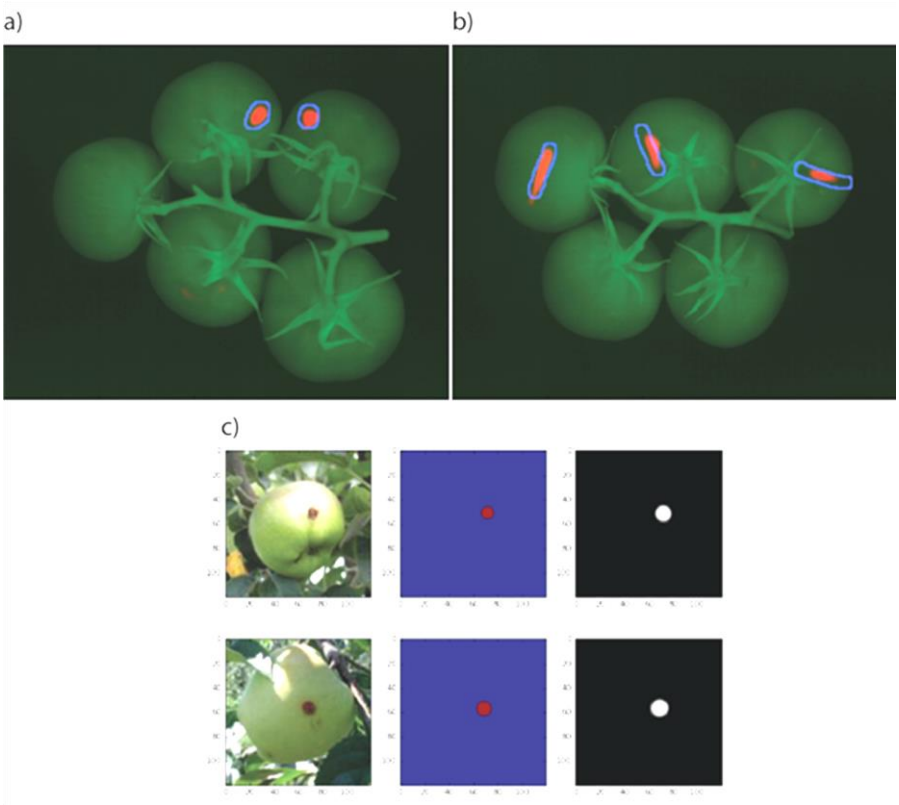


Figure 16: Results from defect detection based on deep learning. a) and b) show examples of using the network with hyperspectral data. The blue contours indicate the user-labelled ground truth and the red blobs are the result of the network. a) shows detection of punctures, b) of cuts. c) gives examples of the detection of worm holes in apples with the input image on the right, ground truth in the middle and the result of the deep network on the right.

Easy changeable configuration

The QAS module was designed to allow for an easy and flexible change of the configuration it is used in. This effect plays on different levels:

- The **QAS module** as a whole can be easily relocated to another position in the line. The software of the module was designed to automatically cope with this change with minimal user input.
- The **sensor configuration** can be easily changed. Sensors can be removed or added or their relative configuration can be changed as desired. As the devices that control the sensors use the same software design as that of the module, this reconfiguration is similar to the reconfiguration of the module described above.
- Sensors can be quickly assigned **new capabilities** in terms of models that are available for processing new features/product types. The processing chain is designed that it can assign any of the available models to a tray/punnet and use that model to calculate the desired features for that model. Models can be easily added or removed without changing the software of the processing chain itself. New features or product types that are added are handled automatically according to the standards described in the PicknPack protocol.