Monitoring mini-tomatoes growth: A non-destructive machine vision-based alternative

Fernando Ferreira Abreu, Luiz Henrique Antunes Rodrigues

School of Agricultural Engineering, University of Campinas, Campinas, Brazil

Abstract

Yield is the most often used metric of crop performance, and it can be defined as the ratio between production, expressed as a function of mass or volume, and the cultivated area. Estimating fruit's volume often relies on manual measurements, and the procedure precision can change from one person to another. Measuring fruits' mass will also destroy the samples; consequently, the variation will be measured with different samples. Monitoring fruit's growth is either based on destructive tests, limited by human labour, or too expensive to be scaled. In this work, we showed that the cluster visible area could be used to describe the growth of mini tomatoes in a greenhouse using image processing in a natural environment with a complex background. The proposed method is based on deep learning algorithms and allows continuous monitoring with no contact with the cluster. The images are collected and delivered from the greenhouse using low-cost equipment with minimal parameterisation. Our results demonstrate that the cluster visible area accumulation is highly correlated (R^2=0.97) with growth described by a parameterised Gompertz curve, which is a well-known growth function. This work may also be a starting point for alternative growth monitoring methods based on image segmentation. The proposed U-Net architecture, the discussion about its architecture, and the challenges of the natural environment may be used for other tasks in the agricultural context.

Correspondence: Luiz Henrique Antunes Rodrigues, School of Agricultural Engineering, University of Campinas, Av. Cândido Rondon 501, Barão Geraldo 13083-875, Campinas, Brazil.
E-mail: lique@unicamp.br

Key words: Deep learning; decision support system; image processing; precision agriculture.

Acknowledgements: this study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.* We thank Ms. Monique P. G. de Oliveira for the text revision and for sharing the infrastructure of her PhD Project granted by *FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), Process Nº 2018/12050-6.*

Received for publication: 27 January 2022.
Accepted for publication: 14 May 2022.

Journal of Agricultural Engineering 2022; LIII:1366
doi:10.4081/jae.2022.1366

Introduction

Mass and volume variation as a function of time usually defines fruit growth (Oswell *et al.*, 2018), and determining them requires measurements at frequent intervals. Obtaining fruits' diameters manually using measuring tapes, sizing rings, or callipers may be a way of estimating their volumes; however, the precision in the procedure can change from one person to another, and samples will either be limited by human labour or too expensive. Masses can be obtained by weighing fresh or dry fruits on a digital scale (Hall *et al*., 2013), but obtaining the dry mass will destroy the samples, and as a consequence, mass variation will be measured with different fruits. Fruit mass can also be estimated using previously obtained statistical models that relate other characteristics measured with mass determined by destructive tests. After fitting the model, mass can then be obtained indirectly with non-destructive approaches. For example, Tabatabaeefar and Rajabipour (2005) proposed estimating the mass of apples using the fruit's *projected area* and observed a correlation of 94%. Using the same methodology, Khoshnam *et al*. (2007) obtained 96.6% with pomegranates, Taheri-Garavand *et al*. (2011) of 94% with tomatoes, and Soltani *et al*. (2011) of 88.4% with bananas.

Although the projected area in these cases is carried out post-harvest in controlled environmental conditions; results suggest that it may be used to estimate mass in a natural non-controlled environment. However, this is a more challenging task: total or partial fruit occlusion, over or under lighting, and the low contrast between unripe fruits and leaves (Chen *et al*., 2017; Fukui *et al*., 2017) make it difficult to segment fruits' pixels from the background using only the human-perceived features, like colours or shapes. These difficulties could be overcome by machine learning (ML) and deep learning (DL) algorithms, and among them, the convolution neural networks (CNNs).

CNNs are a particular type of neural network designed for image processing that has been used for audio and text. As its name suggests, the *convolution* operation is the base of its behaviour, in which a 3D vector (called the *kernel*) extracts patches of features from an image (also a 3D vector) through a linear function (Chollet, 2018). CNNs' performance in image processing is related to two main characteristics: the hierarchical nature of its architecture allows it to detect more complex and abstract elements as layers go deeper; also, the sliding nature of the kernel makes detections translation-invariant, meaning that once they

learn a feature, it may appear anywhere in the image. These aspects of CNNs make them particularly useful in a natural environment scenario leading to accurate predictions, as reported in many works (Table 1). Like other DL algorithms, CNNs have evolved through time, resulting in multiple specialized architectures. U-Nets are one of them, described firstly by Ronnenberger *et al*. (2015) for biomedical image segmentation, where it showed good performance (Ali *et al*., 2020; Jha *et al*., 2020; Zafar *et al*., 2020; Zhou *et al*., 2020). U-Nets are composed of a *contracting path* in which feature information increases and an *expansive path* that concatenates up-convolutions with high-resolution features from the *contracting path.* For agricultural purposes, U-Nets also achieved high accuracy in predictions (Ngugi *et al.*, 2020; Bragagnolo *et al.*, 2021; Su *et al*., 2021), even with small datasets.

Tomatoes are a significant horticultural product, and DL tasks for this crop have been mainly focused on their detection (Liu *et al*., 2019; Afonso *et al*., 2020; Lawal, 2021), environment control, and yield prediction (Solanke and Kumar, 2013; Hemming *et al*., 2020; Johansen *et al*., 2020), so, there would be benefits from monitoring tomato growth non-destructively. Thus, this work proposed monitoring fruits' growth in a greenhouse by comparing their projected areas with a generalized sigmoid function, well-known for describing mass accumulation in tomatoes (Fayad *et al*., 2001; Heuvelink, 2005; Faurobert *et al*., 2007). Although the use of CNNs in agriculture is nothing new, most papers rely on fruit detecting, locating, or counting (Chen *et al*., 2017; Ganesh *et al*., 2019; Song *et al*., 2019; Santos *et al*., 2020; Wan and Goudos, 2020) and so, our main contribution is to use it as a measuring tool in an operational context with low-cost equipment. Besides that, as additional contributions, we made publicly available the dataset described in *Dataset construction* section (Abreu and Rodrigues, 2022) with 385 images and binary masks for semantic segmentation tasks. We also explore using daytime and night time images on models' training and the effects of the different light conditions on its precision and sensitivity.

Materials and methods

This section covers all materials and the experiment we conducted to collect the images used in modelling. In *Experiment overview* section, we describe the crop and the image collecting; in *Dataset construction* section, the image pre-processing and dataset construction; in *Proposed architecture* section, the symmetric U-NET architecture used for image segmentation; in *Training and validation* section, U-NET's training and validation, and at last, the correlation of the areas extracted from the images with the Gompertz function in *Gompertz curve fitting* section.

Table 1. Examples of convolution neural networks used in the fruit culture context.

Work	Task	Crop	Technique	Reported metrics
(Liu *et al*., 2019)	Detection	Cucumber	Faster R-CNN	Precision 0.858, Recall 0.836
			Original Mask R-CNN	Precision 0.887, Recall 0.863
			Improved Mask R-CNN	Precision 0.906, Recall 0.882
			YOLO V2	Precision 0.818, Recall 0.762
			YOLO V3	Precision 0.862, Recall 0.816
(Ganesh *et al.*, 2019)	Segmentation	Orange	Mask R-CNN	Precision 0.975, Recall 0.812
(Song *et al*., 2019)	Detection	Kiwi	Faster R-CNN + ZFNet	AP 0.725
			Faster R-CNN + VGG16	AP 0.876
(Pérez-Borrero *et al*., 2020)	Segmentation	Strawberry	RCNN	AP 0.438
(Wan and Goudos, 2020)	Detection	Apple, orange, mango	YOLO	AP 0.787 (Apple) AP 0.709 (Mango) AP 0.606 (Orange)
			Fast R-CNN	AP 0.787 (Apple) AP 0.798 (Mango) AP 0.765 (Orange)
			Faster R-CNN	AP 0.868 (Apple) AP 0.893 (Mango) AP 0.873 (Orange)
			YOLOv2	AP 0.906 (Apple) AP 0.881 (Mango) AP 0.876 (Orange)
			YOLOv3	AP 0.918 (Apple) AP 0.895 (Mango) AP 0.887 (Orange)
			Improved Faster R-CNN	AP 0.925 (Apple) AP 0.889 (Mango) AP 0.907 (Orange)

CNNs, convolution neural networks.

Experiment overview

We collected the required data from *Milla* mini-tomatoes plants in a greenhouse at the University of Campinas' School of Agriculture Engineering (FEAGRI-UNICAMP) between 18th July and 28th October, 2019. A commercial unit produced 72 seedlings transplanted within 45 days to 8-liter polyethylene pots, arranged 0.5 meters apart in four rows with 1.5 meters spacing. We randomly selected two for continuous monitoring of a cluster in each plant. For continuous monitoring, we used 2 Raspberry Pi Zero W mini computers with 1 GHz Broadcom BCM2835 processor and 512 MB of RAM, equipped with 5MP digital cameras and LED flashes, positioned 70 cm from the clusters. We programmed the computers to take a picture every 15 minutes, from 00:00 a.m. to 06:00 p.m., which resulted in 11,232 images.

Dataset construction

The pictures taken in the greenhouse cover all the fruits' maturation stages, from the flowers blooming to the cluster full ripening, with intentional redundancy to prevent data loss. Using all these images for modelling would increase autocorrelation and processing time. Furthermore, it would have little or no performance improvement since mini-tomatoes mass accumulation is barely noticeable with time intervals below 24 hours. So we selected four pictures a day, from the moment the first cluster's fruit appeared to its full ripening, at 03:00 a.m., 05:30 a.m., 12:00 p.m., and 4:30 p.m., to include different light conditions, resulting in 385 images divided into three sets as shown in Table 2. Pictures originally had 2592×1944 px dimensions and were clipped in an image editor to the smallest possible size that would fit an entire cluster in the frame. Final images then had 1024×1024 px, which reduced the number of pixels processed by the network and RAM consumption. For each image in the dataset, we built a *mask,* a binary representation of positive ('tomato') and negative ('non-tomato') classes in a homemade tool called *TommyGUI*, which created an ellipse from a set of dots drawn on the fruit edges (Figure 1). All visible fruits in the foreground truss were labelled from the moment they could be visually identified. In the case of a missing fruit in the truss, the absent fruit was represented with the smallest possible ellipse for later removal. An automated script loaded and transformed the images into 3D arrays, making it possible to change the size, colour encoding, and augment data in real-time.

Proposed architecture

Ronnenberger *et al*. (2015) originally proposed the U-Net for biomedical image segmentation. These images, like those produced by ultrasound, magnetic resonance, and tomography, have a complex background and diffuse edges; so the U-Net is naturally robust to overcoming them. Therefore, we proposed a modified version of this U-Net called TommyNET as the greenhouse images share some of the biomedical image features. Our architecture was built with the Keras framework with symmetrical contracting, expanding paths, and multi-scale residual blocks, as cited by Xie *et al*. (2017) (Figure 2). The symmetry was strategic to produce feature maps identical to inputs in size, 1024×1024 px.

Training and validation

We carried out TommyNET's training at Google Collaboratory PRO with a Tesla V100-SXM2-16GB GPU and 26 GB RAM. We chose Adadelta with standard parameters as its optimiser, 10-fold cross-validation, and Hubber as its loss function (Eq. 1):

$$Huber(x) = \{0.5 \cdot x^2 \; if |x| \leq 1 \; 0.5 + (|x| - 1) \; if |x| > 1\} \quad (1)$$

Table 2. Train, test, and validation split in the dataset by daytime.

Daytime	Train	Test	Validation
03:00 a.m.	65	17	9
05:30 a.m.	70	15	13
12:00 p.m.	70	22	7
04:30 p.m.	72	15	10

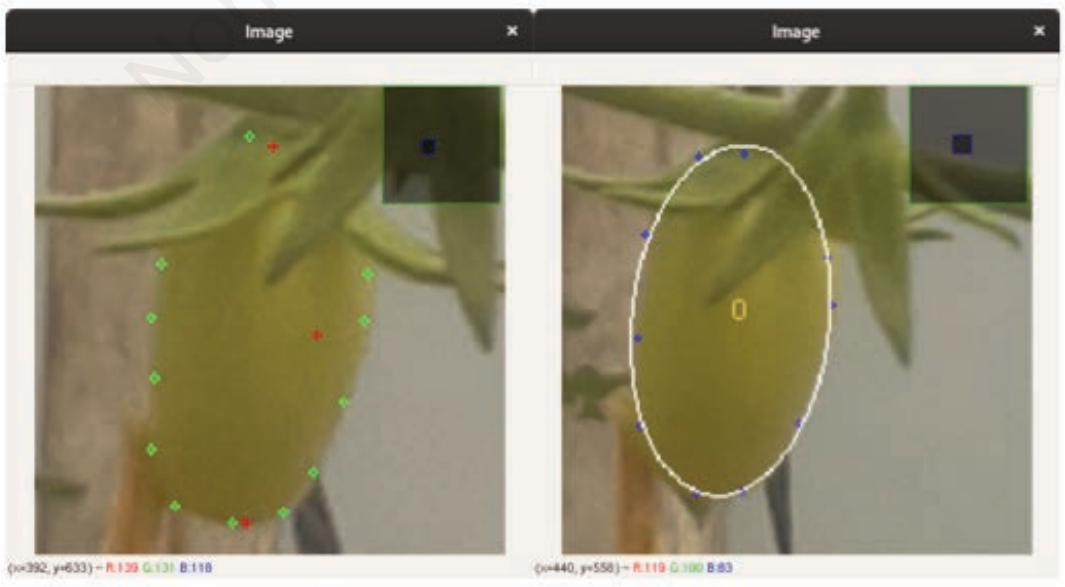

Figure 1. The dots drawn on the edges of the fruit (left) are used to adjust the ellipses (right). The red dots were misplaced and rejected.

We conducted three different training sessions using combinations of day and night images. We augmented the train and test sets with vertical and horizontal flips, rotations, and random noise by a variable augmentation factor that makes their final number of observations close to each other. Table 3 summarises the three sessions. Training sessions were interrupted when the loss did not decrease by at least 0.001 in 20 epochs.

We considered three main metrics for validating the predictions: precision, recall, and IoU. Precision may be interpreted as the accuracy in prediction when we consider all predictions that have been made for the positive class. The recall is similar but considers all true positive class examples in the dataset. Finally, Intersection over Union (IoU) is commonly used for image segmentation tasks and relies on sets theory, matching pixels in the prediction and in the mask. After training, the resulting predictions were thresholded to make it possible to use classification metrics, meaning every pixel greater than 160 was changed to 1 or 0. The threshold value was chosen to maximise the validation set IoU.

Gompertz curve fitting

In each dataset image, we summed the pixels corresponding to the 'tomato' class to obtain the corresponding area. The values were standardised using a z-score and normalised to fit the range [0, 1]. Dates were replaced by 'days after appearance'; the first date is defined as 0 and the following by the number of days counted after the fruit first appeared. We then adjusted the curve using the curve_fit function from the SciPy Python library (Virtanen *et al*., 2019), which uses a non-linear least-squares optimisation. The Gompertz function is described by three parameters, as seen in Eq. 2:

$$G(x) = ae^{-e^{b\cdot(x-c)}} \quad (2)$$

In this form, a is the superior asymptote meaning the maximum cluster size, b is the halfway point representing the speed at which they reach a, and c is the horizontal curve displacement determining the moment the exponential growth starts.

Results and discussion

This section covers the experiment results and the discussion that precedes the conclusions. In *Image and segmentation performance* section, we present models 1, 2, and 3 training conditions and validating results. In *Light influence* section, we will discuss the light influence on model training and how the well and poorly illuminated images contribute to model accuracy and sensitivity. Finally, in *Automatic feature detection* section, the light influence will be correlated with the kernel size and automatic feature detection.

Image segmentation performance

TommyNET's training has achieved the results summarised in Table 4. They achieve their best performance when models are validated using only the same light conditions used in training. The results show that the precision, recall, and IoU are nearly the same for all models in this scenario. Although expected, this information is useful as it may be treated as a performance benchmark.

Light influence

Since the U-Net mimics the natural vision mechanism, light plays an essential role in model performance. We may observe that the precision from the results in Table 4, is higher for Models 1 and 3, indicating that these models benefit from training with well-illuminated scenes. On the other hand, training with poorly illuminated images forces models to be more sensitive to light, which leads to a high average recall for Model 2, which also has the lowest average precision, likely caused by false positives proportionally to the light in the environment (Figure 3).

Table 3. Summary of TommyNET training sessions.

Model	Condition	Train images	Test images	Augmentation factor	Epochs	Train time
1	Day+Night	831	207	2	138	9:00:48
2	Night	825	203	11.70	151	9:45:01
3	Day	830	206	4.89	155	10:05:24

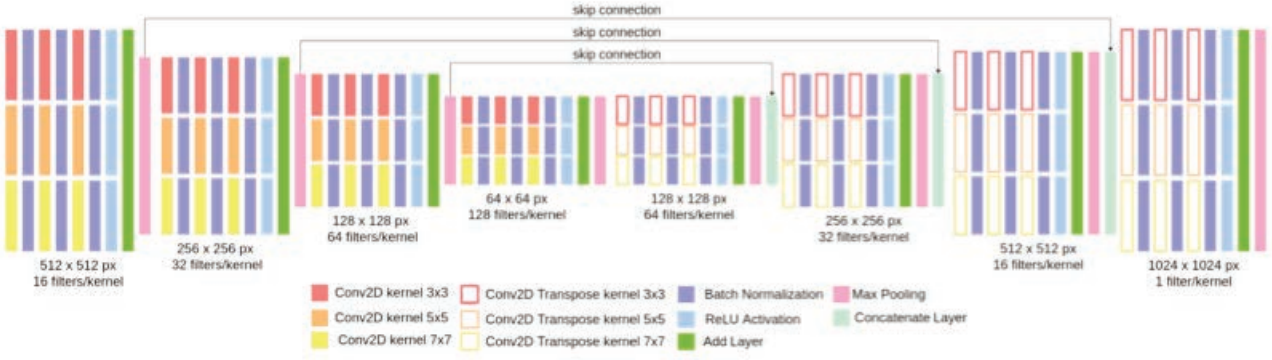

Figure 2. ***TommyNET*** **architecture.**

Automatic feature detection

As deep learning techniques rely on automatic feature detection, features obtained often make little or no sense to humans. However, we may explore intermediate results to study which parts contribute more to prediction.

From the images we sampled to analyse, in all models' intermediate predictions, we observe abstraction levels increasing with network depth. The first convolution block produces an activation map in which we may still notice the cluster silhouette and some light effects such as shadows and highlights, which may be the result of this convolution block using low-level features, such as colours. The second and third convolution blocks' segmentation maps, especially kernel sizes 3×3 and 5×5, appear to edge detectors, reinforcing the fruit silhouette. Finally, the last convolution block is nearly abstract, and its resolution does not allow us to make any interpretations. This behaviour may be easily seen in the examples for Model 1 (Figure 4) but is also noticeable for Model 2 (Figure 5) and Model 3 (Figure 6) when using their respective validation datasets.

The kernel size is related to model performance. Smaller kernels tend to use shallow features compared to fewer neighbouring pixels. They also allow deeper networks since the activation maps are smaller on each layer, but very small kernels cause instability in the model's training (Agrawal and Mittal, 2020). On the other hand, larger kernels are related to performance decreasing (Chen

Table 4. Training session results. The results were evaluated in validation images described in Table 2.

Model	Train set condition	Validation set condition	Precision	Recall	IoU
1	Day+Night	Day+Night	0.97	0.97	0.94
		Day	0.87	0.86	0.83
		Night	0.96	0.97	0.93
		Average	0.93	0.93	0.90
2	Night	Day+Night	0.42	0.96	0.41
		Day	0.18	0.97	0.17
		Night	0.98	0.97	0.95
		Average	0.53	0.97	0.51
3	Day	Day+Night	0.95	0.81	0.80
		Day	0.99	0.95	0.94
		Night	0.83	0.34	0.33
		Average	0.92	0.70	0.69

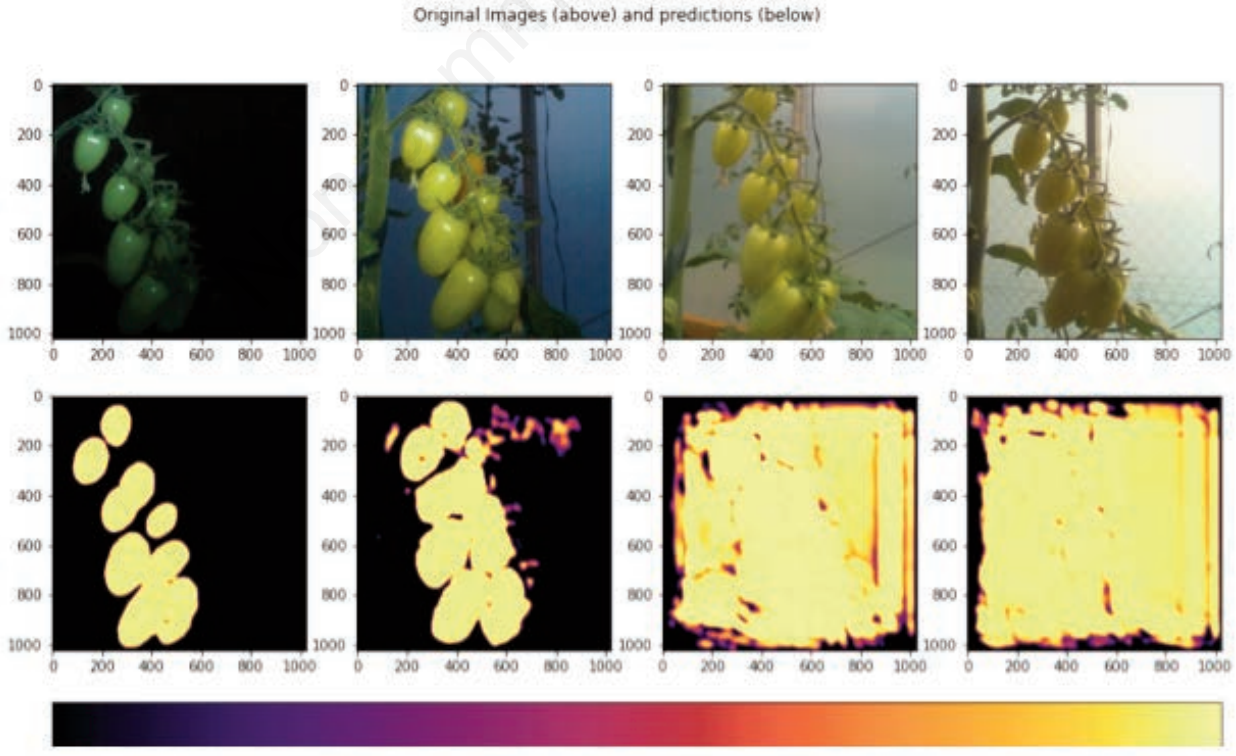

Figure 3. Example of Model 2's predictions (second line) for the same day images (first line) in different light conditions. Darker colours for predictions mean lower pixel values, *i.e.*, lower probability of the positive tomato class.

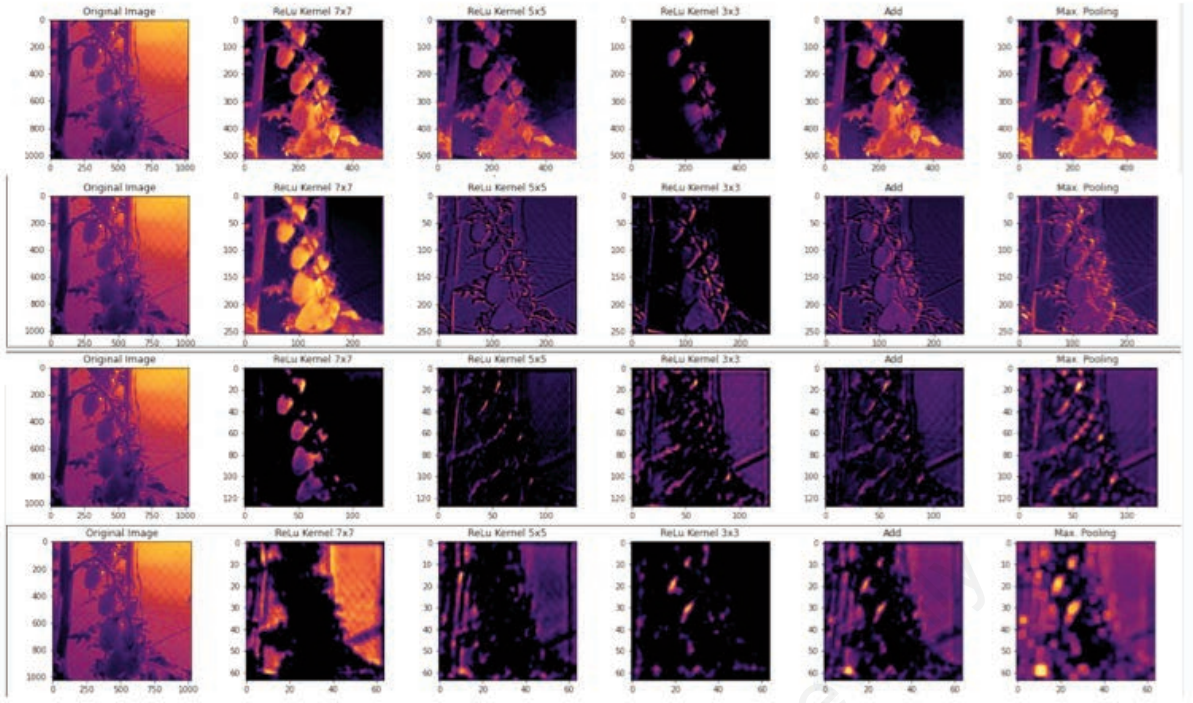

Figure 4. One random filter of each size chosen from Model 1's first convolution block (top line) to the last (bottom line). The first column is the original image, and the next three are the activation maps produced by different-sized kernels, followed by a sum (add) and a max-pooling layer, which give the result for this filter. Darker colours mean lower pixel values.

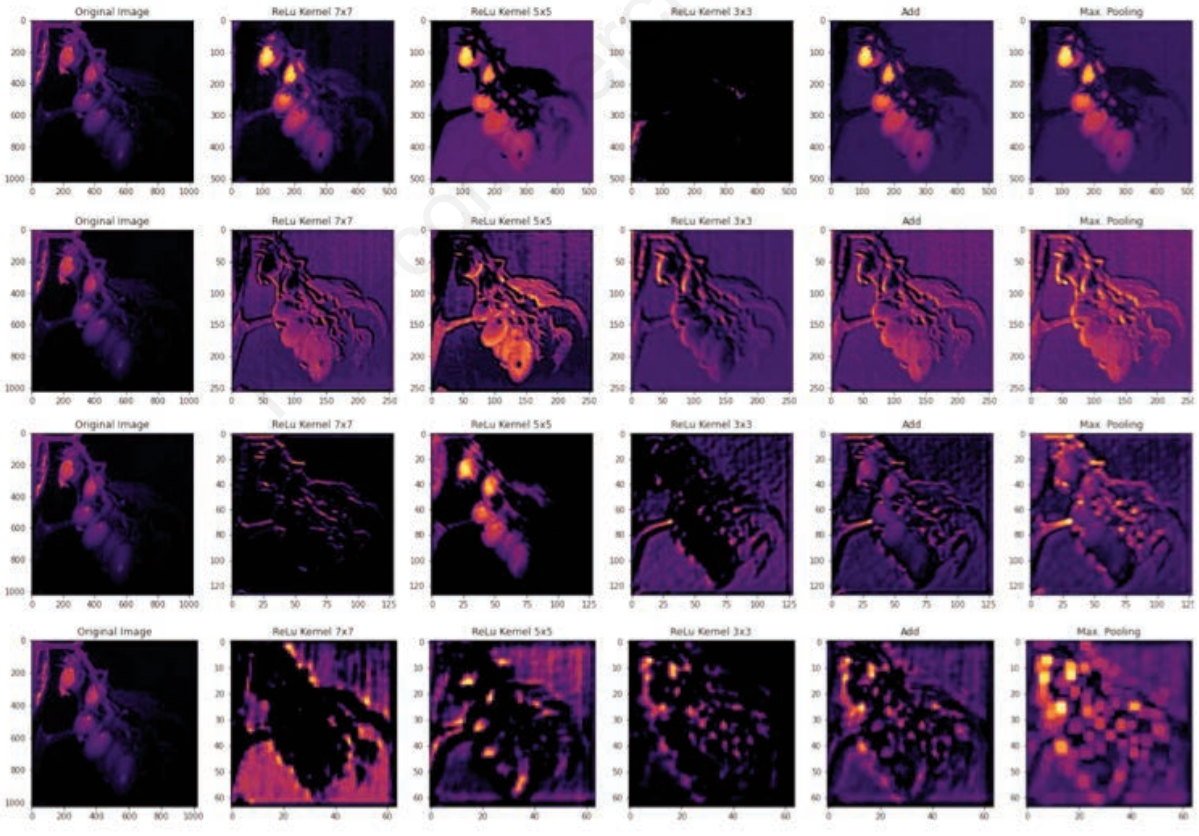

Figure 5. One random filter of each size chosen from Model 2's first convolution block (top line) to the last (bottom line). The first column is the original image, and the next three are the activation maps produced by different-sized kernels, followed by a sum (add) and a max-pooling layer, which give the result for this filter. Darker colours mean lower pixel values.

and Shen, 2017; Ozturk *et al*., 2018) and are more common in shallow networks since the activation maps are considerably smaller in each layer.

In this work, the kernel size appears to be related to the training condition of each model, and their behaviour is similar to the one of a digital camera that needs to open its diaphragm to receive more light for a poorly illuminated scene. Kernels 7×7 and 5×5 (Figure 5) on the first convolution block of Model 2 have higher pixel values, *i.e*., pixels with a value close to the maximum of 255 on images and masks, which represent the positive class, in a small, concentrated area on the cluster's top-left section, which is better illuminated at the moment. The same area and the cluster were mainly neglected by kernels 3×3, indicating that this size was insufficient to retrieve the scene information for this image. We may observe a similar situation in Model 3's predictions. Still, in this case, the absence of darker images in training leads to the most insensitive model, like a digital camera with a low ISO sensitivity. Consequently, kernel sizes 5×5 and 3×3, representing the low apertures, detect almost nothing from the original image. This suggests that using different size kernels in the first convolution block is helpful to overcome environment light conditions.

Convolution blocks two, three, and four appear to compensate for this difference, and the last one is very similar in all models, indicating that higher-level features were detected and used for prediction at that point. This compensation leads to image reconstruction results appearing more understandable and consistent among models.

Figures 7, 8, and 9 exemplify deconvolution blocks. As we can see for Models 1 (Figure 7), 2 (Figure 8), and 3 (Figure 9), the first deconvolution block shows the cluster's silhouette, but the background is also very noticeable. The difference between them is reinforced by blocks two and three, where some spots and stains remain apparent and reach their maximum in the last block, where the foreground cluster is entirely segmented.

Gompertz curve fitting

We used train and test sets to fit the Gompertz function since it does not require hyperparameter tuning. The optimal curve (Figure 10) was validated with validation set images and showed a high correlation with data, with a symmetric mean absolute percentage error (sMAPE) of 10.1% and an R^2 of 0.97. Curve parameters are a=0.846; b= –0.175; and c=10.501.

In Figure 10, it is possible to see the different growth stages, particularly the rapid growth moment, between days 5 and 25 and the slow growth moment, after day 25, with durations very close to the ones described by Heuvelink (2005) and Faurobert (2007). It is also possible to notice a discontinuity caused by camera displace-

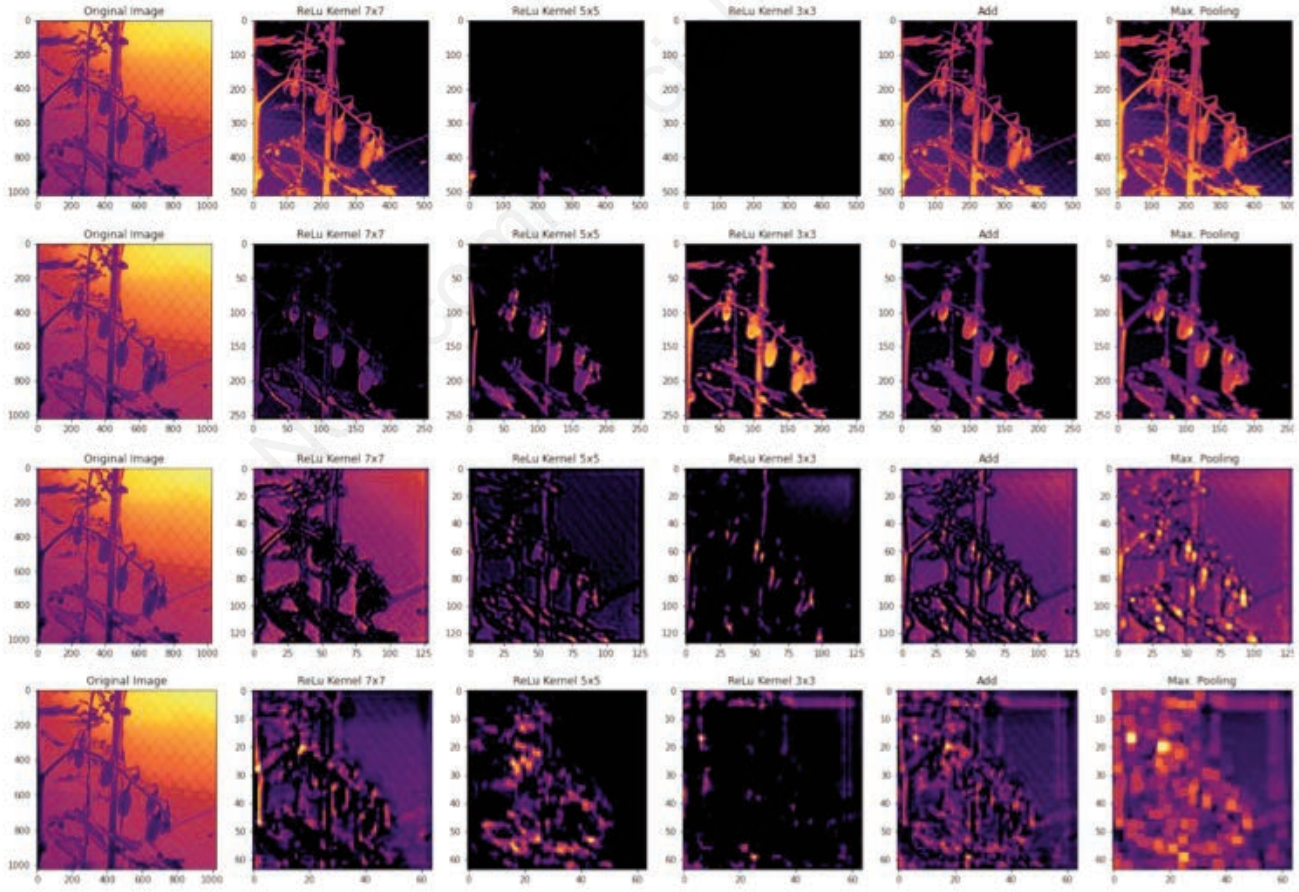

Figure 6. One random filter of each size chosen from Model 3's first convolution block (top line) to the last (bottom line). The first column is the original image, and the next three are the activation maps produced by different-sized kernels, followed by a sum (add) and a max-pooling layer, which give us the result for this filter. Darker colours mean lower pixel values.

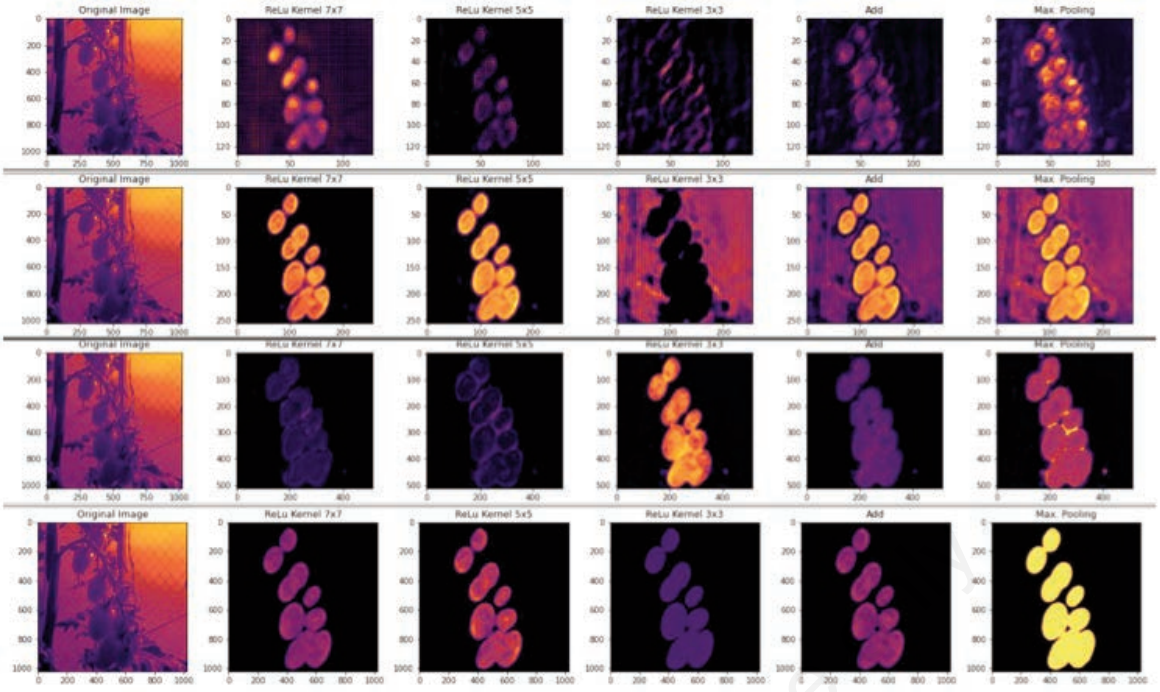

Figure 7. One random filter of each size chosen from Model 1's first deconvolution block (top line) to the last (bottom line). The first column is the original image, and the next three are the activation maps produced by different-sized kernels, followed by a sum (add) and a max-pooling layer, which give us the result for this filter. Darker colours mean lower pixel values.

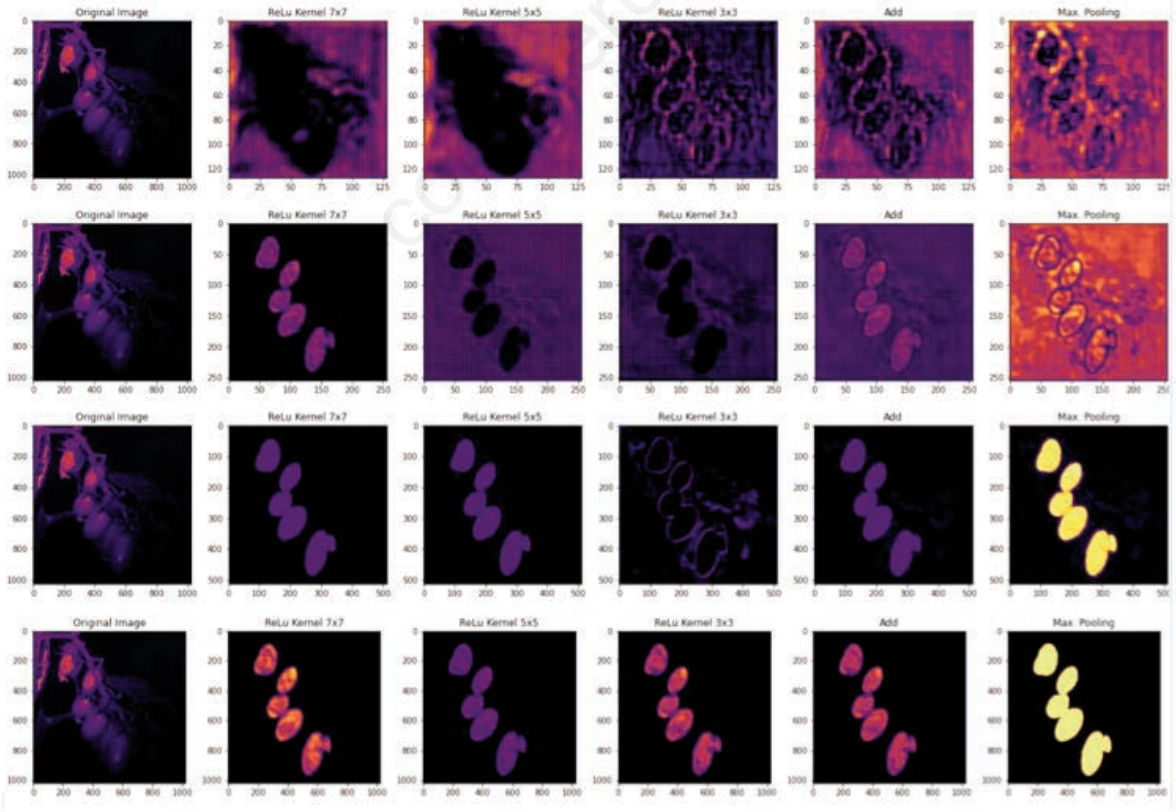

Figure 8. One random filter of each size chosen from Model 2's first deconvolution block (top line) to the last (bottom line). The first column is the original image, and the next three are the activation maps produced by different-sized kernels, followed by a sum (add) and a max-pooling layer, which give us the result for this filter. Darker colours mean lower pixel values.

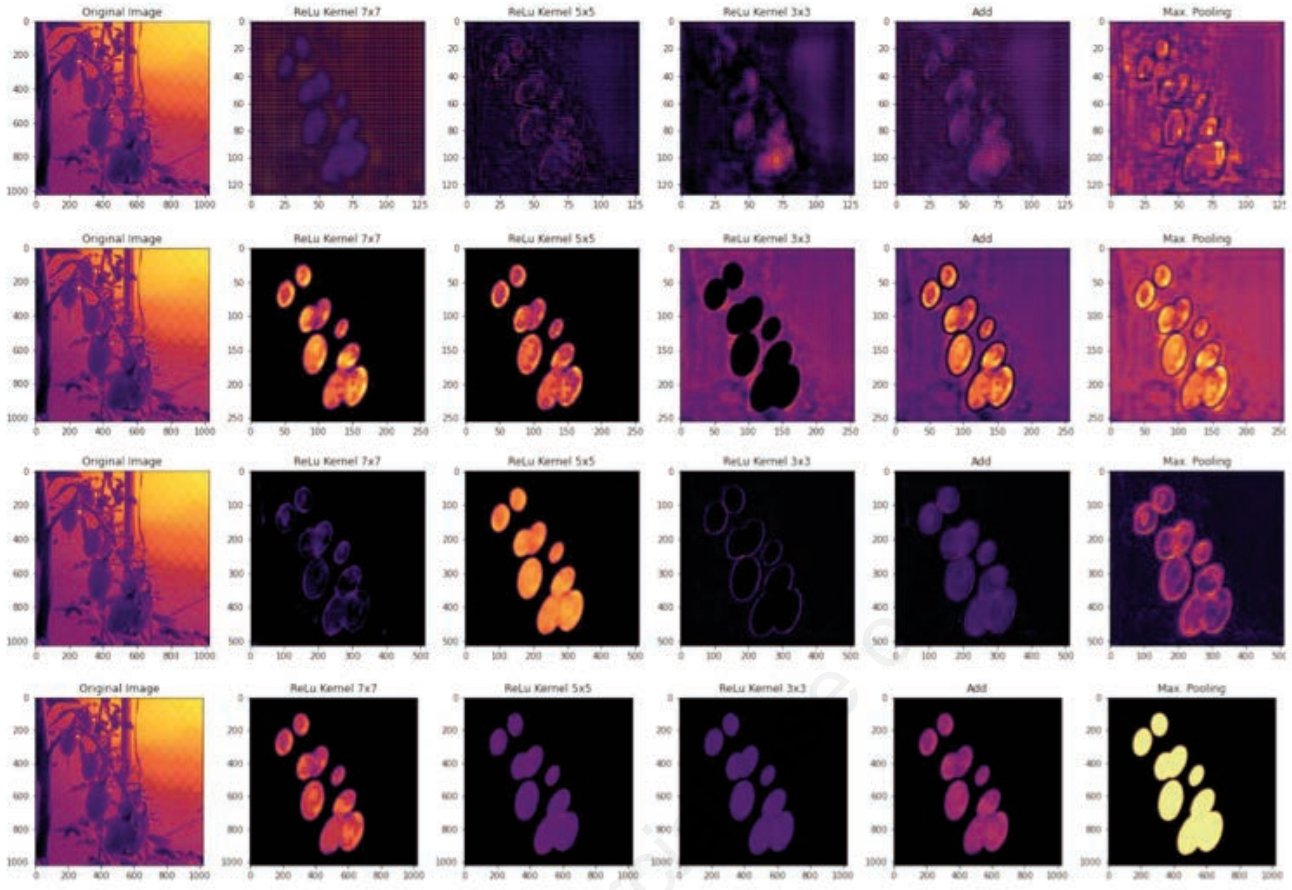

Figure 9. One random filter of each size chosen from Model 3's first deconvolution block (top line) to the last (bottom line). The first column is the original image, and the next three are the activation maps produced by different-sized kernels, followed by a sum (add) and a max-pooling layer, which give us the result for this filter. Darker colours mean lower pixel values.

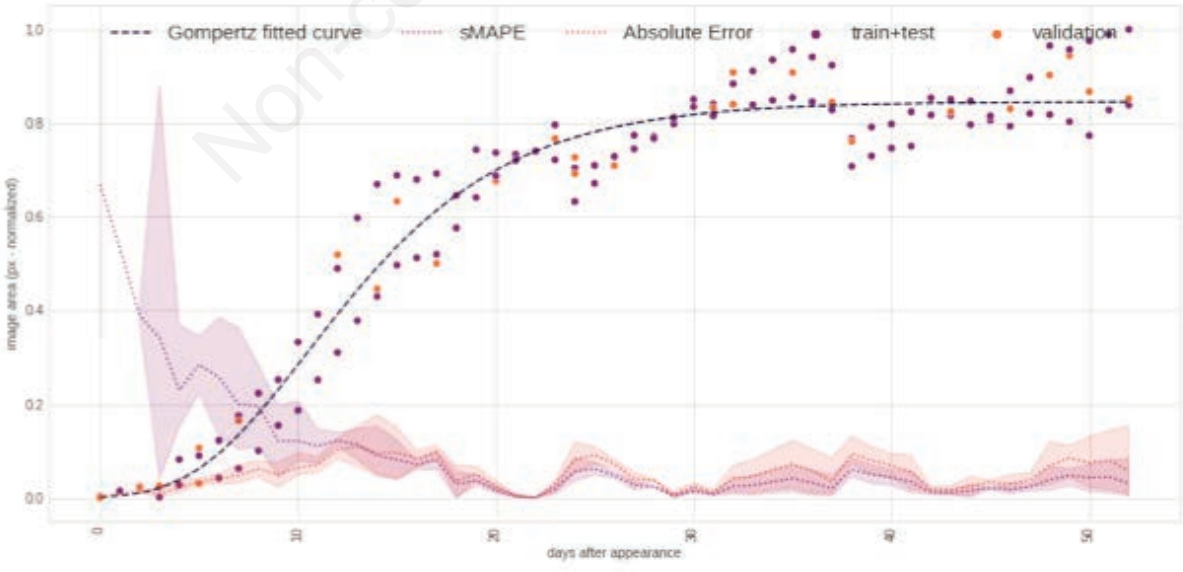

Figure 10. Optimal curve (dashed) over clusters' normalised predicted areas. This curve was fitted over train and test image sets (purple dots) and validated with validation image set only (orange dots). Regression sMAPE (purple dotted line) and absolute error (orange dotted line) curves were fitted over the entire dataset. The shaded areas represent the error variance for that point.

ment on days 24 and 38. The fruit's image size is inversely proportional to the distance between the camera and the cluster. Since we measure the cluster area variation concerning the first image collected, this distance is fundamental to the method.

sMAPE appears inversely proportional to time (Figure 10). This is mainly because it has the disadvantage of being unstable with low values, which is the case for the early stages of growth. If we disconsider this period of instability, the error falls from 10.1% to 3.8% after day 5, 2.9% after day 10, and 1.5% after day 15. However, when we confront it with the predictions of absolute error, this tendency is not noticed.

Conclusions

In this work, we proposed to use the projected area of mini tomatoes as an alternative to destructive mass measurements to describe their growth. The high correlation between the segmented areas and the Gompertz function and its performance being similar to other results in the literature suggest its usefulness as a method for quantifying growth.

The predictions' sMAPE error is higher in the early stages of growth, but not with the absolute error, which points to an sMAPE instability with low values, a well-known characteristic of this function, rather than some issue with flowers classification.

Light played an important role in model performance. Combining well and poorly illuminated scenes lead to a model with high precision and recall.

The network architecture overcame light discrepancies in all models, supporting reported results that the U-net is effective for segmentation problems even with small datasets.

The multi-scale residual blocks with different-sized kernels were more effective in the first block of the contracting path. The kernels behave as a digital camera diaphragm balancing the environment light discrepancies.

Compared to the traditional methods discussed in this paper, the proposed methodology may benefit tomato producers by providing real-time data about the crop with no need for special tests or equipment. Also, we have made the images gathered in the experiment publicly available, expecting they may contribute to future research in computer vision and agricultural fields.

References

Abreu F.F., Rodrigues L.H.A. 2022. MTIL - Mini tomato image library. Repositório de Dados de Pesquisa da Unicamp. Available from: https://doi.org/10.25824/redu/3CP9NK

Afonso M., Fonteijn H., Fiorentin F.S., Lensink D., Mooij M., Faber N., Polder G., Wehrens R. 2020. Tomato fruit detection and counting in greenhouses using deep learning. Front. Plant Sci. 11:571299.

Agrawal A., Mittal N. 2020. Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. Visual Comput. 36:405-12.

Ali M., Gilani S.O., Waris A., Zafar K., Jamil M. 2020. Brain tumour image segmentation using deep networks. IEEE Access 8:153589-98.

Bragagnolo L., Da Silva R.V., Grzybowski J.M.V. 2021. Amazon forest cover change mapping based on semantic segmentation by U-Nets. Ecol. Inf. 62:101279.

Chen J., Shen Y. 2017. The effect of kernel size of CNNs for lung nodule classification. pp. 340-344 in Proc. 9th International Conference on Advanced Infocomm Technology (ICAIT), Chengdu, China.

Chen S.W., Shivakumar S.S., Dcunha S., Das J., Okon E., Qu C., Taylor C.J., Kumar V. 2017. Counting apples and oranges with deep learning: a data-driven approach. IEEE Robot. Autom. Lett. 2:781-8.

Chollet F. 2018. Deep learning with Python. Manning Publications Co., Shelter Island, New York, NY, USA.

Faurobert M., Mihr C., Bertin N., Pawlowski T., Negroni L., Sommerer N., Causse M. 2007. Major proteome variations associated with cherry tomato pericarp development and ripening. Plant Physiol. 143:1327-46.

Fayad J.A., Fontes P.C.R., Cardoso A.A., Finger F.L., Ferreira F.A. 2001. Crescimento e produção do tomateiro cultivado sob condições de campo e de ambiente protegido. Hortic. Brasil. 19:365-70.

Fukui R., Schneider J., Nishioka T., Warisawa S., Yamada I. 2017. Growth measurement of tomato fruit based on whole image processing. pp. 153-158 in Proc. IEEE International Conference on Robotics and Automation (ICRA), Singapore.

Ganesh P., Volle K., Burks T.F., Mehta S.S. 2019. Deep orange: mask R-CNN based orange detection and segmentation. IFAC-PapersOnLine 52:70-5.

Hall D.O., Scurlock J.M.O., Bolhàr-Nordenkampf H.R., Leegood R.C., Long S.P. (Eds.). 2013. Photosynthesis and production in a changing environment: a field and laboratory manual. Springer, Amsterdam, The Netherlands.

Hemming S., Zwart F., Elings A., Petropoulou A., Righini I. 2020. Cherry tomato production in intelligent greenhouses - sensors and AI for control of climate, irrigation, crop yield, and quality. Sensors 20:6430.

Heuvelink E. (Ed.). 2005. Tomatoes. CABI Publishing, Wallingford, UK - Cambridge, MA.

Jha D., Riegler M.A., Johansen D., Halvorsen P., Johansen H.D. 2020. DoubleU-Net: a deep convolutional neural network for medical image segmentation. pp. 558-564 in Proc. IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS), Rochester, MN, USA.

Johansen K., Morton M.J.L., Malbéteau Y., Aragon B., Al-Mashharawi S., Ziliani M.G., Ángel Y., Fiene G., Negrão S., Mousa M.A.A., Tester M.A., McCabe M.F. 2020. Predicting biomass and yield in a tomato phenotyping experiment using UAV imagery and random forest. Front. Artif. Intellig. 3:28.

Khoshnam F., Tabatabaeefar A., Ghasemi-Varnamkhasti M., Borghei A. 2007. Mass modeling of pomegranate (Punica granatum L.) fruit with some physical characteristics. Sci. Hortic. 114:21-6.

Lawal M.O. 2021. Tomato detection based on modified YOLOv3 framework. Sci. Rep. 11:1447.

Liu X., Zhao D., Jia W., Ji W., Ruan C., Sun Y. 2019. Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access 7:139635-42.

Ngugi L.C., Abdelwahab M., Abo-Zahhad M. 2020. Tomato leaf segmentation algorithms for mobile phone applications using deep learning. Comput. Electron. Agric. 178:105788.

Oswell N.J., Amarowicz R., Pegg R.B. 2019. Fruits and fruit products. pp. 428-435 in Reference module in chemistry - Molecular sciences and chemical engineering. Encyclopedia of Analytical Science (Third Edition). Elsevier, Amsterdam, The Netherlands.

Öztürk S., Özkaya U., Akdemir B., Seyfi L. 2018. Convolution Kernel size effect on convolutional neural network in

histopathological image processing applications. pp. 1-5 in Proc. International Symposium on Fundamentals of Electrical Engineering (ISFEE). Bucharest, Romania.

Ronneberger O., Fischer P., Brox T. 2015. U-Net: convolutional networks for biomedical image segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (Eds.), Medical image computing and computer-assisted intervention - MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Berlin, Germany.

Santos T.T., Souza L.L., Santos A.A., Avila S. 2020. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Comput. Electron. Agric. 170:105247.

Solanke A.U., Kumar P.A. 2013. Phenotyping of tomatoes. In: Panguluri S., Kumar A. (Eds.), Phenotyping for plant breeding. Springer, New York, NY, USA, pp. 169-204.

Soltani M., Alimardani R., Omid M. 2011. Modeling the main physical properties of banana fruit based on geometrical attributes. Int. J. Multidiscipl. Sci. Engine. 2:1-6.

Song Z., Fu L., Wu J., Liu Z., Li R., Cui Y. 2019. Kiwifruit detection in field images using Faster R-CNN with VGG16. IFAC-PapersOnLine 52:76-81.

Su J., Yi D., Su B., Mi Z., Liu C., Hu X., Xu X., Guo L., Chen W.-H. 2021. Aerial visual perception in smart farming: field study of wheat yellow rust monitoring. IEEE Trans. Ind. Inf. 17:2242-9.

Tabatabaeefar A., Rajabipour A. 2005. Modeling the mass of apples by geometrical attributes. Sci. Hortic. 105:373-82.

Taheri-Garavand A., Rafiee S., Keyhani A. 2011. Study on some morphological and physical characteristics of tomato used in mass models to characterize best post harvesting options. Austr. J. Crop Sci. 5:433-8.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors. 2019. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17:261-72.

Wan S., Goudos S. 2020. Faster R-CNN for multi-class fruit detection using a robotic vision system. Comput. Netw. 168:107036.

Xie S., Girshick R., Dollár P., Tu Z., He K. 2021. Aggregated residual transformations for deep neural networks. pp. 5987-5995 in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA.

Zafar K., Gilani S.O., Waris A., Ahmed A., Jamil M., Khan M.N., Kashif A.S. 2020. Skin lesion segmentation from dermoscopic images using convolutional neural network. Sensors 20:1601.

Zhou Z., Siddiquee M.M.R., Tajbakhsh N., Liang J. 2020. UNet++: redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans. Med. Imaging 39:1856-67.