Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Computer vision based detection of external defects on tomatoes using deep learning

Arthur Z. da Costa [a,*], Hugo E.H. Figueroa [a], Juliana A. Fracarolli [b]

[a] FEEC, University of Campinas, Campinas, SP, Brazil
[b] FEAGRI, University of Campinas, Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 28 January 2019
Received in revised form
30 November 2019
Accepted 6 December 2019

Keywords:
Defect sorting
Deep learning
Sorting machines
Computer vision

Sorting machines use computer vision (CV) to separate food items based on various attributes. For instance, sorting based on size and colour are commonly used in commercial machines. However, detecting external defects using CV remains an open problem. This paper presents an experimental contribution to external defect detection using deep learning. An uncensored dataset with 43,843 images including external defects was built during this study. The dataset is heavily imbalanced towards the healthy class, and it is available online. Deep residual neural network (ResNet) classifiers were trained that are capable of detecting external defects using feature extraction and fine-tuning. The results show that fine-tuning outperformed feature extraction, revealing the benefit of training additional layers when sufficient data samples are available. The best model was a ResNet50 with all its layers fine-tuned. This model achieved an average precision of 94.6% on the test set. The optimal classifier had a recall of 86.6% while maintaining a precision of 91.7%. The posterior class-conditional distributions of the classifier scores showed that the key to classifier success lies in its almost ideal healthy class distribution. The results also explain why the model does not confuse stems/calyxes with external defects. The best model constitutes a milestone for detecting external defects using CV. Because deep learning does not require feature engineering or prior knowledge about the dataset content, the methodology may also work well with other foods.

1. Introduction

Industrial sorting and grading machines use computer vision to separate food according to several criteria, such as colour, size and texture; however, the current commercial solutions using traditional computer vision systems (TCVS) still lack algorithms capable of detecting most types of external defects, such as bruises, rain damage, or abrasions (Cubero, Lee, Aleixos, Albert, & Blasco, 2016; Li, Huang, & Zhao, 2015). The presence of external defects decreases the price of food for two reasons. First, external defects make food look worse. Second, they are indicators of less nutritious or even infected foods. A better detection algorithm would be able to sort the fruit, sending the best fruit to the consumer market and the worst fruit to the food industry. The net result would be an increase in profits for the machine owner and an increase in food safety and satisfaction for the consumers. To improve

* Corresponding author.
E-mail addresses: arthurzc23@gmail.com (A.Z. da Costa), hugoehf@gmail.com (H.E.H. Figueroa), juliana.fracarolli@unicamp.br (J.A. Fracarolli).
https://doi.org/10.1016/j.biosystemseng.2019.12.003
1537-5110/

Nomenclature

λ	L_2 regularization coefficient
Acc	Accuracy, %
AI	Artificial Intelligence
AP	Average Precision, %
AUC	Area Under the Receiver Operating Characteristic Curve, %
convy_x	Convolutional Block y
CV	Computer Vision
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
GPU	Graphics Processing Unit
ILSVRC	ImageNet Large Scale Visual Recognition Competition
JPEG	Joint Photographic Experts Group
L_2	regularisation $\|\|.\|\|_2^2$
MLP	Multilayer Perceptron
OT	Optimal Threshold, %
P	Precision, %
per	Percentile
$\|\|\cdot\|\|_p$	p norm
PR curve	Precision Recall Curve
R	Recall, %
ResNet	Residual Network
ResNetx	Residual Network With x Layers
TCVS	Traditional Computer Vision Systems

defect detection, two main challenges must be solved: detecting a large variety of defects with different visual features and distinguishing between stems/calyxes and genuine defects.

To improve defect detection, some authors have proposed the use of multispectral or hyperspectral imaging (Cubero et al., 2016; Zhang et al., 2014; Li et al., 2016; Wang et al., 2011; Folch-Fortuny, Prats-Montalbán, Cubero, Blasco, & Ferrer, 2016). Despite some promising results, both methods still have shortcomings when compared with TCVS. Hyperspectral systems are expensive, ranging from dozens to hundreds of thousands of dollars[1]; furthermore, they do not acquire images as fast as TCVS. Multispectral systems are not as expensive as hyperspectral solutions, but they are still more expensive than TCVS. Moreover, it is common for a multispectral system working at specific bands to excel at one task while performing poorly in others (Zhang et al., 2014). This limitation is particularly serious in defect detection tasks because of the different types of defects and their distinctive visual features. Finally, TCVS are more available on the market and easier to set up and maintain.

A 2014 review of external inspection methods for fruit and vegetables shows that most algorithms for defect detection are based on multispectral or hyperspectral imaging (Zhang et al., 2014). The few methods based on TCVS report lower scores and are restricted to a few types of defects that significantly alter colour or texture and do not resemble the calyx or peduncle (Hu, Dong, Liu, & Malakar, 2014; Blasco, Aleixos, & Moltó, 2007; Xiao-bo, Jie-wen, Yanxiao, & Holmes, 2010). The review also shows that both TCVS and non-TCVS use the same standard machine learning techniques, such as principal component analysis, K-means, linear discriminant analysis and classical artificial neural networks. Therefore, it appears that increasing the performance of TCVS requires going beyond standard machine learning algorithms and testing the state-of-the-art models. Recently, the neural network community has witnessed the greatest revolution to date in artificial intelligence (AI) due to the introduction of a new paradigm: deep learning. The deep learning proposal is rather simple: train a neural network with several intermediate layers. The end result is a deep neural network (DNN) capable of extracting patterns and abstractions from data hierarchically from layer to layer until it reaches the final layer and performs the required task. Despite this simple idea, more than three decades elapsed from the introduction of the deep convolutional neural network to the first outstanding results of a DNN in the ImageNet Large Scale Visual Recognition Competition (ILSVRC)[2] (Krizhevsky, Sutskever, & Hinton, 2012; Fukushima & Miyake, 1982; ImageNet Large Scale Visual Recognition Competition (ILSVRC), 2017; Deng et al., 2009). Due to an enormous increase in available data, better parallel computer hardware and theoretical advances in neural networks, deep learning solutions now dominate the state of the art for object classification, object detection, object localisation, image translation, speech to text, natural language processing and several other AI-related tasks (Chiu et al., 2017; Huang, Yu, & Wang, 2018; LeCun, Bengio, & Hinton, 2015; Young, Hazarika, Poria, & Cambria, 2017).

Among its other achievements, deep learning has already demonstrated remarkable results in various agricultural applications. For instance, deep neural networks have achieved state-of-the-art results in several types of research involving the disease detection in food and plants (Arnal Barbedo, 2019; Brahimi, Boukhalfa, & Moussaoui, 2017; Cruz et al., 2017, 2019; Lu, Yi, Zeng, Liu, & Zhang, 2017). Other areas of research, such as weed species classification (Dyrmann, Karstoft, & Midtiby, 2016; dos Santos Ferreira, Matte Freitas, Gonçalves da Silva, Pistori, & Theophilo Folhes, 2017), mapping vegetation-quality coverage (Minh et al., 2017) and crop type estimation (Ruβwurm & Körner, 2017), have also undergone significant improvements by adopting deep learning. Nevertheless, several agricultural problems exist where deep learning could have a significant impact but has not yet been studied (Kamilaris and Prenafeta-Boldú, 2018). In particular, no previous works could be found that used deep learning to detect external defects in food using a big dataset (containing at least tens of thousands of images). As a result, one goal of this work was to use a massive dataset to develop a deep learning detection model that could detect external defects in food. For this study, a deep binary classifier was adopted because such models are not only simple and well-understood (Goodfellow, Bengio, & Courville, 2016; Murphy, 2012) but are also capable

[1] Quotation acquired in the beginning of 2018.

[2] The ILSVRC is the most famous computer vision competition in the world. Every year, the winners set new benchmark levels for object classification, detection and localisation in images. Since 2012, all the winning models have been based on deep neural networks.

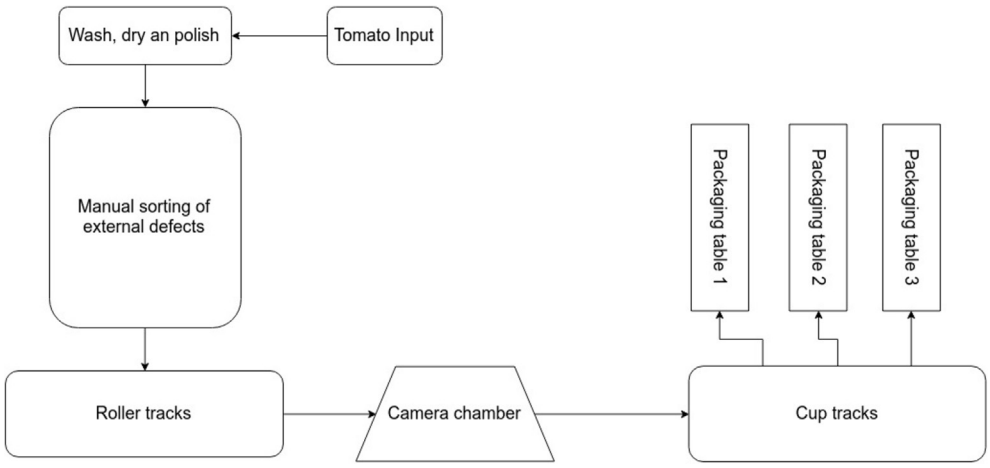

Fig. 1 – Tomato-sorting flowchart.

of delivering state-of-the-art results. This paper does not focus on identifying particular types of defects, such as differentiating between rottenness or bruises. The only goal is to detect defects. Due to the time required to build a proper dataset for the task, this study analysed only one fruit: the tomato.

2. Materials and methods

2.1. *Data acquisition*

Because deep neural networks are large capacity models that are prone to overfitting, they require large datasets to achieve sufficient generalisability. However, the limited availability of open source datasets in agricultural fields, especially those with thousands of samples or more, is a well-recognised problem in academia (Kamilaris and Prenafeta-Boldú, 2018) because the lack of sufficient samples makes benchmarking and training DNNs difficult. Additionally, small datasets might overestimate the performances of some algorithms due to the lack of variety within class instances (Dyrmann et al., 2016). As such, the compilation of a big[3] high-quality tomato dataset is one of the key contributions of this work.

All the images were acquired from a Fomesa tomato-sorting and grading machine working normally during business hours at a commercial establishment in Campinas, Brazil. The machine is not capable of detecting external defects automatically. Figure 1 shows a flowchart of the complete sorting process. Appendix A dedicates a full figure for each step on the flowchart with the exception of Fig. 4, which shows the roller tracks and camera chamber simultaneously. The author acquired the machine images at random business days over 2 months. Three Brazilian varieties of tomatoes pass through this machine: Italiano, the hybrids Carmen and Debora. The cameras are positioned at the top of the camera chamber and capture images of the black roller tracks below them. Six roller tracks pass through the camera chamber, each illuminated by two lamps. The lamps are OSRAM L36 W/827-1 fluorescent tubes. Figure 2 displays the interior of the camera chamber. The tomatoes roll on the track so that the camera can film their entire surface because the TCVS requires a view of the entire fruit surface to make accurate predictions about each tomato. The speed of the machine is controlled by an R4K7 potentiometer. At maximum speed, the camera chamber has an estimated throughput of 8 tomatoes per track per second (i.e., a total of 48 tomatoes per second considering all 6 tracks). The Fomesa machine's hardware includes a rack with three computers: one main computer for control and two vision computers that process the images from the different tracks in parallel. The frames acquired by the machine's cameras during normal grading are saved. No change was made in any machine configuration or in lighting to facilitate the classification task. More importantly, no restrictions were imposed concerning which types of defects passed through the machine or were included in the dataset. Thus, the acquired images are a reliable representation of what to expect in a real-life sorting and grading scenario.

At the end of the image acquisition phase, 100,000 frames with a size of $100 \times 100 \times 3$ were captured and saved as JPEG images. Next, through a visual inspection the images were filtered to remove frames containing no tomatoes or highly similar frames. After the filtering process 43,843 suitable images were obtained.

It is important to note that the filtering was not performed to simplify the classification task: the sorting machine ignores frames with no tomatoes, and repetitive similar frames are not useful for the TCVS.

The final step was labeling. When a tomato was fit for the consumer market in terms of both health and beauty, the image was labelled as healthy and encoded as a 0. If the tomato was fit for the food industry, e.g., the ketchup factory, the image was labelled defective and encoded as 1. It is

[3] A "big" dataset is a subjective measure. In the deep learning literature, a large dataset might imply hundreds of thousands or even millions of samples. In this paper, a dataset is considered "big" if it contains more than 10,000 instances.

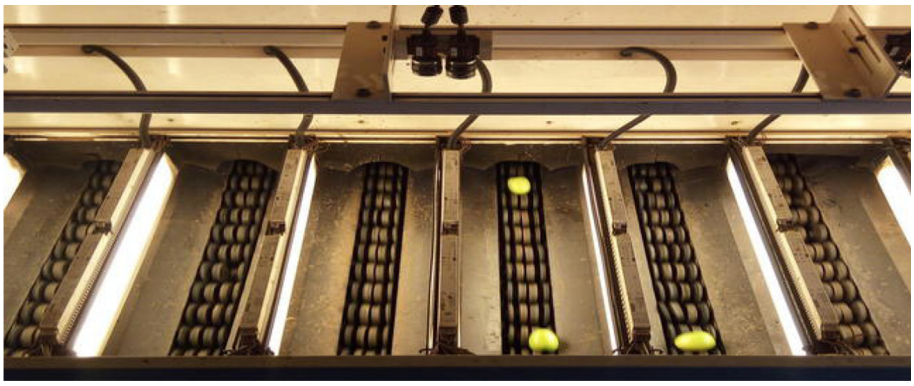

Fig. 2 – Inside the camera chamber. The cameras are positioned at the top of the chamber, filming the black roller tracks below them. The machine has 6 tracks, each illuminated by two lamps.

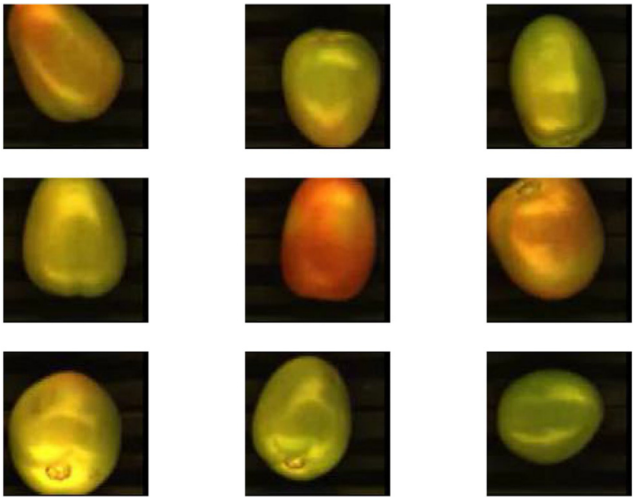

Fig. 3 – Images of healthy tomatoes acquired from the sorting machine.

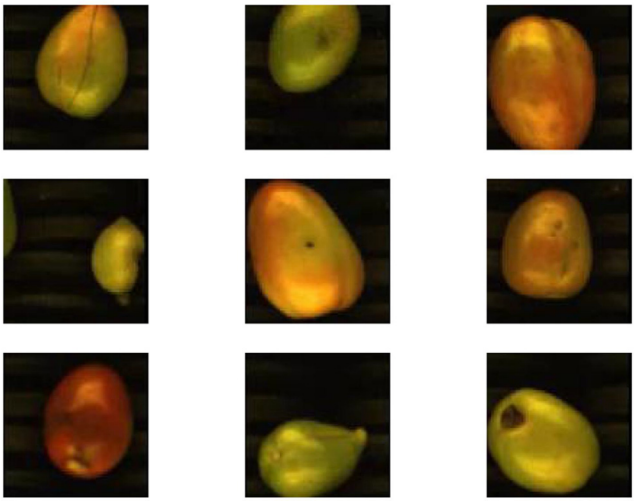

Fig. 4 – Images of defective tomatoes acquired from the sorting machine. The images show the following external defects: from left to right and top to bottom are longitudinal cut, dark spot, rain bruise, small and deformed fruit, puncture, punctures and bruises, rain spot, longitudinal cut, large and dark peduncle. The images give an overview of the variety of defects in the constructed dataset.

important to observe that when a healthy tomato does not have a pleasant appearance, it should be classified as a 1. An expert tomato grader who had worked with the factory sorting machine for more than 10 years helped to label the images. The final dataset contains 38,884 healthy samples and 4959 defective samples; thus, the dataset is highly imbalanced towards healthy instances because of the quality control of the tomato suppliers. Figures 3 and 4 show examples of healthy and defective samples, respectively. In Fig. 4, the tomato with a large and dark calyx is labelled as defective because its appearance is not ideal for the consumer market. Nevertheless, a healthy tomato with a normal peduncle is not considered defective, as some of the samples in Fig. 3. Figure 4 provides a glimpse of the variety of defects in the dataset and their different visual features. Appendix B contains additional images of defective tomatoes. The dataset was split into training, validation and testing parts. The images were sorted before separation. In total, 50% of the images were designated for training, 25% for validation and the remaining 25% for testing. Steps were taken to ensure that the new datasets had approximately the same class distribution as the original dataset. The dataset is freely available for research purposes.[4]

The constant black background, controlled lighting and the centralised tomatoes in each image facilitates machine learning vision-related tasks using this dataset. In any case, there was no need to change the background, lighting or to include more than one tomato in each image. The camera chamber in the sorting machine is designed to facilitate the sorting process; thus, those parameters are already

[4] The dataset is available at https://github.com/ArthurZC23/Deep-learning-classifier-for-external-defects-in-tomatoes.

controlled. Changing the parameters to make the dataset more challenging would defeat the purpose of creating an algorithm to detect external defects in sorting machines.

Despite some control in the image parameters, the task of detecting external defects in this dataset is not simple due to the nature of the problem. First, some defects, such as bruises, rottenness and chilling injuries, do not significantly alter fruit colour or texture. Most of the current algorithms for TCVS are based on manually designed features that involve colour or texture; thus, they are not suitable for detecting these types of defects (Zhang et al., 2014). Second, as no defect types were censored during the image acquisition step, the characteristic visual features of external defects are not well defined. Owing to the variety of external defects and their visual presentations, it is almost certain that the dataset lacks some types of external defects and includes some that are poorly sampled. The lack or poor sampling of some types of external defects makes classification more challenging. Classifiers and deep neural networks are no exception; they learn the decision boundary between its classes during the training phase. The final dataset is heavily imbalanced, with fewer instances of the large variety class. As a result, the differentiation process must rely primarily on the decision boundary of the healthy tomato class.

2.2. *Image preprocessing*

One major benefit of deep neural networks is their ability to work with raw data or minimally processed data. There is no need for segmentation, feature engineering or feature selection. During training, the intermediate layers of a DNN learn what features to compose hierarchically. For instance, for image classification tasks, DNNs detect edges in the first layers, textures and simple shapes in the intermediate layers and entire objects in the final layer (Zeiler & Fergus, 2014). This property of DNNs is very desirable in the current problem because it means that the model should also work with different types of food if sufficient data is available for training.

2.3. *Neural network architecture*

Rather than using a traditional multilayer perceptron (MLP), DNNs use other network architectures. Deep convolutional neural network (DCNN) models are the standard choices for computer vision problems (Goodfellow et al., 2016) and were also selected for this research. The network has convolutional layers for processing grid-like data, e.g., images, and pooling layers for calculating summary statistics. The network may also include traditional MLP layers after the convolutional and pooling layers.

The specific selected DCNN models are residual networks (ResNets) (He, Zhang, Ren, & Sun, 2016). The residual architecture includes network shortcuts, as shown in Fig. 5. The purpose of these shortcuts is to provide references for deeper layers. Instead of learning unreferenced functions from scratch, the layers learn the residual functions with reference to the shortcut input. The decision to use ResNets considered their generalisation potential while remaining simple to train and understand. Strong empirical evidence shows that residual networks are easier to train than previous DCNN models, and they are more generalisable due to their increased depth (He et al., 2016). Meanwhile, ResNets are easier to implement and understand than the DNNs that came afterwards because the former are constructed of repetitions of simple neural layers in a manner similar to the VGG model (Simonyan & Zisserman, 2014).

2.4. *Transfer learning*

Training DNNs from scratch is challenging: they are difficult to optimise, require powerful GPUs and need thousands (if not millions) of training samples. Transfer learning is a common strategy used in deep learning to avoid some of these problems (Yosinski, Clune, Bengio, & Lipson, 2014). The idea behind transfer learning is to start with a pretrained DNN optimised for one task and then transfer its knowledge to a different model. In practice, this means using the pretrained weights as the starting point of optimisation rather than using a DNN initialisation strategy (Mishkin & Matas, 2015).

Thus, instead of training the ResNets from scratch, this study adopted transfer learning because it facilitates the training effort. The adopted models were ResNets pretrained on the ImageNet dataset for object classification (*ImageNet Large Scale Visual Recognition Competition (ILSVRC)*, 2017; Deng et al., 2009). Two different methods were tested: feature extraction and fine-tuning. The feature extraction approach freezes all the weights except for the softmax classification layer. The DCNN acts as an image feature extractor, and the only goal is to train a linear softmax classifier on those features. The fine-tuning approach allows for part of the network to receive additional training. Some intermediate layers are fine-tuned to increase the network's performance on the desired task. Fine-tuning usually occurs from the last layers to

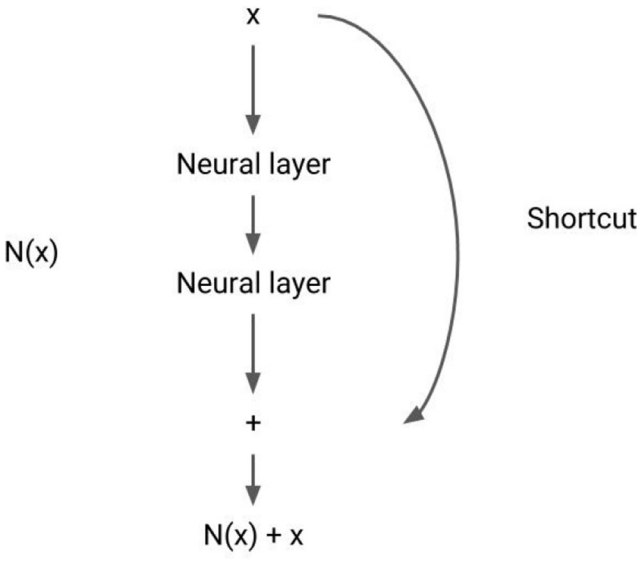

Figure 5 – Residual neural architecture. In addition to the traditional path, the tensor x passes through a residual shortcut. As a result, the function N to be learned needs to learn only a residue with reference to x to perform the desired task.

the first layers because the low-level learned features generalise better across different tasks.

Because the tomato dataset has 100 × 100 images and ImageNet contains 224 × 224 images, the ResNets needed an adjustment. The last layer before the softmax is a 7 × 7 average pooling layer in all the ResNet models reported in (He et al., 2016). As the pooling size decreases, smaller images can fit the network, and the softmax has more parameters. For feature extraction, the pooling size is a hyperparameter. Here, the sizes 4 × 4, 2 × 2 and 0, i.e., no pooling layer, were validated. For fine-tuning, the pooling size was set to 4 × 4. Furthermore, the softmax layer was altered to match the number of outputs of the pooling layer and to have only two classes.

Another important consideration when working with transfer learning from the ImageNet dataset is to perform basic image preprocessing tasks. The preprocessing applied here involved the following steps:

1. Load RGB images as arrays where each pixel is in the range [0, 1].
2. Subtract 0.485, 0.456, and 0.406 from each pixel of the R, G and B channels, respectively. These are the mean values for each channel of the ImageNet dataset.
3. Divide each pixel of the R, G and B channels by 0.229, 0.224, and 0.225, respectively. These are the standard deviations for each channel of the ImageNet dataset.

Applying a standardisation procedure to the input images is a recommended step because it accelerates convergence (Ioffe & Szegedy, 2015). The mean and standard deviations of the ImageNet dataset were used in this standardisation. Favouring the original dataset summary statics is reasonable because all the pretrained weights were determined using them.

2.5. *Training strategy*

Both transfer learning strategies were trained in similar ways. The loss function was cross-entropy loss with L_2 regularisation ($\left|\left|.\right|\right|_2^2$). The L_2 regularisation coefficient λ was a hyperparameter of the models. Feature extraction had the last pooling layer size as a hyperparameter, as described in 2.4. Fine-tuning had the number of fine-tuned layers as a hyperparameter. The layers were separated into blocks following the convention of (He et al., 2016), and the models were trained with the following fine-tuned blocks: conv5_x, conv4_x-conv5_x, conv3_x-conv5_x and full network. In all cases, the softmax layer was trained from scratch. The fine-tuning process jumped from conv3_x-conv5_x to full network fine-tuning because the increase in the number of parameters from conv2_x-conv5_x to the full network is minimal compared to the increase in the other cases.

The optimisation used mini-batch backpropagation under the first-order Adam algorithm with default hyperparameters (Goodfellow et al., 2016). The initial learning rates were 10^{-2} and 10^{-3} for feature extraction and fine-tuning, respectively. The mini-batch size was 64. During training, there was a 50% chance that the image would be flipped horizontally before entering the network for data augmentation. The traditional area under the receiver operating characteristic curve (*AUC*) was not used as a performance measure because the dataset is heavily imbalanced. In this situation, the precision–recall curve (PR curve) is more appropriate (Davis & Goadrich, 2006). Here, average precision (*AP*) was adopted as performance metric (Boyd et al., 2013). An early termination meta-algorithm (Goodfellow et al., 2016) was used to validate the best number of epochs to train each network. The early termination algorithm used in this study checked whether the validation *AP* score increased in each epoch. If the validation performance improved, the weights and the current epoch were recorded. If the validation performance did not improve for 5 consecutive epochs, the learning rate was decreased to 10% of its current value. The optimisation ended when a learning rate threshold was achieved and the best weights and epochs were returned. The thresholds were 10^{-5} and 10^{-6} for feature extraction and fine-tuning, respectively.

Model selection for the ResNet architecture and the L_2 regularisation coefficient λ used the following three-step strategy:

1. A wide grid search where all architectures presented in (He et al., 2016), i.e., ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152, are combined with $\lambda \in \{2^{-10}, 2^{-5}, 2^{0}, 2^{5}, 2^{10}\}$ for feature extraction and $\lambda \in \{2^{-10}, 2^{-5}, 2^{0}, 2^{3}\}$ for fine-tuning. The best validation *AP* score on this step determined which architecture was used in the next two steps.
2. A neighbourhood power-of-2 grid search around the best result of the previous step was performed. For instance, if the best score from step 1 was ResNet34 with $\lambda = 2^{-5}$, the second step trained ResNet34 models with $\lambda \in \{2^{-7}, 2^{-6}, 2^{-4}, 2^{-3}\}$. If the best result was one of the edges, e.g., $\lambda = 2^{-7}$, then the model with $\lambda = 2^{-8}$ was trained because it may contain the best validation *AP* score.
3. A 20-point linear grid search around the best result so far was performed. For instance, if ResNet34 with $\lambda = 2^{-8}$ was the best model thus far, then ResNet34 models with $\lambda \in \left\{ 2^{-9} + \frac{k}{21}(2^{-7} - 2^{-9}) \middle| k = 1, \ldots, 20 \right\}$ were trained.

Model selection was used both for feature extraction and fine-tuning. The three-step strategy was repeated for each different pooling kernel for feature extraction and for each different number of fine-tuned layers for fine-tuning.

3. Results

All the code for this experiment was written in Python. The deep learning framework PyTorch was used to define the residual network computational graphs and to optimise them in accordance with the two transfer learning strategies. The optimisation made use of NVIDIA GTX 1080 Ti GPU to increase training speed. Tables 1 and 2 show the 10 best results for feature extraction and fine-tuning, respectively.

Comparing the results of Tables 1 and 2, the ResNet50 model with $\lambda = 0.0004185$ (discovered on the third step of full network fine-tuning and trained for 10 epochs) achieved the

Table 1 – 10 best feature extraction ResNet deep neural network models for binary classification of external defects in the adopted dataset.

Model	λ	Step	Best epoch	Pooling kernel size	Train AP (%)	Validation AP (%)
ResNet101	1.526	3	22	0	83.3	81.5
ResNet101	1.000	1	18	0	84.1	80.5
ResNet101	1.132	3	25	0	84.0	80.4
ResNet101	0.7368	3	36	0	84.6	80.0
ResNet101	1.289	3	21	0	82.9	80.0
ResNet101	1.921	3	25	0	81.6	79.7
ResNet101	0.5789	3	20	0	85.3	79.5
ResNet34	0.3618	3	16	2 × 2	82.7	79.2
ResNet34	0.2500	2	27	2 × 2	83.7	78.6
ResNet34	1.000	2	25	2 × 2	80.9	77.9

Table 2 – 10 best fine-tuned ResNet deep neural network models for binary classification of external defects in the adopted dataset.

Model	λ	Step	Best epoch	Fine tuned layers	Train AP (%)	Validation AP (%)
ResNet50	0.0004185	3	10	All layers	96.2	94.2
ResNet50	0.0003488	3	13	All layers	96.4	94.1
ResNet50	0.0007673	3	25	All layers	96.9	94.0
ResNet50	0.0009766	1	29	All layers	96.9	93.6
ResNet50	0.0003139	3	9	All layers	95.0	93.6
ResNet50	0.0002790	3	24	All layers	97.6	93.5
ResNet50	0.0008022	3	18	All layers	96.2	93.2
ResNet50	0.0004883	2	13	All layers	96.9	93.1
ResNet50	0.0005929	3	9	All layers	95.3	92.4
ResNet50	0.0006278	3	30	All layers	97.1	92.2

best result: $AP = 94.2\%$ on the validation set. The best model performance on the test set was $AP = 94.6\%$.

4. Discussion

The experiment showed satisfactory results. Fine-tuning significantly outperformed feature extraction, and all the top 10 best fine-tuning results updated all the layers. This result shows the potential of deep learning when powered with a sufficient quantity of data. Feature-extraction transfer learning involves nothing more than combining good prior visual features with a traditional softmax classifier. There are no intermediate neural layers that need to learn data-driven features. As more layers of the network were tuned during fine-tuning, a tradeoff appeared between having good prior visual features and having more layers. Because neural layers adapt to become data-driven features in the presence of sufficient data, these findings make sense.

The training, validation and test AP scores of the best model were close, indicating that there was little or no overfitting. The size of the dataset, the L_2 penalty and the early stopping meta-algorithm were sufficient to tune all layers of the ResNets efficiently.

Figure 6 shows the PR curve of the best model on the test set. A PR curve shows the tradeoff between the precision P and the recall R as the score threshold of the model changes. R should be increased to guarantee that all the defects will be detected. However, as R increases, it is common for some healthy tomatoes to be mistaken for defective ones, decreasing P. Ideally, the upper-right corner of the curve should reflect $R = 100\%$ and $P = 100\%$, which is a perfect score. However, this is often impossible in real scenarios, because the posterior class-conditional score distributions have some degree of overlap. Figure 6 explains why the AP of the proposed model was high. P remains close to 100% for R values as high as 80%. The optimal point, i.e., the point closer to the perfect score, is $R = 86.6\%$, $P = 91.7\%$. Owing to the severe dataset imbalance, $P_{optimal} = 91.7\%$ has almost no impact on

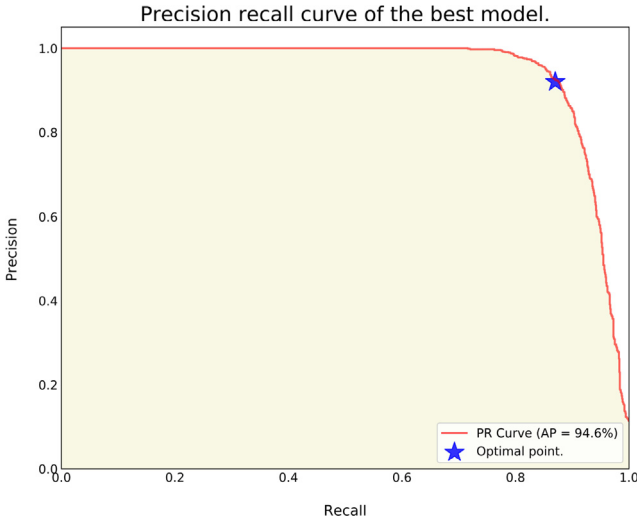

Fig. 6 – Precision-recall curve of the best model on the test set. The precision remains close to 100% for recall values as high as 80%. The optimal point (i.e., the closer to the upper-right corner) is $P = 91.7\%$ and $R = 86.6\%$.

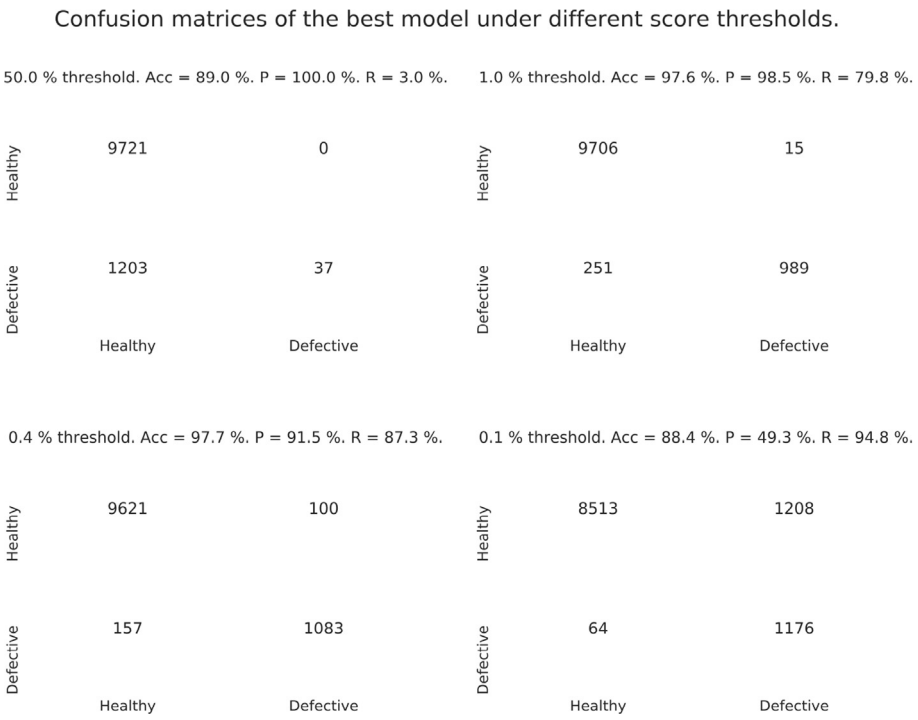

Fig. 7 – Confusion matrices for the best model. The row labels correspond to the true classes, and the column labels correspond to the predicted classes.

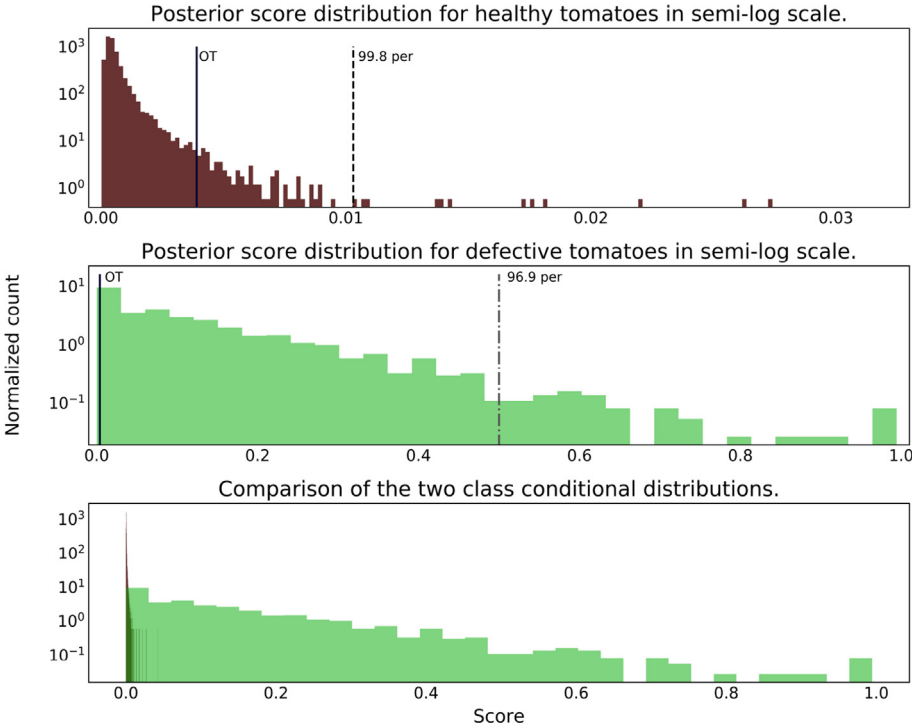

Fig. 8 – Posterior class-conditional score distributions for both classes. OT is the optimal threshold at approximately 0.4%. The 99.8th percentile (per) for the healthy class occurs at the 1% score. The 96.9th percentile for the defective class occurs at the 50% score.

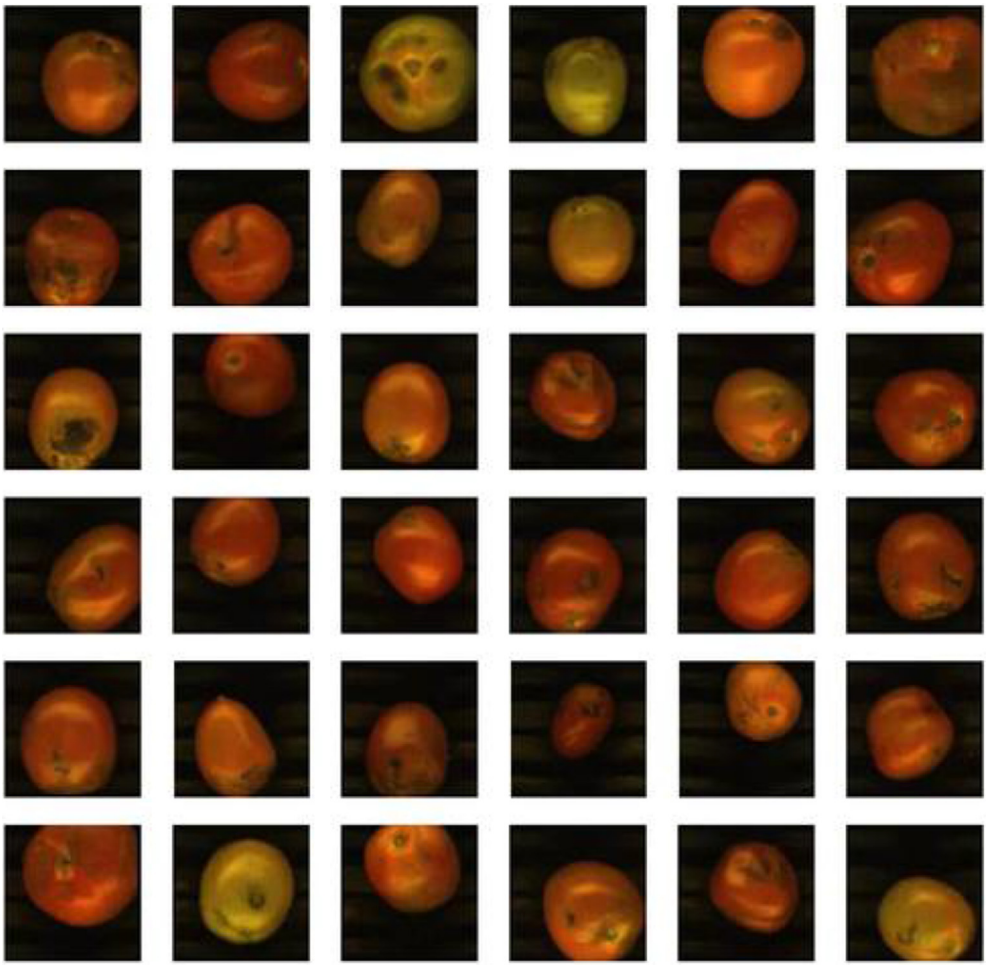

Fig. 9 – Defective samples with scores above 50%. The tomatoes are severely damaged, exhibiting decay, large punctures, cuts and black spots.

the healthy class. Of the 10,961 tomato images on the test set, 9721 are healthy and 1240 are defective. Thus, at the optimal point, only approximately 1.2% of the healthy tomatoes are misclassified.

The user responsible for configuring the sorting machine can set the classifier to operate at or near the optimal threshold based on whether it is desirable to recall more/fewer external defects with the decrease/increase in the precision. Figure 7 shows the confusion matrices for the model operating at 4 different threshold scores. The 50.0% threshold, which is traditionally used on balanced classification problems, has mediocre results. The accuracy (Acc) = 89.0% and P = 100.0%, but R = 3.0%. In contrast, near the optimum point, Acc = 97.7%, P = 91.5% and R = 87.3%. Not only does the latter model have considerable recall, but its Acc is also far greater than that of the former model. On the one hand, taking the threshold near the optimum point as a reference for the two remaining operating points in Fig. 7, increasing its value implies a drop in R nearly equal to the increase in P. On the other hand, decreasing the value of this threshold implies a small increase in R, at the cost of a significant drop in P.

The PR curve summarises the model performance at any threshold value. The confusion matrices show more details of model behaviour at different operating points. However, none of the matrices provide a straightforward visual explanation of why the model performed so well on an unbalanced dataset when the high-variance defective class had few samples. The posterior class-conditional score distribution for both classes plotted in Fig. 8 gives the explanation. The distributions were estimated using normalised histograms on a semi-log scale. The histograms have equal-width bins, and the number of bins was determined using Knuth's rule. Knuth's rule is best suited for this task compared to the classical Scott and Freedman-Diaconis rules not only because it uses a fitness function to determine the optimal number of binning but also because it does not assume that the distribution is Gaussian (Knuth, 2006). The posterior distributions summarise the difficulty of defect detection well. On the one hand, the healthy tomato class has an ideal scenario. The distribution has a Dirac delta shape with 99.8% of the probability mass concentrated between 0.0% and 1.0% scores. On the other hand, the defective class has a poor scenario. The distribution is right-skewed,

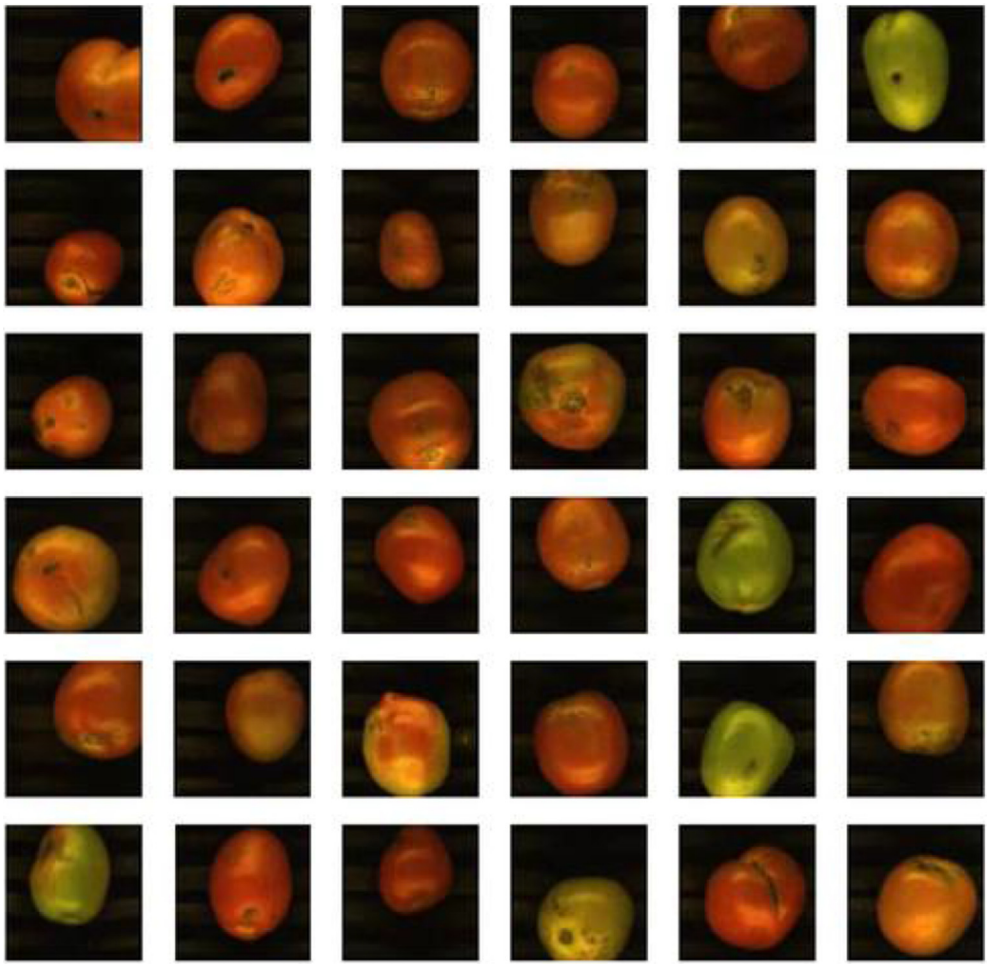

Fig. 10 – Defective samples with scores between 50% and the optimal score (0.4%). The tomatoes exhibit less intense decay, smaller punctures, cuts and black spots. Additionally, there are some tomatoes with abrasions.

with most probability masses concentrated on the wrong side of the plot. Figure 8 clarifies that the classifier did not learn the patterns of the defective tomato class well.

Despite having little knowledge of what a defect is, the classifier works quite well because it has an almost perfect sense of what a healthy tomato is. Hence, as the score threshold is swept from the right to the left of the distributions, the recall increases with almost no loss in precision until the score reaches 1.0%. Thereafter, the increase in the recall begins to have an impact on the precision because the threshold passes by the impulse-shaped healthy class distribution. The optimal point occurs at approximately the 0.4% score, once again affirming how well the classifier is able to identify a healthy tomato.

Figures 9–11 show some defective tomatoes based on their score range. As expected, the tomatoes of each score range exhibit different intensities and types of visual features that are considered to be external defects. Tomatoes with scores above 50.0% are severely damaged, displaying decay, large punctures, cuts and black spots. Tomatoes with scores between the optimal score (0.4%) and 50.0% display less intense decay, smaller punctures, cuts and black spots. Additionally, this score range included some tomatoes with abrasions. Tomatoes with scores below the optimal score (i.e., those that are not detected by the optimal classifier) present very discrete bruises and deformations. The presence of tomatoes with light abrasions is particularly strong in this range. The visual features of the samples make sense in a classification framework with a high diversity class. On one extreme are severely damaged tomatoes that do not look at all like healthy tomatoes and have high classification scores. At the other extreme are tomatoes with light abrasions that appear more similar to healthy tomatoes than to defective ones and have such low classification scores that even the optimal classifier cannot detect them. The fact that only a small number of defective samples exists to reinforce the fact that light abrasions are defects simply makes the problem more challenging.

The classification method presented in this paper could easily be extended to more images of the same fruit easily, thereby enabling the system to take into account more, if not all, of the fruit surface area. To detect external defects using multiple images, each image would pass through the deep neural network and be classified. At the end, any image classified as a defect represents a defective tomato; otherwise, the

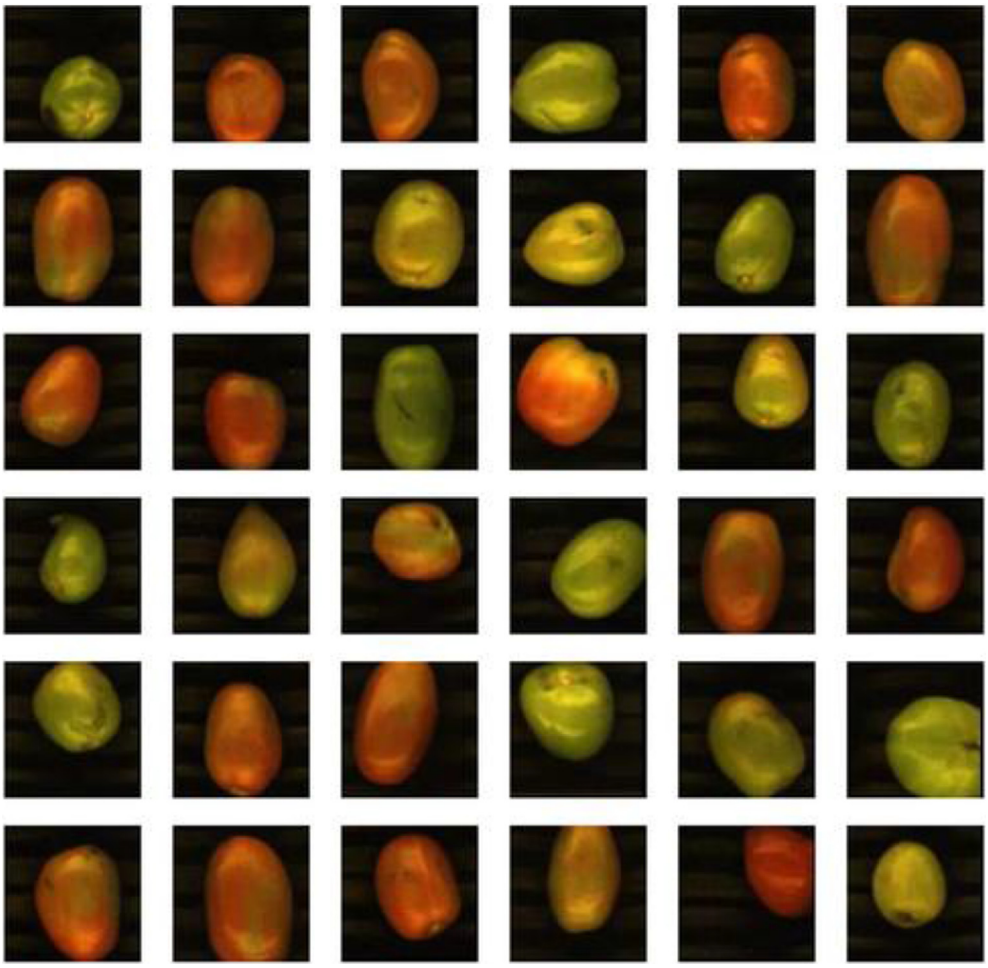

Fig. 11 – Defective samples with scores below the optimal (0.4%). These tomatoes exhibit very discrete bruises, deformations or have light abrasion.

tomato is healthy. Figure 12 shows a decision flowchart representing the detection process to be implemented in the sorting machine. The flowchart concerns itself only with the detection process inside the camera chamber; it abstracts most of the sorting machine details such as sensors and actuators. The detection threshold is chosen by the user and can be either the optimal value (0.4%) or another value that satisfies the precision and recall requirements of the sorting process. TCVS sorting machines for other foods could use the same decision flowchart, provided that they trained a DNN with data from their sorting machine, generated a new PR curve and computed the optimal threshold.

Apart from performance, another major requirement for the model in a real sorting and grading scenario is classification throughput. During the test phase, the best model was able to classify approximately 2150 images per second on the same computer used for training. This result was obtained by measuring the elapsed time required for the model to classify all the test datasets with the *time* Python native library and then dividing the number of instances of the test set (10,961 images) by the elapsed time. Consider a machine operating at max speed with an estimated throughput of 48 tomatoes per second, and assume that each tomato requires 4 pictures to cover its whole surface. Then, the throughput is 192 images per second. Next, if the new version of the hardware were to have one GTX 1080 Ti for each vision computer, it could handle $2 \times 2150 = 4300$ images per second—more than enough for this sorting machine. For more demanding applications, each vision computer could stack up to 4 GTX 1080 Ti at a total GPU cost below U$ 6000.00 before taxes and shipping.[5]

In this study, the best model, which achieved an *AP* of 94.6% on the test set, is a significant milestone in the detection of external defects using TCVS. Comparing the results of this study with those of other works is challenging due to the lack of open datasets for benchmarking algorithms in this field. Published papers on computer vision for agriculture and food usually use of their own datasets and use performance metrics based on the properties of their dataset and model, which makes comparisons difficult (Kamilaris and Prenafeta-Boldú,

[5] Quotation acquired from the NVIDIA website https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/ on 23/01/2019.

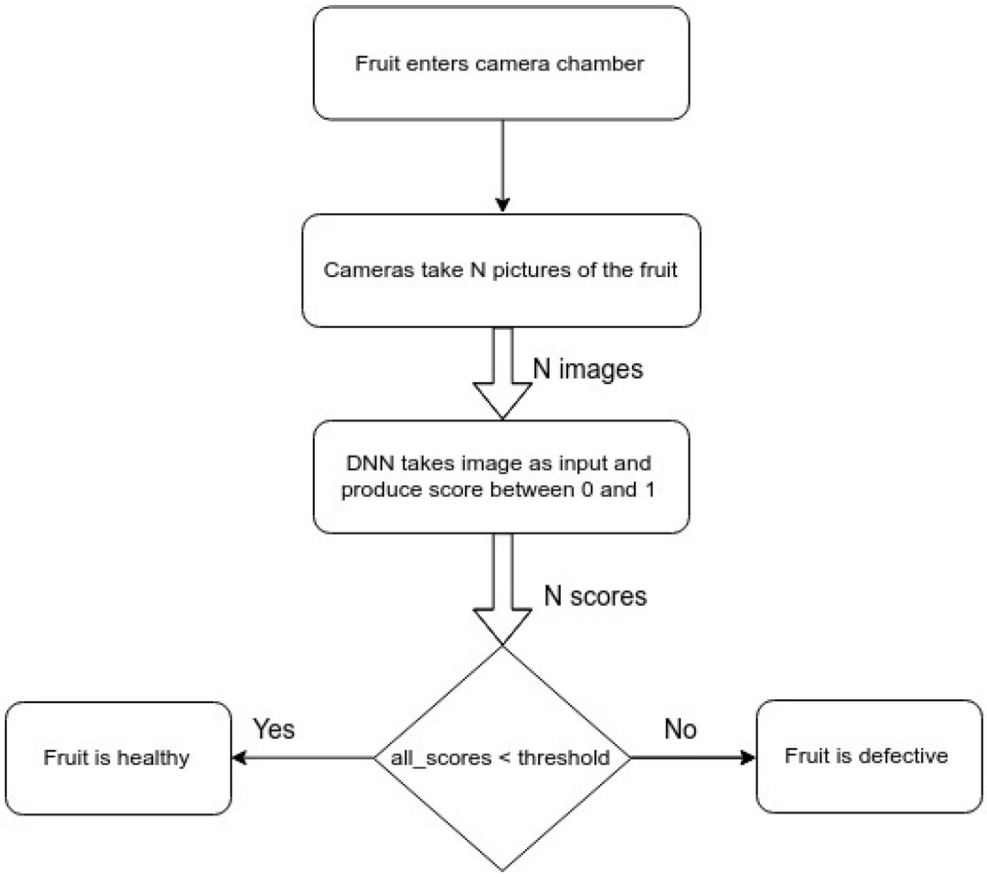

Fig. 12 – Decision flowchart for detecting external defects.

2018). Although not a proper comparison, considering the research studies that inspired us to investigate using state-of-the-art machine learning models for defect detection (Zhang et al., 2014), most works based on TCVS achieve an accuracy below 90.0% on small datasets with only a few hundred or a few thousand images. Furthermore, even the best models were tested only on common external defects, which are only a subset of the full external defect class.

Although the lack of reference datasets makes it difficult to make a fair comparison between the proposed algorithm and previously published works, it can be considered that the size of the compiled dataset, the lack of censorship and the study's fidelity to real-life conditions make the results reliable. Moreover, no other known open source dataset for studying external defect detection in any type of food using TCVS exists that is as large and comprehensive as the one compiled during this study. Contrary to previous research, no restrictions were imposed on which defects could appear in the data. As a final comment, but no less important, the Dirac delta posterior score distribution of the healthy class tells us that the stems/calyxes were not mistaken for external defects. This fact constitutes a milestone achievement because being able to distinguish between stems/calyxes and external defects is considered an open problem (Zhang et al., 2018). Thus, this study achieved state-of-the-art performance on the problem of external defect detection.

Based on the posterior distribution of the scores, the problem of external defect detection could be formulated as an unsupervised anomaly detection. Due to the variety of external defects, working with this particular framework could bring better results. For example, healthy samples could train a deep autoencoder or deep one-class classifier. In particular, the latter suggestion has some similarities with the proposed binary classifier. The proposed model uses healthy samples to build an almost ideal healthy score distribution, and the defects are mainly used to determine where to establish the final score threshold. Along the same line of thought, one-class classifiers have training methodologies in which they use not only the target class but also the outliers (Khan & Madden, 2010). Nevertheless, a binary classifier is a wise choice because the model and its score metrics are well established in the literature and because it is simple to understand, train and use while exhibiting state-of-the-art performance.

As a last observation, the model had no prior knowledge about tomato properties based on manually designed features or any other type of preprocessing. Therefore, the model

should achieve similar performance levels when applied to other foods if a sufficiently large dataset is compiled for the new task using the methodology reported in this paper.

5. Conclusions

In this paper, a deep neural network binary classifier was proposed for detecting external defects in tomatoes. The model overcame the challenges of the high variance in the poorly sampled defective class by learning an almost ideal representation for the healthy tomatoes. The proposed model achieved a milestone for the external defect detection problem because it did not confuse stems or calyxes with external defects and had 94.6%AP in a big and uncensored dataset. Overall, the results constitute yet another computer vision agricultural application for which deep learning achieved state-of-the-art performance. The results can likely be further improved by using anomaly detection techniques such as deep autoencoders and one-class classifiers, although the effect on performance will depend on the increase in model complexity. The proposed model can be extended to other foods if the corresponding datasets are created because the model itself has no prior knowledge about tomatoes and because no manual feature engineering was applied to improve the performance.

Acknowledgements

The authors acknowledge the Conselho Nacional de Desenvolvimento Científico e Tecnológico, process number 424016/2016-8, for the financial support to this work. The authors thank Zé Amparo Hortifruti LTDA and its employees for allowing us to use its sorting machine and teach us how they inspect tomatoes for defect detection. Special thanks go to Francisco José Soares de Sousa.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.biosystemseng.2019.12.003.

REFERENCES

Arnal Barbedo, J. G. (2019). *Plant disease identification from individual lesions and spots using deep learning*, *180* pp. 96–107). https://linkinghub.elsevier.com/retrieve/pii/S1537511018307797.

Blasco, J., Aleixos, N., & Moltó, E. (2007). Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm. *Journal of Food Engineering*, *81*(3), 535–543. http://linkinghub.elsevier.com/retrieve/pii/S0260877406007242.

Boyd, K., Eng, K. H., & Page, C. D. (2013). Area under the precision-recall curve: Point estimates and confidence intervals. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Advanced information systems engineering*, *7908* pp. 451–466). Berlin, Heidelberg: Springer Berlin Heidelberg. http://link.springer.com/10.1007/978-3-642-40994-3_29.

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, *31*(4), 299–315.

Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., et al. (2017). *State-of-the-art speech recognition with sequence-to-sequence models*. arXiv: 1712.01769 [cs, eess, stat] http://arxiv.org/abs/1712.01769.

Cruz, A., Ampatzidis, Y., Pierro, R., Materazzi, A., Panattoni, A., De Bellis, L., et al. (2019). *Detection of grapevine yellows symptoms in vitis vinifera l. with artificial intelligence*, *157* pp. 63–76). https://linkinghub.elsevier.com/retrieve/pii/S0168169918312353.

Cruz, A. C., Luvisi, A., De Bellis, L., & Ampatzidis, Y. (2017). *X-FIDO: An effective application for detecting olive quick decline syndrome with deep learning and data fusion*, *8*. http://journal.frontiersin.org/article/10.3389/fpls.2017.01741/full.

Cubero, S., Lee, W. S., Aleixos, N., Albert, F., & Blasco, J. (2016). 'Automated systems based on machine vision for inspecting citrus fruits from the field to postharvest—a review'. *Food and Bioprocess Technology*, *9*(10), 1623–1639. http://link.springer.com/10.1007/s11947-016-1767-1.

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *'Proceedings of the 23rd international conference on Machine learning - ICML '06'* (pp. 233–240). Pittsburgh, Pennsylvania: ACM Press. http://portal.acm.org/citation.cfm?doid=1143844.1143874.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer vision and pattern recognition, 2009. CVPR 2009* (pp. 248–255). IEEE Conference on', IEEE.

dos Santos Ferreira, A., Matte Freitas, D., Gonçalves da Silva, G., Pistori, H., & Theophilo Folhes, M. (2017). Weed detection in soybean crops using ConvNets. *Computers and Electronics in Agriculture*, *143*, 314–324. https://linkinghub.elsevier.com/retrieve/pii/S0168169917301977.

Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*, *151*, 72–80. http://linkinghub.elsevier.com/retrieve/pii/S1537511016301465.

Folch-Fortuny, A., Prats-Montalbán, J., Cubero, S., Blasco, J., & Ferrer, A. (2016). VIS/NIR hyperspectral imaging and N-way PLS-DA models for detection of decay lesions in citrus fruits. *Chemometrics and Intelligent Laboratory Systems*, *156*, 241–248. http://linkinghub.elsevier.com/retrieve/pii/S0169743916301058.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, *15*(6), 455–469. http://linkinghub.elsevier.com/retrieve/pii/0031320382900243.

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

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). https://arxiv.org/pdf/1512.03385.pdf.

Huang, H., Yu, P. S., & Wang, C. (2018). *An introduction to image synthesis with generative adversarial nets*. arXiv:1803.04469 [cs] http://arxiv.org/abs/1803.04469.

Hu, M.-h., Dong, Q.-l., Liu, B.-l., & Malakar, P. K. (2014). The potential of double K-means clustering for banana image segmentation: Image segmentation on banana. *Journal of Food Process Engineering*, *37*(1), 10–18. https://doi.org/10.1111/jfpe.12054.

ImageNet large scale visual recognition competition (ILSVRC). (2017). http://www.image-net.org/challenges/LSVRC/.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, *37*.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. https://linkinghub.elsevier.com/retrieve/pii/S0168169917308803.

Khan, S. S., & Madden, M. G. (2010). A survey of recent trends in one class classification. In L. Coyle, & J. Freyne (Eds.), *Artificial intelligence and cognitive science, 6206* pp. 188–197). Berlin, Heidelberg: Springer Berlin Heidelberg. http://link.springer.com/10.1007/978-3-642-17080-5_21.

Knuth, K. H. (2006). *Optimal data-based binning for histograms*. arXiv:physics/0605197 http://arxiv.org/abs/physics/0605197.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://www.nature.com/articles/nature14539.

Li, J., Chen, L., Huang, W., Wang, Q., Zhang, B., Tian, X., et al. (2016). Multispectral detection of skin defects of bi-colored peaches based on vis–NIR hyperspectral imaging. *Postharvest Biology and Technology*, *112*, 121–133. http://linkinghub.elsevier.com/retrieve/pii/S0925521415301514.

Li, J. B., Huang, W. Q., & Zhao, C. J. (2015). Machine vision technology for detecting the external defects of fruits — a review. *The Imaging Science Journal*, *63*(5), 241–251.

Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, *267*, 378–384. https://linkinghub.elsevier.com/retrieve/pii/S0925231217311384.

Minh, D. H. T., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F., et al. (2017). *Deep Recurrent Neural Networks for mapping winter vegetation quality coverage via multi-temporal SAR Sentinel-1*. arXiv:1708.03694 [cs] http://arxiv.org/abs/1708.03694.

Mishkin, D., & Matas, J. (2015). *All you need is a good init*. arXiv:1511.06422 [cs] http://arxiv.org/abs/1511.06422.

Murphy, K. (2012). *Machine learning: A probabilistic perspective* (4th ed.). MIT Press.

Rußwurm, M., & Körner, M. (2017). *'MULTI-TEMPORAL land cover classification with long short-term memory neural networks', ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-1/W1* (pp. 551–558). http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-1-W1/551/2017/.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556 https://arxiv.org/abs/1409.1556.

Wang, J., Nakano, K., Ohashi, S., Kubota, Y., Takizawa, K., & Sasaki, Y. (2011). Detection of external insect infestations in jujube fruit using hyperspectral reflectance imaging. *Biosystems Engineering*, *108*(4), 345–351. http://linkinghub.elsevier.com/retrieve/pii/S1537511011000183.

Xiao-bo, Z., Jie-wen, Z., Yanxiao, L., & Holmes, M. (2010). In-line detection of apple defects using three color cameras system. *Computers and Electronics in Agriculture*, *70*(1), 129–134. http://linkinghub.elsevier.com/retrieve/pii/S0168169909002051.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In *Advances in neural information processing systems* (pp. 3320–3328). https://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017). *Recent trends in deep learning based natural language processing*. arXiv:1708.02709 [cs] http://arxiv.org/abs/1708.02709.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.

Zhang, B., Gu, B., Tian, G., Zhou, J., Huang, J., & Xiong, Y. (2018). Challenges and solutions of optical-based nondestructive quality inspection for robotic fruit and vegetable grading systems: A technical review. *Trends in Food Science & Technology*, *81*, 213–231. https://linkinghub.elsevier.com/retrieve/pii/S0924224417307380.

Zhang, B., Huang, W., Li, J., Zhao, C., Fan, S., Wu, J., et al. (2014). Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research International*, *62*, 326–343. http://linkinghub.elsevier.com/retrieve/pii/S0963996914001707.