STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXV**, Number 2, 2020
DOI: 10.24193/subbi.2020.2.04

OVERVIEW OF RECENT DEEP LEARNING METHODS APPLIED IN FRUIT COUNTING FOR YIELD ESTIMATION

H. B. MUREŞAN, A. D. CĂLIN, AND A. M. COROIU

Abstract. This paper is an overview of the latest advancements of image recognition for fruit counting and yield estimation. Considering this domain is developing rapidly, we have considered the cutting-edge literature in the field, for the last 5 years, focused on the task of yield estimation by detecting and counting fruit in the tree canopy. This is a much more complex task than the classification of fruit post-harvesting, which has been more widely reviewed. Moreover, we identify the major challenges and propose the next steps for advancing this research field.

1. Introduction

This paper presents state of the art models and methods based on artificial intelligence for detecting fruits in orchards and on plantations. A system that can accurately and automatically detect and count fruit before harvest gives agricultural enterprises the ability to optimize and streamline their harvest process. Through a better understanding of the variability of yield across their farmlands, growers can make more informed and cost-effective decisions for labor allotment, storage, packaging, and transportation. While this process is performed manually, it involves a very high labour cost, which can be reduced using automated fruit counting computer vision systems.

Therefore, we analysed several papers tackling this issue using deep learning techniques. We selected papers of the latest 5 years of research in the field of fruit counting in tree canopies for yield estimation. We have searched for precision and digital agriculture publications using ACM digital library,

Received by the editors: 10 November 2020.

2010 *Mathematics Subject Classification.* 68T45.

1998 *CR Categories and Descriptors.* I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *Object recognition*; I.2.6 [**Artificial Intelligence**]: Learning – *Connectionism and neural nets; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – Intensity, color, photometry, and thresholding*.

Key words and phrases. smart-agriculture, deep learning, yield estimation, transfer learning, intersection over union, F1-score.

Science direct, IEEE and Google Scholar platforms, and used keywords such as "fruit counting", "deep learning", and "yield estimation".

(A) (B) (C)

(A) (B) (C)

FIGURE 1. Grapes, apricots and tomatoes in different lighting conditions, backgrounds and occlusions

We have identified review papers focused on agri-tech that present a very broad overview of applications of deep learning in various fields of agriculture [4]. Other papers present the methods overview for the specific domain of fruit or image classification [6]. However, most papers deal with post-harvesting classification of fruits for packaging or similar purposes. In this review, we aim to narrow the focus specifically on the task of pre-harvest yield estimation of fruit in an orchard, which is a problem of localising and counting fruit in the tree canopy, with different backgrounds, occlusions and lighting conditions. This enables farmers to estimate their yield and plan resources required for harvesting, storage, processing/packaging accordingly. Furthermore, with accurate computer vision technology, the harvesting could be performed by robots, with increased efficiency.

The main goals of the reviewed papers presented here are:

- accurately predicting the number of fruits in an image under various illumination conditions and different levels of fruit occlusion in the

tree canopy (this is to optimise and streamline the harvesting process and the fruit distribution for commercialisation or processing)
- reducing the labor costs required to perform yield estimation or harvest (representing 50-70% of the total costs [22])
- correctly estimating the size of the fruits from images
- adapting existing classification techniques for automatic robot harvesting or low-power devices (mobile phone).

The paper is structured as follows: firstly, we will present the methods identified in the papers, then we will discuss the datasets particularities used for training and testing performance. Next, we present results and the conclusions, as well as the challenges that remain in this research area and possible steps that can be taken to address them.

2. State of the art

2.1. **Methods.** The main methods we have analysed can be split into three main categories:

- **Deep learning with simple pre-processing** [13] - These methods involve the use of convolutional neural networks (CNNs) for the task of object detection (fruits in this case). They apply basic image pre-processing, such as rotations, vertical/horizontal flips, random zoom level, image cropping and colorspace conversion to augment the training dataset. Typically such methods require a large number of images to train the model.
- **Deep learning with complex pre-processing** [11, 2, 1, 5] - In addition to the previous category, complex image pre-processing, involving filtering features (such as background), colorspace changing and colorband isolation, or applying other intelligent algorithms for feature selection, is applied to enhance the training dataset before running a CNN on it.
- **Transfer learning** [21, 3, 23] - This method uses existing trained models and replaces the top layers to retrain them on a particular dataset. As opposed to previous methods, it can yield good results with reduced input data.

A novel approach for counting the number of tomato fruits is presented by Rahnemoonfar [13]. The authors proposed an approach based on a convolutional neural network (CNN) for object counting (different scales and occlusions) rather than area calculation (average pixel coverage), incorporating a modified Inception-ResNet-A module and a scaling module (17×17 to 8×8). The latter involves calculating the total pixel coverage of the target fruit and

then divide it to an average pixel coverage of a single fruit. The former typically requires a detection step, but it is not influenced by image scale and occlusion. Furthermore, as the proposed goal was only to provide a fruit count per image, rather than the actual location of fruits, the bounding box proposal was skipped, decreasing the inference time. The proposed model was a CNN which incorporated modified Inception-ResNet-A modules [18] and a 17×17 to 8×8 reduction module. The authors also use convolutional layers with large kernels towards the top of the network, to extract large scale features. The output of the network is given by a fully connected layer with a single output, the number of predicted fruits.

The model presented was compared with an area-based method, in terms of accuracy and speed. The accuracy for predicting the number of fruits in an image was defined as follows:

$$\left(1 - \frac{|predicted\_count - actual\_count|}{actual\_count}\right) \times 100 \tag{1}$$

In paper Mao [11], the authors presented a novel approach for detecting cucumbers. On top of the common difficulties of object detection, cucumbers easily blend in with the background due to their green color, further complicating the task. In order to compensate for this, the authors devised a four-component system that aimed to improve the accuracy of the detection, containing:

(1) ***Color component selection:*** From a total of 15 color components from 5 color spaces (RGB, HSI, YCbCr, Lab, YIQ), this extracts the top 3 that make it the easiest to differentiate the cucumber from the background. For this, the I-RELIEF [17] algorithm was used, which calculates weights for given features. The most relevant color bands selected were red and green from RGB color space and the intensity from the HSI color-space.

(2) ***Background pre-processing:*** The green component (from the RGB representation) was smoothed using a 3×3 median filter. Afterwards, the OTSU [20] method was applied to obtain a filtered background, and, finally, the Maximally Stable Extremal Regions [12] were used to eliminate the leaves from the background.

(3) ***Deep learning-based feature extraction:*** The authors segmented the original input image into small areas, applied pixel interpolation, and used LeNet5 [9] model, which support input data size 32×32. Each color component was passed through a separate instance of LeNet5 and fused, creating a multipath CNN. For this step, the authors elected to segment the original input image into small areas

and apply pixel interpolation on the image. The result of the interpolation is a collection of 32×32 areas. As noted in the paper, using the VGG [16] or AlexNet [7] models would require resizing the areas with a factor of 20, which could produce image distortion. Thus, the LeNet5 [9] model was used, which has a 32×32 input size, so no resizing is necessary. To fully make use of the selected color components, each one was passed through a separate instance of LeNet5, fusing the output of the last layers of the models, creating a multipath convolutional neural network (MPCNN).

(4) ***Cucumber region detection:*** The feature maps produced by the convolutional neural networks were merged and a Principal Component Analysis [8] algorithm was applied to reduce their dimensionality. The classification was done by a Support Vector Machine approach.

A fruit yield estimation pipeline that can map fruit counts from an input image is created in Chen [2]. The pipeline includes:

(1) ***Data labeling:*** makes use of a crowd-sourcing web platform for data labeling. Images for labeling are subdivided into several windows, each window annotated by 3 different users.

(2) ***Blob extraction:*** trains a fully convolutional network to extract candidate regions (blobs). Input is an image $h \times w \times 3$ and the output a score tensor $h \times w \times n$ where n is the object class (the probability that the pixel may contain a fruit or not, using a softmax function).

(3) ***Fruit counting:*** uses a second CNN algorithm trained for counting fruit in each region. For each blob, the output is a number representing the fruit count. The fine-tuning process involves running the blob detection network on the training images to obtain segmented images and bounding boxes, which are resized to 128×128. Next, ground truth counts are associated with the count network.

(4) ***Count mapping:*** maps a linear regression model between fruit count estimates and final fruit count. This trains a linear regression to intermediate count estimates with human-generated labels as ground truth, minimising the loss function between the count network and the blob network.

This pipeline is evaluated using two datasets (oranges in day and apples at night) and human-generated count and labeling for ground truth. For each image x_i, we have the actual number of fruit z_i and the human ground truth $\widetilde{z_i}$. If $f(x_i)$ is the algorithm generated count, the problem is to minimise the l^2 error:

$$l^2 = \sqrt{\sum_{i=1}^{n} (f(x_i) - \widetilde{z_i})^2} \tag{2}$$

Paper Bargoti [1] is focused on developing an image processing framework for fruit detection and counting using orchard image data. They use a general-purpose image segmentation approach, including two feature learning algorithms: multiscale multilayered perceptrons (MLP) and convolutional neural networks (CNN). These networks were extended by including metadata which correlates with appearance variations and/or class distributions. Further, the authors utilised watershed segmentation (WS) and circular Hough transform (CHT) algorithms to process image pixels, and then detect and count fruits. Finally, the counts from each row in the orchard were summed up and compared with the total post-harvest counts (done by a grading and counting machine).

In paper Kang [5], authors developed a real-time apple detector based on the LedNet architecture. The presented model uses the Feature Pyramid Network (FPN) [10] and Atrous Spatial Pyramid Pooling (ASPP) algorithms. The one-stage model was chosen by the authors as it offers the same, or superior performance to two-stage detectors, but with fewer network parameters. The FPN used in LedNet fuses feature maps at three levels of downsampling (1/8, 1/16, 1/32) to increase the model's capability of detecting objects at various ranges. The ASPP technique was employed to process multi-scale features. The custom ResNet backbone was a light-weight version of a typical ResNet architecture to reduce the inference time on an embedded system, such as an autonomous robot.

One study, Xiang [21], presents fruit image classification using a lightweight neural network MobileNetV2 [15] (pretrained using ImageNet dataset, for feature extraction). Here, the top layer was replaced with a conventional convolution layer (conv2d) and a Softmax classifier (for feature classification into 5 classes of fruits) [21]. They also applied dropout to the new-added conv2d at the same time to reduce overfitting. The new model was trained and fine-tuned in two stages, using Adam optimizer of different learning rate, and batch size of 64. TensorFlow 1.14 stable was used for performance evaluation. Compared to others, this method can be deployed in low-power and limited-computing devices such as mobile phones.

Another study uses machine vision to accurately identify and localise grapes and apples, Fourie [3]. With the advantage of less time need for training and good performance with limited training data, transfer learning was used, based on deep convolutional neural network (DCNN). The authors pre-trained the

InceptionV3 model [19] on the ImageNet database, as a generic image feature extractor. Next, their classifiers were added to separate fruit and background features. Further, a final layer was replaced with one trained on a custom dataset of apple trees and vines, acting as a classification head, specialising the network with custom images training separately. For the last step of localising and counting fruit new layers were added. The output of the last convolution and the remaining spatial correlated outputs are pooled into a single high-dimensional feature vector, linked to the classification head. The localizer outputs a grid of confidence scores that indicate the fruit localisation in the image.

A more advanced study, focuses on detecting six different types of fruits: lime, lemon, apple, mandarin, tomato and orange in orchard settings, Yu [23]. The algorithms used are color based - Faster R-CNN (Convolutional Neural Networks, two stage region-based model) and SSD (Single Shot Detector, which is a region free method) applying transfer learning for fruit detection and counting.

2.2. **Datasets.** The studies we present make use of public datasets with fruits, where available (for example, ImageNet, COCO or dataset in [14]). Others have either collected their own datasets of specific images in orchards under various lighting conditions or used unspecific images from the web, retrieved with a web crawler.

In Rahnemoonfar [13], the lack of available public datasets with annotated images of tomatoes was handled using an interesting approach. They created their own, consisting of fully synthetic images. Their images were created as follows: firstly, they added green and brown circles for background, then applied a Gaussian filter to blur them, and finally, added red circles to simulate the tomato fruits. The authors also took into consideration variations in fruit size, scale, illumination and overlap, generating 24000 images for training and 2400 for validation, and using 100 real images for testing.

For study Mao [11], 225 images were collected from a cucumber planting base, in Shouguang, China, Shandong. The images were taken between 7 am to 10 am and from 3 pm to 6 pm to reduce the impact of illumination conditions. The images were resized from 4032×3016 to 1024×768.

The experiments in Chen [2] used two datasets that differ from the perspective of lighting conditions, occlusion levels, resolution and camera type. The first dataset contains orange images of size 1280×960 and was collected during the day, using a steady camera carried by a human operator at walking speed. The orange trees were in a nontrellis arrangement. There were 5000 images, labeled by 22 users. The second one, an apple dataset, collected at

night using an external flash, with images of size 1920 × 1200, from a utility vehicle at the speed of $1m/s$, with trees in a trellis arrangement.

In Bargoti [1], the dataset was collected in a commercial orchard of Kanzi and Pink Lady apple varieties, over a 0.5ha block of v-trellis arrangement of 17 rows, using a teleoperated vehicle, in daylight. The set contains more than 8,000 images of size 1232 ×1616. For experiments, random sub-sampling was used, dividing each image into 32 parts of size 308 × 202, manually annotated to binary fruit and non-fruit classes.

In Kang [5], 800 apple images were collected from an orchard in Qingdao, China, using a Kinect-v2, from a distance between 0.5 - 1.5 meters. An additional set of 300 images of apples in different scenes were collected to diversify the dataset. Due to the distance at which the images were taken, the apples would be represented largely in the small scale features. To avoid this imbalance, the authors applied a crop-and-resize algorithm. The labelling process was done with the help of a clustering region-based neural network. The model would extract multi-scale features, proposing potential regions of interest (ROI). The pixels of the ROIs were segmented using pixel-connection into independent candidate patches and bounding box coordinates were assigned to it. With the help of this model, the labelling of training data was done in only two days.

In Xiang [21] the ImageNet dataset was used, a large dataset of 1.4M images [14], and 1000 classes of web images having complex backgrounds: 3,670 images of 5 fruits collected from the Internet, including apples (633), bananas (898), carambola (641), guava (699) and kiwi (799). For the experiment, these were split into two subsets, 3,213 images for training and 457 images for validation. Images were adjusted to size 224×224 as required by the MobileNetV2 model.

In Fourie [3], authors used the ImageNet dataset, and collected their own datasets from an apple orchard and a vineyard. The apple set contaied 21 images (2000 × 3000 px), with various light conditions and view angles. Images were normalised to the same mean RGB intensity. 442 areas of interest were extracted for training and augmented through random transformations. 20% of the data was used for testing. The vineyard set was also split into a training set (95 images), and a testing set (52 images). Testing images were captured under different light conditions than those in the training set.

In Yu [23] authors used a Python Web Crawler to create a 2000 dataset of images. They augmented the set by rotations and RGB adjustments with different brightness and saturation, obtaining 2995 images (tomato 124, mandarin 301, orange 377, apple 680, lemon 605, lime 909).

2.3. **Results.** The models are often evaluated using testing data, measuring especially accuracy and processing time, but also loss, F1 score, Recall, True Positives, False Positives, and other specific measures defined by the authors.

As seen in Table 1, the proposed method by Rahnemoonfar [13] is significantly better in terms of accuracy than an area-based counting method. From a processing time perspective, the proposed method and the area-based are both much faster than manual counting.

Method	Avg accuracy	Stdev	Avg time/image
CNN based	91.03%	2.5	0.006
Area based	66.16%	7.9	0.05
Manual count	-	-	6.5

TABLE 1. Average accuracy and time over 100 test images of the methods studied in Rahnemoonfar [13]

The proposed method in Mao [11] was compared with one that uses an MPCNN with the red, green, blue channels and another that uses a single CNN for an RGB image. The best results were obtained by the model that was using the color bands selected by the color selection component (red, green, intensity), presented in Table 2. The metrics depicted are:

- correct recognition rate (CRR) - ratio between the number of true positive(TP) pixels and the total number of pixels in an image
- false recognition rate (FRR) - ratio between the number of false positive(FP) pixels and the total number of pixels in an image
- correct tot false ratio (CFR) - ratio between the CRR and FRR

Another observation was that the multi-path convolutional neural network performed strictly better than the regular convolutional network. This showed that applying late fusion instead of early fusion on multiple color components yields better results.

The results of Chen [2] in terms of reducing the l^2 error were obtained for the combination of blob + count + regression, values obtained are shown in

Methods	TP	FP	TP + FP	CRR	FRR	CFR
RGB + CNN + softmax	76,954	25,431	102,385	92.77%	24.84%	3.73
RGB + CNN + SVM	77,436	17,627	95,063	93.36%	18.54%	5.04
RGB + MPCNN + SVM	78,688	15,884	94,572	94.87%	16.80%	5.65
RGI + MPCNN + SVM	80,670	10,571	91,241	97.25%	11.59%	8.39

TABLE 2. The performance of the methods proposed in paper Mao [11]

Table 3. To evaluate pixel-wise accuracy, Intersection over Union and ROC curves were used. There were in total 7200 oranges in 71 images and 1749 apples in 21 images in the testing sets.

Model	l^2 error	Ratio Counted	Std Dev
Orange blob	16.9	0.935	15.6
Orange blob+regression	15.9	0.999	15.9
Orange blob+count	19.2	0.851	12.7
Orange blob+count+regression	**13.8**	**0.968**	13.5
Apple blob	46.5	1.475	24.9
Apple blob+regression	20.4	1.025	20.3
Apple blob+count	20.9	0.767	8.4
Apple blob+count+regression	**10.5**	**0.913**	7.7

TABLE 3. Count accuracy of the CNN proposed in Chen [2] for orange and apple set.

The metrics used for evaluating the proposed model in paper Kang [5] were the inference time, the number of parameters, F1-score, precision, recall, intersection over union (IoU), area under the curve - which was denoted as AP_m (where m is the threshold for IoU used to accept or reject proposed regions of interest). A comparison between the crop-and-resize augmentation process with the regular data augmentation method is presented in Table 4. As anticipated, the model performs poorly on medium and large fruits when augmenting images with rotates and color/brightness alteration due to the size imbalance. It can be noted that the model trained on data augmented with the crop-and-resize operation performed well regardless of the object size. Several popular architectures were compared with the proposed model (Table 5). The LedNet with the light-weight backbone performed just as well as the other much larger networks, while having the fastest inference time.

The results in the MLP network in Bargoti [1] improved after including the metadata, which can be observed in Table 6. Extending this with CNNs, the best pixel-wise F1-score of 0.791 was achieved, while the WS produced

Method	AP_{50}	AP_{small}	AP_{med}	AP_{large}	IoU
Crop and Resize	0.826	0.832	0.817	0.763	86.7%
Standard	0.797	0.818	0.778	0.652	78.3%

TABLE 4. The impact of the two augmentation methods utilised in Kang [5] on the performance of the LedNet model.

Method	AP_{50}	F1	Recall	Acc	IoU	Time	Params
LedNet(LW-Net)	0.826	0.834	0.821	0.853	86.3%	28 ms	7.4 M
LedNet (ResNet-101)	0.843	0.849	0.841	0.864	87.2%	46 ms	188 M
YOLOv3	0.803	0.803	0.801	0.82	84.2%	45 ms	248 M
YOLOv3 (Tiny)	0.782	0.783	0.776	0.796	82.4%	30 ms	35.4 M
Faster-RCNN (VGG)	0.814	0.818	0.814	0.835	86.3%	145 ms	533 M

TABLE 5. Evaluation the 5 different backbones used for the detector in Kang [5] .

the best results, with a detection F1-score of 0.861. Comparing the count estimates using CNN and WS with the base counting the squared correlation coefficient obtained $r^2 = 0.826$.

	Both	ms-MLP	CNN	Neither
ms-MLP	0.834	0.860	0.739	0.709
CNN	0.921	0.843	0.849	0.731

TABLE 6. Comparing ms-MLP and CNN approaches for fruit detection with image segmentation output and WS detection algorithm in Bargoti [1]

In Xiang [21], the classification accuracy obtained for the 5 fruits was 85.12%. To demonstrate its effectiveness on source-limited platforms, the models were deployed also on an Android smartphone (Honor 10 by Huawei). Through transfer learning, the new model was able to accelerate and optimise the learning process (MobileNetV2 as the best running time, as described in Table 7).

Apple classification accuracy was 98% (deciding if an area of interest contained an apple or background) in Fourie [3]. For the vine set, the network could correctly classify 99% of the testing areas of interest if they contain grape bunches. Next step would be to correlate counting and yield estimation.

Model	Training		Validation		Run
	Loss	Acc	Loss	Acc	(sec)
MobileNetV2	0.0109	0.9984	0.4719	0.8512	327
MobileNetV1	0.0335	0.9960	1.3527	0.7352	1618
InceptionV3	0.0071	0.9994	0.6322	0.8578	670
DenseNet121	0.0036	0.9994	0.3695	0.8906	7965

TABLE 7. Loss and Accuracy on the training and validation sets for the proposed models in Xiang [21].

The results for Yu [23] show that the accuracy of the model trained by Faster R-CNN was higher (at 89%) that for the model trained by SSD (at 82%). The average speed per image in seconds 8.21 (Faster-RCNN) and 6.70 (SSD) respectively (see Table 8).

Fruit	Orange	Mandarin	Lemon	Apple	Tomato
SSD	0.86	0.85	0.81	0.81	0.77
Faster R-CNN	0.91	0.90	0.89	0.89	0.87

TABLE 8. Accuracy comparison between the Faster R-CNN and SSD models on 5 different fruit classes, as described by Yu [23]

3. Discussion

3.1. **Results analysis.** Figure 2 presents the accuracy values of each reviewed model against the size of the dataset and number of classes. All but one of the studied papers (Bargoti [1], Mao [11], Rahnemoonfar [13], Kang [5], Fourie [3], Chen [2] and Yu [23]) implement deep learning detectors. These networks have the advantage of providing both class prediction as well as coordinates to locate the object in the image, making them better for fruit counting and yield estimation. The downside is that they also must be trained using data containing the same information, data which is scarce, they are more complex than classifiers and thus require more training resources.

The model in Xiang [21] is a classifier, which is simpler to train and deploy, compared to a detector, however it provides only class predictions. Despite this, the model was outclassed by all other works. The reason behind this is very likely to be the dataset obtained by scraping images from the Internet, which would contain a high degree of variance and potentially too few samples per variant.

The average accuracy across these works is 91.98%. The datasets of images used by authors range in size (from 168 to 24000), depending on the method used, and high accuracy has been obtained for low and high number of testing images, in association with the proper model.

One observation is that models that were trained to classify or detect 5 or more classes of fruits have not achieved an accuracy over 90% (89% and 85.10%, respectively).

3.2. **Challenges imposed by datasets.** Although popular datasets (ImageNet, COCO) are often used for transfer learning, they do not contain real field image data of occluded fruits and various lighting conditions throughout

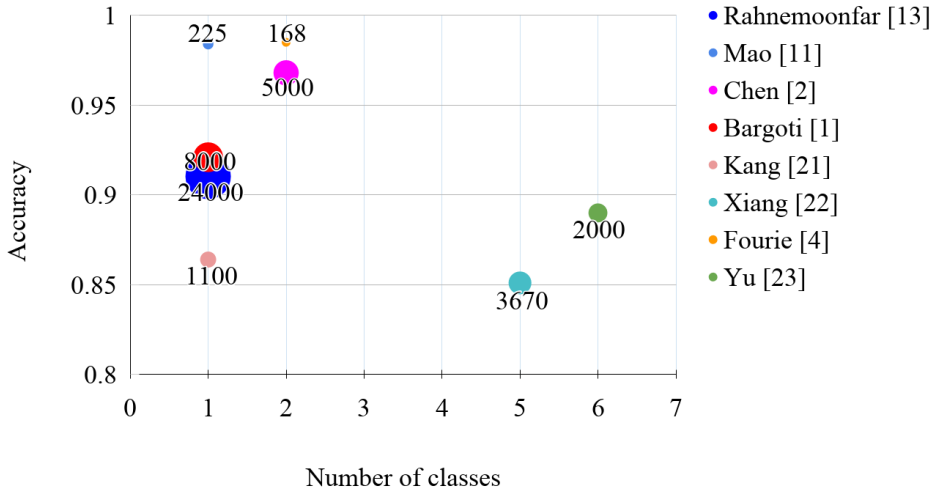

FIGURE 2. Accuracy in relation with the Number of Classes and Dataset Size: Chart that helps visualize the best performing models from each of the studied papers. We took into account the accuracy, number of classes and images that the model was trained on.

the day. As such, a big disadvantage is that every research group focused on fruit detection needs to create their own dataset to suit their needs. Creating such a dataset is both a difficult and time cosuming process for the following reasons:

- The collected images must contain samples in each lighting conditions under which the model is expected to perform (e.g. sunny, rain, cloudy, early morning, late evening)
- The images should capture as many variations of the targeted fruit classes as possible (each fruit class contains different amounts of variations in shape, size and colour, and in some cases manifest visual deformations or defects)
- The background of the images must be relevant to the desired application (e.g. if the model should detect ripe fruits on trees in orchards, the background should include tree branches and leaves)
- The dataset must be annotated with the class name for classification, to which bounding box annotations must be added for detection, an overall time consuming process if done manually

A publicly available dataset with such images would be advantageous to achieve shorter research times. Alternatively, it has been shown that synthetic images do not degrade the performance of trained models and are much easier to create than real images. Perhaps a combination of a synthetic image generation algorithm together with a small dataset of real images for fine-tuning can serve as a starting point.

Another subject of research is the impact of illumination conditions on images due to the position on the globe. Specifically, if the images are taken in an area close to the Ecuador during the daytime, they will be differently lit than images taken during the daytime in areas closer to the poles. Thus, the goal is to investigate whether a model trained on images from one of these areas performs equally well on images from the other area.

3.3. **Model optimisation.** One more possible direction is increasing of the accuracy of models with a new approach based on the collected methods already existed in literature. It was proven that convolutional neural networks achieve better performance than the alternative methods in tasks of fruit detection. However, there are tasks that are still challenging for this class of algorithms, detecting partially occluded fruits or correctly counting grouped fruits being among the more frequent ones.

Since the majority of reviewed papers proposed models that use images or frames extracted from videos, the area of video analysis remains largely unexplored. The LSTM architecture is well suited to process time series, and videos can be seen as a series of frames. This approach could further address the aforementioned issues as the video could cover trees/plants from multiple angles.

4. Conclusions and next steps

In this paper, we presented an overview of the latest research involving deep learning for fruit yield estimation using orchard images. Some very good results (up to 97% accuracy) were obtained using simple or complex pre-processing techniques and large data for CNN training Mao [11]. Some successful attempts have used transfer learning for limited training data, proving that there is good potential even for low-resource platforms to be used Fourie [3], Xiang [21]. Some studies focus on mapping and adjusting, based on the manual count, the algorithm generated count Bargoti [1], Rahnemoonfar [13], with specific applications in an orchard (results of up 96.8% accuracy in counting Chen [2]).

We have also highlighted the limitations of these studies and possible directions of research, derived from challenges posed by the need to use real

field data with various fruits, and the need improve the models by increasing accuracy of detection for a larger variety of fruits.

Overall, the results show a very good potential for further research and improvement up to the use in practical settings for pre-harvest yield estimation and designing harvesting robots.

This paper is a very useful initial step for a more elaborate project, the role of the current paper is to set a ground from we can develop particular approaches.

References

[1] Bargoti, S., and Underwood, J. P. Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics 34*, 6 (2017), 1039–1060.

[2] Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C. J., and Kumar, V. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters 2*, 2 (2017), 781–788.

[3] Fourie, J., Hsaio, J., and Werner, A. Crop yield estimation using deep learning. In *7th Asian-Australasian Conference on Precision Agriculture* (2017), pp. 1–10.

[4] Kamilaris, A., and Prenafeta-Boldú, F. X. Deep learning in agriculture: A survey. *Computers and electronics in agriculture 147* (2018), 70–90.

[5] Kang, H., and Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture 168* (2020), 105108.

[6] Koirala, A., Walsh, K. B., Wang, Z., and McCarthy, C. Deep learning–method overview and review of use for fruit detection and yield estimation. *Computers and Electronics in Agriculture 162* (2019), 219–234.

[7] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM 60*, 6 (May 2017), 84–90.

[8] Lavania, S., and Matey, P. S. Novel method for weed classification in maize field using otsu and pca implementation. In *2015 IEEE International Conference on Computational Intelligence Communication Technology* (2 2015), pp. 534–537.

[9] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (6 1998), 2278–2324.

[10] Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (7 2017), pp. 936–944.

[11] Mao, S., Li, Y., Ma, Y., Zhang, B., Zhou, J., and Wang, K. Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. *Computers and Electronics in Agriculture 170* (2020), 105254.

[12] Nistér, D., and Stewénius, H. Linear time maximally stable extremal regions. In *Computer Vision – ECCV 2008* (Berlin, Heidelberg, 2008), D. Forsyth, P. Torr, and A. Zisserman, Eds., Springer Berlin Heidelberg, pp. 183–196.

[13] Rahnemoonfar, M., and Sheppard, C. Real-time yield estimation based on deep learning. In *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II* (2017), J. A. Thomasson, M. McKee, and R. J. Moorhead, Eds., vol. 10218, International Society for Optics and Photonics, SPIE, pp. 59 – 65.

[14] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision 115*, 3 (2015), 211–252.

[15] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 4510–4520.

[16] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556* (2014).

[17] Sun, Y. Iterative relief for feature weighting: Algorithms, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 6 (6 2007), 1035–1051.

[18] Szegedy, C., Ioffe, S., and Vanhoucke, V. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR abs/1602.07261* (2016).

[19] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2818–2826.

[20] Wang, L., and Duan, H.-c. Application of otsu'method in multi-threshold image segmentation [j]. *Computer Engineering and Design 11* (2008), 2844–2845.

[21] Xiang, Q., Wang, X., Li, R., Zhang, G., Lai, J., and Hu, Q. Fruit image classification based on mobilenetv2 with transfer learning technique. In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering* (2019), pp. 1–7.

[22] Yang, R., Wu, M., Bao, Z., and Zhang, P. Cherry recognition based on color channel transform. In *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science* (2019), pp. 292–296.

[23] Yu, H., Song, S., Ma, S., and Sinnott, R. O. Estimating fruit crop yield through deep learning. In *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies* (2019), pp. 145–148.

Department of Computer Science, Faculty of Mathematics and Computer Science, Babeş-Bolyai University, 1 Kogălniceanu St., 400084 Cluj-Napoca, Romania

Email address: {horea.muresan, alinacalin, adrianac}@cs.ubbcluj.ro