

Article

Using YOLOv3 Algorithm with Pre- and Post-Processing for Apple Detection in Fruit-Harvesting Robot

Anna Kuznetsova, Tatiana Maleva and Vladimir Soloviev * 

Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, 38 Shcherbakovskaya, 105187 Moscow, Russia; anakuznetsova@fa.ru (A.K.); tvmaleva@fa.ru (T.M.)

* Correspondence: vsoloviev@fa.ru

Received: 19 June 2020; Accepted: 7 July 2020; Published: 14 July 2020



Abstract: A machine vision system for detecting apples in orchards was developed. The system was designed to be used in harvesting robots and is based on a YOLOv3 algorithm with special pre- and post-processing. The proposed pre- and post-processing techniques made it possible to adapt the YOLOv3 algorithm to be used in an apple-harvesting robot machine vision system, providing an average apple detection time of 19 ms with a share of objects being mistaken for apples at 7.8% and a share of unrecognized apples at 9.2%. Both the average detection time and error rates are less than in all known similar systems. The system can operate not only in apple-harvesting robots but also in orange-harvesting robots.

Keywords: machine vision; fruit harvesting robot; fruit yield counting robot; YOLO

1. Introduction

As a result of intensification, mechanization, and automation, agricultural productivity has increased significantly. In general, in developed countries, the number of people employed in agriculture decreased by 80 times during the 20th century. Nevertheless, manual labor is the main component of costs in agriculture, reaching 40% of the total value of vegetables, fruits, and cereals grown [1,2].

Horticulture is one of the most labor-intensive sectors of agriculture: the level of automation in horticulture is about 15%, fruit harvesting is done manually, and crop shortages reach 50%. At the same time, as a result of urbanization, every year, it is becoming increasingly difficult to recruit seasonal workers for the harvest [3]. It is evident that the widespread use of robots can bring significant benefits in horticulture, increase labor productivity, reduce the share of heavy manual routine harvesting operations, and reduce crop shortages.

Fruit-picking robots have been developing since the late 1960s. Yet to this day, not a single prototype has entered the phase of practical use in orchards, since the cost of production of such robots reaches several hundred thousand dollars, even though the fruit-harvesting speed is extremely low, and the share of unhandled apples left on trees remains very high. To a large extent, the low speed of fruit harvesting and the high percentage of unhandled fruits left on trees are due to the insufficient quality of machine vision systems used in fruit-picking robots [4,5].

Recently, many neural network models have been trained to recognize apples. However, computer vision systems based on these models in existing prototypes of harvesting robots do not detect darkened apples and apples with a lot of overlapping leaves and branches, as well as green apples on a green background, take yellow leaves as apples, etc.

To solve these problems, in this paper, it is proposed to use the YOLOv3 algorithm for detecting apples on trees in orchards, with special pre- and post-processing of images taken by the cameras placed on the manipulator of the harvesting robot.

This paper is an extended version of [6]. The literature review has been expanded. The overall research methodology has been significantly refined, including a more detailed description of apple-harvesting robot design, image acquisition, and apple detection quality evaluation. All the proposed pre- and post-processing techniques, as well as all the algorithms' parameters, were described in detail. The number of evaluated apple detection quality metrics was broadened significantly, and discussion on apple detection quality was expanded. An additional procedure for apple detection in far-view canopy images has been proposed. The results of apple detection by using the proposed technique of combining the YOLOv3 algorithm with the pre- and post-processing procedures are compared with the standard YOLOv3 algorithm without additional procedures and with other modern algorithms (YOLOv3-Dense, DaSNet-v2, Faster-RCNN, LedNet). Besides, the possibility of applying the proposed technique to other spherical fruit detection (oranges and tomatoes) is discussed.

The remainder of the paper is structured as follows. This section reviews related works on apple detection in orchards using intelligent algorithms. Section 2 presents our technique of images pre- and post-processing for improving the apple detection efficiency of the YOLOv3 algorithm. The results showing the average apple detection time, the share of objects mistaken for apples, and a share of unrecognized apples better than in all known similar systems are presented in Section 3 and discussed in Section 4.

1.1. Color-Based Fruit Detection Techniques

The efficiency and productivity of harvesting robots are primarily determined by algorithms used to detect fruits in images. In various prototypes of such robots, various recognition techniques based on one or more factors were used.

The set color threshold can be used for each pixel in the image to determine if this pixel belongs to the fruit. Since color detection depends very much on the lighting conditions, different color spaces other than RGB are usually used: HIS, CIE L*a*b, LCD, and their combinations [7–9]. In [10,11], this approach showed a 90% share of correctly recognized apples, and in [12], this approach showed a 95% share of correctly recognized apples, although on very limited datasets (several dozen images).

Of course, color-based apple detection works well in the case of red apples, but usually, it does not provide satisfactory quality for green apples [13]. To solve the problem of green fruit detection, many authors combine image analysis in the visible and infrared spectra [11,14–16]. For example, in [16], a 74% fraction of correctly detected apples (accuracy), obtained by combining analysis of the visible and infrared spectra, is compared with 66% of correctly detected fruits based on analysis of the visible spectrum only and with 52% accuracy based on analysis of the infrared spectrum only.

The apparent advantage of fruits detection by color is the ease of implementation, but this method detects green and yellow-green apples very poorly. In addition, clusters of red apples merge into one giant “apple”, and this leads to incorrect determination of the apple bounding box coordinates.

Thermal cameras are quite expensive and inconvenient in practical use, since the difference between apples and leaves is detected only when shooting is made within two hours after dawn.

1.2. Shape-Based Fruit Detection Techniques

To detect spherical fruits such as tomatoes, apples, and citrus, fruit-recognition algorithms based on the analysis of geometric shapes could be used. The main advantage of the analysis of geometric shapes is the low dependence of the object recognition quality on the lighting level. To identify various shapes in images, Hough transformation allows representing the boundaries of objects in the form of circles (this is applicable for spherical fruits) [17,18]; Canny operator [19], and other techniques can be used as well.

In [20–22], modifications of the Hough circular transformation were used to improve the detection quality of fruits partially hidden by leaves or other fruits. In [23,24], algorithms for the detection of mature fruits based on the identification of convex objects in images were proposed. Systems based on such algorithms work very quickly, but complex scenes, especially with fruits overlapped by leaves or other fruits, are usually not recognized effectively by such systems.

To improve the quality of fruit detection in uncontrolled environments, which may deteriorate due to uneven lighting, the partial overlapping of fruits by other fruits and leaves, as well as other features, many researchers use various combinations of color and shape analysis algorithms. Simultaneous analysis of color, color intensity, perimeter shape, and orientation in [25] led to the correct detection of 90% peaches. The combination of color analysis and perimeter shapes analysis in [26] also gave 90% accuracy in detecting oranges. The authors of [27] combined the analysis of chromatic aberration and brightness to detect citrus fruits. This allowed detecting 86% of the fruits correctly.

The Open Source Computer Vision Library (OpenCV) implements a significant number of computer vision algorithms [28]. Many prototypes of fruit detection systems use various OpenCV algorithms: median filters, color separation, clipping by the color threshold, recognition of object boundaries using the Hough transformation, Canny and Sobel operators, etc. OpenCV algorithms were used in [29] to detect apples and in [30] to detect cherries.

The main advantages of the geometric shapes analysis are the high fruit detection speed and the low dependence of the quality of recognition of objects on the level of lighting [22]. However, detecting fruits by shape gives significant errors, since not only apples are round, but also gaps, leaf silhouettes, spots, and shadows on apples. Combining circle selection algorithms with subsequent pixel analysis is inefficient in terms of calculation speed.

1.3. Texture-Based Fruit Detection Techniques

Fruits photographed in orchards in natural conditions differ from the leaves and branches in texture, and this can be used to facilitate the separation of fruits from the background. Differences in texture play a particularly important role in fruit recognition in situations where the fruits are grouped in clusters or overlapped by other fruits or leaves. For example, in [31], apples were detected based on image texture analysis in combination with color analysis, and the proportion of correctly recognized fruits was 90% (on a limited dataset). In [32], apples were detected in images using texture analysis combined with geometric shapes analysis, and in [33,34], simultaneous analysis of texture, color, and shapes was performed, which made it possible to recognize correctly 75% of citrus fruits.

Detecting fruits by texture works only in close-up images with good resolution. The low speed of texture-based fruit detection algorithms and too-high proportion of undetected fruits lead to the inefficiency of practical use of this technique.

1.4. Early Stage of Using Machine Learning Algorithms for Fruit Detection

Machine learning methods have been used to detect fruits for a long time. The first robot designed to detect red apples against the background of green leaves using machine learning algorithms was developed in 1977 [35].

In [16], in order to detect green apples against the background of green leaves, K-means clustering was applied to a and b CIE L^*a^*b color space coordinates in the visible spectrum, as well as to image coordinates in the infrared spectrum with the subsequent removal of noise. This allowed the authors to correctly detect 74% of apples in the images from the test dataset. The use of linear classifiers and KNN classifiers to detect apples and peaches in the machine vision system was compared in [36], with both classification algorithms yielding similar accuracy at 89%. In [37], linear classifier has shown 80% accuracy of apple detection. The authors of [38] recognized apples, bananas, lemons, and strawberries in images, using KNN classifier and reported 90% accuracy. Applying KNN classifier to color and texture data allowed finding 85% of green apples in raw images and 95% in hand-processed images [39]. In [40], the SVM-based apple detection algorithm was introduced. This classifier balanced the ratio

between accuracy and recognition time, showing 89% of correctly detected fruits at an average apple detection time equal to 3.5 s. Using SVM for apple detection in [41] has shown an accuracy of 92%. It is very unusual that boosted decision trees were practically not used in fruit detection systems. In [42], the AdaBoost algorithm was used to recognize kiwi in orchards, which made it possible to achieve a 92% share of correctly detected fruits against branches, leaves, and soil. In [43,44], AdaBoost was applied to color analysis in order to automatically detect ripe tomatoes in a greenhouse, showing 96% accuracy. The search for examples of the use of modern algorithms such as XGBoost, LightGBM, and CatBoost for detecting fruits in images has not yielded results.

It should be noted that all the works mentioned in this section on the use of machine learning for fruit detection were tested on very limited datasets of several dozen images, which does not allow generalizing the results for practical use evaluation. For example, the authors of paper [41] published in 2017 reported a 92% accuracy of apple recognition using SVM, based on a test dataset of 59 apples.

1.5. Using Deep Neural Networks for Fruit Detection

Since 2012, with the advent of deep convolutional neural networks, in particular, AlexNet [45], machine vision, and its use for detecting various objects, including fruits in images, received an impetus in development. In 2015, VGG16 convolutional neural network was proposed as an improved version of AlexNet [46]. The machine vision system in the kiwi fruit-harvesting robot based on VGG16 was able to detect 76% of kiwi fruits during the field tests [47]. At the same time, the machine vision system also determined the fruits that the manipulator can reach (55% of them turned out to be reachable). In the field trials, 50.9% of 1456 kiwi fruits in the orchard were harvested, 24.6% were lost during the harvesting process, and 24.5% were left on the trees. The harvesting of one fruit took on average about 5 s. However, today, it is one of the fastest harvesting robots. VGG16 has also shown 90% accuracy in detecting kiwi fruits [48]. The authors published the dataset on which this model was trained in open access. A similar convolutional neural network was built in [49] and trained on the Fruits 360 dataset consisting of 4000 images of real fruits [50]. As a result, the share of correctly detected fruits in the test set of images was equal to 96.3%.

The next advancement in computer vision was R-CNN network [51] and its modifications: Fast R-CNN [52], Faster R-CNN [53], and Mask R-CNN [54], which made it possible to detect large numbers of objects, as well as determine their boundaries and relative positions. The ResNet network [55] based on Faster R-CNN won first place in the ImageNet Large-Scale Visual Recognition Challenge 2015, giving 96.4% correct answers.

In [56], using R-CNN, 86% of apple branches were correctly detected. In [57], Faster R-CNN was used to detect tomatoes, in [58], Faster R-CNN was used to recognize apples, mangoes, and almonds, and in [59], Faster R-CNN was used to recognize asparagus in images. In [57,58], the F1 score exceeded 90%, the authors of [59] reported F1 at 73%. The authors of [58] published an open-access ACFR-Multifruit-2016 dataset [60], on which their model was trained. This dataset contains 1120 images of apple crowns with fruits, 1964 images of mango crowns, and 620 images of almond crowns. In [61], Mask R-CNN was used to detect strawberries, and the F1 score exceeded 90%. The authors of [62] used Mask R-CNN for apple detection reporting on a test dataset of 368 apples in 120 images; the algorithm showed 97% precision and 95% recall. In [63], Mask R-CNN was applied to the analysis of three-dimensional images obtained from lidar. This allowed achieving 99% of correctly detected apples. The model was trained on a dataset consisting of three-dimensional images of 434 apples on 3 trees, and the test dataset included 1021 apples on 8 trees. In [64], Faster R-CNN was used to recognize green citrus fruits; 95.5% precision and 90.4% recall was achieved.

In 2016, a new algorithm was proposed—YOLO (You Look Only Once) [65]. Before this, to detect objects in images, classification models based on neural networks were applied to a single image several times, in several different regions, and/or on several scales. The YOLO approach involves a one-time application of one neural network to the whole image. The model divides the image into regions and immediately determines the scope of objects and probabilities of classes for each object.

The third version of the YOLO algorithm was published in 2018 as YOLOv3 [66]. The YOLO algorithm is one of the fastest, and it has already been used in robots for picking fruits. In [67,68], a modification of the YOLO model was proposed and applied to detect apples in images. The modification consisted of making the network tightly connected: each layer was connected to all subsequent layers, as the DenseNet approach suggests [69]. To assess the quality of fruit detection using the YOLOv3-Dense algorithm, *IoU* (Intersection over Union) was calculated and turned out to be 89.6%, with an average apple recognition time at 0.3 s. The use of the Faster R-CNN model in the same paper gave an 87.3% *IoU* with an average detection time at 2.42 s.

In [70], the DaSNet-v2 neural network was proposed, which (similarly to YOLO) determines objects in an image in a single pass, considering their overlapping. The *IoU* in this model built, especially for apple detection, turned out to be 86.3%.

The authors of [71] compared three algorithms: the standard Faster R-CNN, the proposed by them modification of the Faster R-CNN, and the YOLOv3 for the detection of oranges, apples, and mangoes. It turned out that the modification proposed by the authors reveals about 90% of the fruits, which is 3–4% better than the standard Faster R-CNN on the same dataset and at about the same level as the YOLOv3. However, the average recognition time for the YOLOv3 was 40 ms versus 58 ms for the modified Faster R-CNN and 240 ms for the standard Faster R-CNN.

It should be noted that the share of correctly recognized fruits and the share of errors of the first and second kind are given in an absolute minority of papers, and the *IoU* indicator is given only in a few works.

2. Materials and Methods

2.1. Apple Harvesting Robot Design

The Department of data analysis and machine learning of the Financial University under the Government of the Russian Federation, together with the Laboratory of machine technologies for cultivating perennial crops of the VIM Federal Scientific Agro-Engineering Center, is developing a robot for harvesting apples. The VIM Center develops the mechanical component of the robot, while the Financial University is responsible for the intelligent algorithms for detecting fruits and operating the manipulator for their picking. In the apple-harvesting robot we are developing, the machine vision system is based on a combination of two stationary Sony Alpha ILCE-7RM2 cameras with Sony FE24-240mm f/3.5-6.3 OSS lenses (Sony Electronics Inc., 16535 Via Esprillo, San Diego, CA 92127 USA) and one Logitech Webcam C930e camera (Logitech Europe S.A., EPFL-Quartier de l'Innovation, Daniel Borel Innovation Center, CH-1015 Lausanne, Switzerland) mounted on the second movable shoulder of the manipulator before the grip. The first two cameras take general far-view canopy shots for detecting apples and drawing up the optimal route for the manipulator to collect them, while the camera on the manipulator adjusts the position of the grip relative to the apple during the apple picking process. Therefore, it is essential to precisely detect apples both in far-view canopy images and in close-up images.

2.2. Image Acquisition

As a test dataset, 878 images with 5142 ripe apples of different varieties, including red and green apples, were used:

- 553 far-view canopy images (4365 apples in total, 7.89 apples per image on average);
- 274 close-up images (533 apples in total, 1.95 apples per image on average).

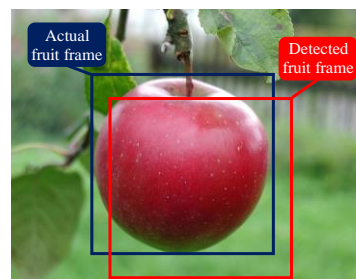
The images were taken manually by the VIM Center employees on the industrial plantation of the apple orchard during the 2019 harvesting season (All-Russian Research Institute for Fruit Crop Breeding, Zhilina, Oryol Region, Russia). Image acquisition was conducted using Nikon D3500 AF-S 18-140 VR cameras equipped with Nikon Nikkor AF-P DX F 18–55 mm lenses. These equipment

specifications are similar to Sony cameras that are installed in our robot. Different pixel resolutions were used: 3888×5184 , 2528×4512 , 3008×4512 , 5184×3888 , and 4032×3024 . The images were collected during sunny and cloudy weather conditions. In order to obtain close-up images and far-view canopy images, different distances for shooting were used: 0.2 m, 0.5 m, 1.0 m, and 2.0 m. In order to obtain images under different natural light conditions, different camera angles were used. As a result, the dataset includes images with front lighting, side lighting, backlighting, and scattered lighting.

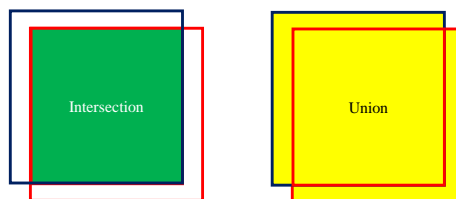
2.3. Apple Detection Quality Evaluation

With the development of the use of convolutional neural networks to evaluate fruit detection quality, the *IoU* (Intersection over Union) metric has become popular. In Figure 1, the navy rectangular bounding box is described around the true fruit, and the red bounding box is obtained as a result of applying the fruit detection algorithm by the machine vision system. *IoU* is the ratio of the area of intersection to the area of union between detected and the ground-truth bounding boxes. A fruit detection system is considered to work satisfactorily if

$$\text{Intersection over Union} = \frac{\sum_{\text{all objects}} \text{Area of Intersection}}{\sum_{\text{all objects}} \text{Area of Union}} > 0.5$$



(a)



(b)

(c)

Figure 1. Intersection over Union for fruits detection. (a) Ground-truth fruit bounding box and detected fruit bounding box; (b) Intersection; (c) Union.

However, in practice, this metric is only an indirect indicator of the fruit-picking system quality.

From a practical point of view, to assess the quality of a fruit detection system, it is important to understand what share of objects is mistaken by the algorithm for apples (False Positive Rate):

$$FPR = 1 - \text{Precision} = \frac{FP}{TP + FP}$$

and what share of apples remains undetected (False Negative Rate):

$$FNR = 1 - \text{Recall} = \frac{FN}{TP + FN}$$

Here:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

TP (True Positives), *FP* (False Positives), and *FN* (False Negatives) are respectively real apples detected by the algorithm in images, objects mistaken by the algorithm for apples, and undetected apples. The *TN* metric (True Negatives) representing background detected as a background is not applicable for deep learning fruit detection frameworks such as YOLO, since these algorithms do not require labeling background class. *Precision* gives the number of correct detections out of total detections, while *Recall* gives the number of correct detections out of total ground-truth fruits. The object detection algorithm is assumed as good if *Precision* remains high as *Recall* increases, i.e., the model can detect a high proportion of True Positives before it starts collecting False Positives. Finally, one more measure is used to evaluate object detection models' quality, *F1 Score*, which is the harmonic mean of the *Precision* and *Recall*.

In this paper, the apple detection results were compared to the ground-truth apples labeled by the authors manually in the images.

2.4. Using YOLOv3 without Pre- and Post-Processing for Apple Detection

First of all, to detect apples, we tried to use the standard YOLOv3 algorithm [64] trained on the COCO dataset [72], which contains 1.5 million objects of 80 categories marked out in images (66,808 persons, 5756 backpacks, 4142 umbrellas, 2346 bananas, 1662 apples, 1784 oranges, etc.).

We did not change the YOLOv3 architecture parameters. We just used trained weights. We used a standard 416×416 input image size and standard anchor boxes ($[116 \times 90, 156 \times 198, 373 \times 326]$, $[30 \times 61, 62 \times 45, 59 \times 119]$, $[10 \times 13, 16 \times 30, 33 \times 23]$) in order to detect large, medium, and small objects in images. The object threshold was set to 0.4. The original images were resized to 416×416 resolution. Since we considered only apple orchards, we were guided by the round shape of objects, and the categories “apples” and “oranges” were combined. Using the standard YOLOv3 algorithm to detect apples in the test images showed that 90.9% of fruits were not detected (Figure 2, Table 1).



Figure 2. Examples of apple detection results by using You Look Only Once (YOLOv3) without pre- and post-processing.

Table 1. Quality metrics for apple detection by standard YOLOv3 without pre- and post-processing.

No. of Images	No. of Apples	Average No. of Apples Per Image	No. of Detected Apples	No. of Not Detected Apples	No. of Objects Mistaken for Apples	Precision	Recall	FNR	FPR
878	5142	5.86	469	4673	52	90.0%	9.1%	90.9%	10.0%

FNR: False Negative Rate; FPR: False Positive Rate.

It means that the algorithm could not be used to detect apples in the harvesting robot. In the following sections, we will introduce some pre- and post-processing techniques that improve apple detection quality significantly.

2.5. Basic Pre- and Post-Processing of Images for YOLOv3-Based Apple Detection Efficiency Improvement

To improve the quality of apple detection, the images were pre-processed, which included:

- contrast increasing by applying histogram normalization and contrast limited adaptive histogram alignment (CLAHE) [73] with 4×4 grid size and clip limit set to 3;
- slight blur by applying the median filter with 3×3 kernel;
- thickening of the borders by use of morphological opening with a flat 5×5 square structuring element.

As a result, it was possible to mitigate the negative effects of shadows, glare, minor damages of apples, and the presence of thin branches overlapping apples. Figure 3a shows examples of images where YOLOv3 without pre-processing is not able to detect apples because of shadows, glare, and overlapping leaves, and Figure 3b shows the same images where pre-processing helped to detect the apples.

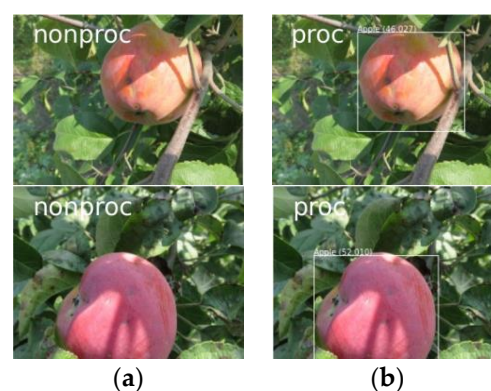


Figure 3. Detecting apples with shadows, glare, and overlapping leaves by YOLOv3 without pre-processing (a) and with pre-processing (b).

On the test dataset, the following main factors preventing the recognition of apples in images were identified:

- backlight;
- existence of dark spots on apples and/or noticeable perianths;
- existence of empty gaps between the leaves, which the network mistook for small apples;
- the proximity of the green apple shade to leaves shade;
- overlapping apples by other apples, branches, and leaves.

To attenuate the negative influence of backlight, images where this problem was detected by the prevailing average number of dark pixels were strongly lightened. Figure 4a shows examples of images

where YOLOv3 without pre-processing is not able to detect apples because of a backlight, and Figure 4b shows the same images where apples are detected by YOLOv3 applied to pre-processed images.

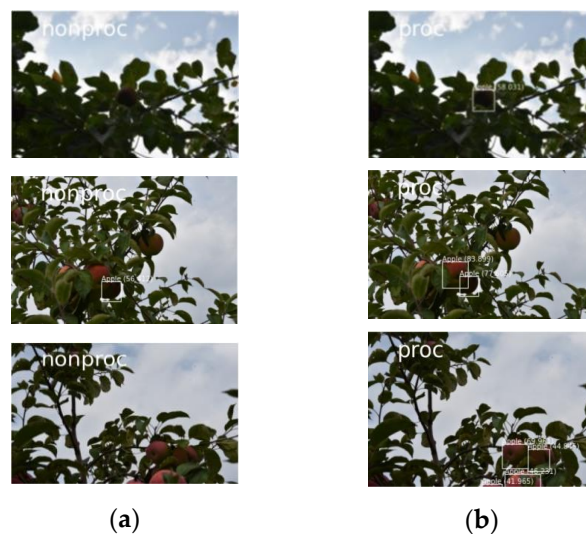


Figure 4. Detecting apples on images with backlight by YOLOv3 without pre-processing (a) and with pre-processing (b).

Since spots on apples, perianth, as well as thin branches, are represented in images by pixels of brown shades, such pixels (with RGB values from (70, 30, 0) to (255, 1540, 0)) were replaced by yellow ones (248, 228, 115). It allowed the system to recognize apples in such images successfully as shown in Figure 5.

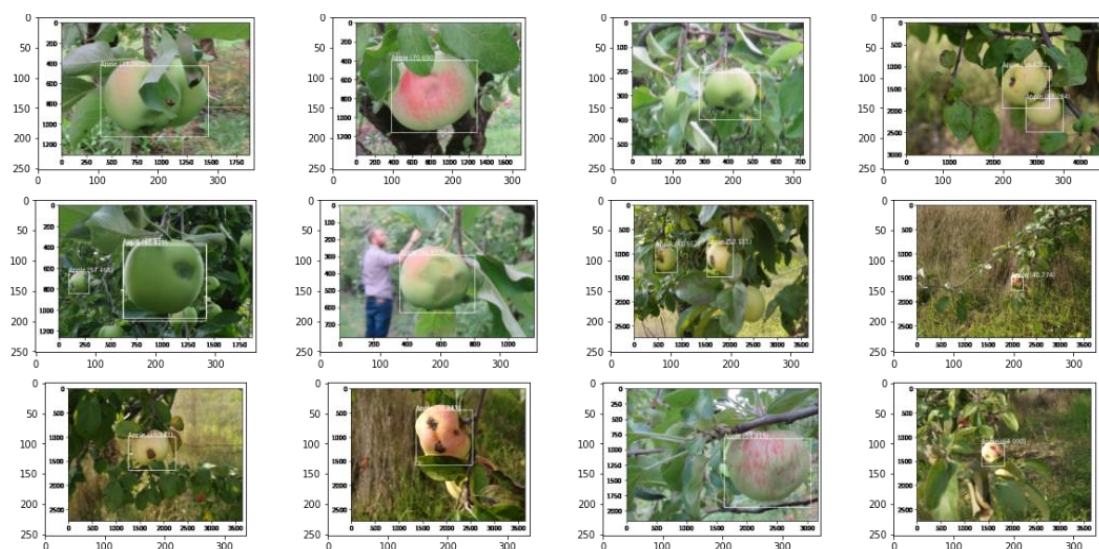


Figure 5. Examples of detected apples with dark spots and overlapping thin branches.

Figure 6 shows examples of images in which yellow leaves, as well as small gaps between leaves, are mistakenly recognized as apples. To prevent the system from taking yellow leaves for apples, during post-processing, we discarded recognized objects whose ratio of the greater side of the circumscribed rectangle to the smaller one was more than 3. In order not to take the gaps between the leaves for apples, during the post-processing, objects were discarded whose area of the circumscribed rectangle was less than the threshold.

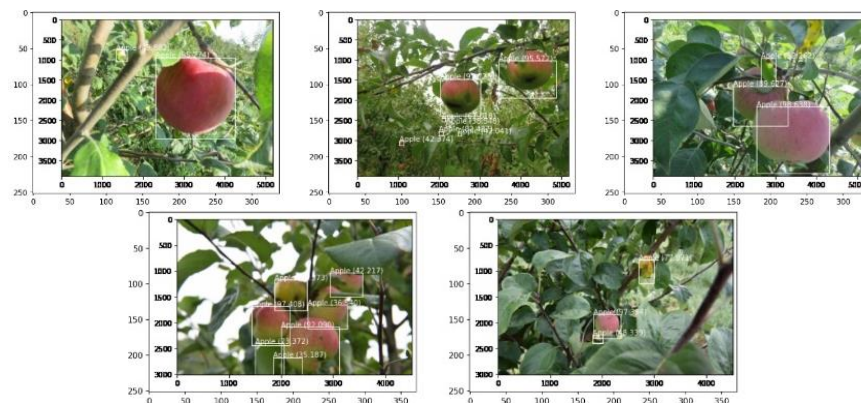


Figure 6. Examples of yellow leaves and gaps between leaves mistaken for apples.

In general, the YOLOv3 algorithm, supplemented by the described pre- and post-processing procedures, quite precisely detects both red and green apples (Figures 7 and 8). Green apples are better detected when the shade of the apple is at least slightly different from the shade of the leaves (Figure 8).

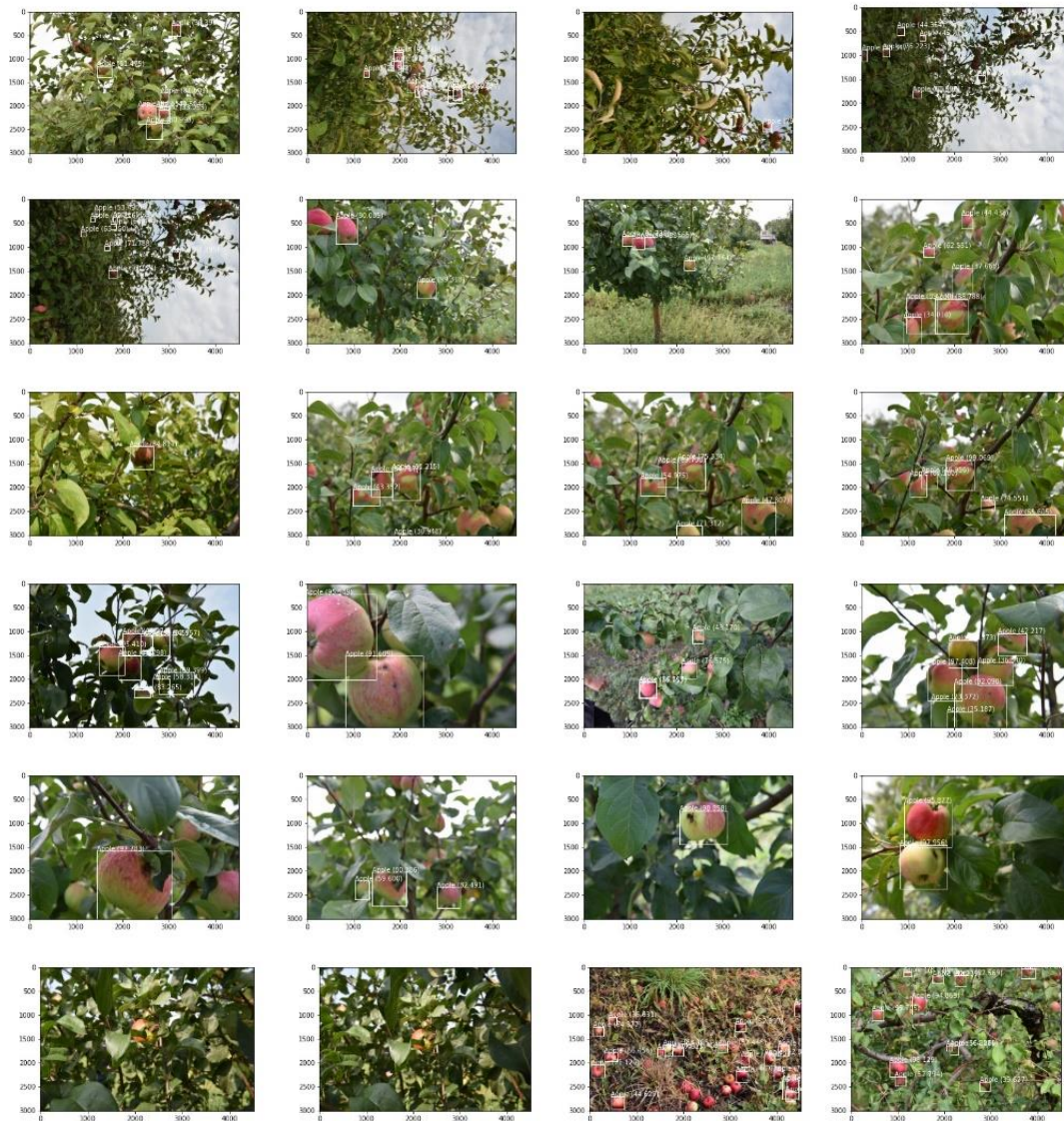


Figure 7. Examples of red apple detection.



Figure 8. Examples of green apple detection.

2.6. Special Pre-Processing for Detecting Apples in Far-View Canopy Images

It turned out that in canopy images, many apples remain undetected. For example, in the images shown in Figure 9a,b, only 2 and 4 apples were detected respectively among several dozens.

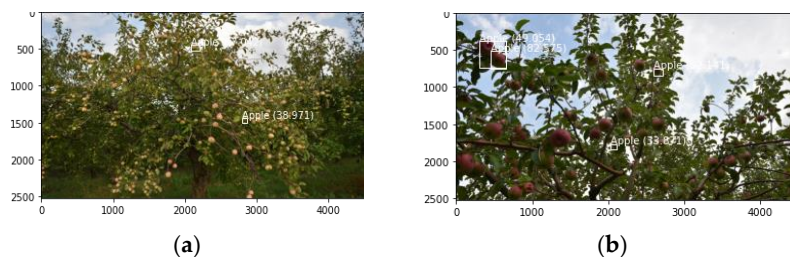


Figure 9. Two (a) and 4 (b) apples detected in the far-view canopy images without pre-processing.

In close-up images, the small apples became smaller than the small anchors of the algorithm. If we increase the number of anchors, the algorithm will work more slowly. Since the far-view canopy images are taken in high resolution, it was more efficient to cut images rather than to tune the anchors. If we assume that the apple in the canopy image is k times smaller than the smallest anchor, then we should divide the original image into k^2 parts. Of course, k increases with the distance from the camera to the object. In addition, the larger k requires a higher resolution for images. So, we took $k = 3$. Setting $k = 2$ and $k = 4$ leads to worse results. Tiny apples in images may be similar to the gaps between the leaves, and therefore, the algorithm cannot detect very distant apples.

Dividing canopy images into 9 regions with the subsequent application of the algorithm separately for each region made it possible to increase the number of detected apples significantly. So, after applying this procedure to the image presented in Figure 9a, 57 apples were detected (Figure 10a), and applying this technique to the image in Figure 9b made it possible to detect 48 apples (Figure 10b).



Figure 10. Fifty-seven (a) and 48 (b) apples found in the far-view canopy images after pre-processing.

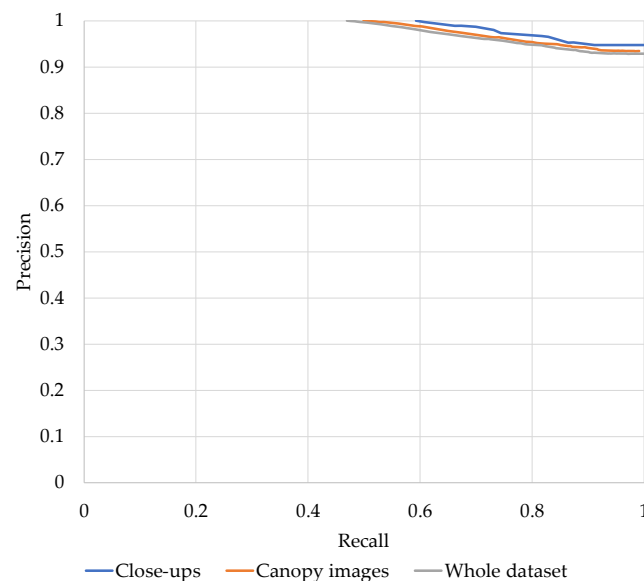
3. Results

During the quality assessment, 878 images from the test dataset described in Section 2.2 were processed using Python scripts on Microsoft Azure NC6 virtual machine with Intel Xeon E5-2690 v3 six-core CPU (2.60GHz), NVIDIA Tesla K80 GPU (24 GiB, 4992 CUDA cores), and 56GiB RAM running on Ubuntu operating system. The software tools include Python 3.8.3 and OpenCV 4.3.0. The detection time for one apple ranged from 7 to 46 ms, considering pre- and post-processing. On average, one apple was detected in 19 ms. We also measured the average detection time for one apple on Intel Core i5-7300U CPU (2.60 GHz) machine with 8 GB RAM running on Ubuntu, and it was 40 ms, which is quite acceptable.

The results of apple detection quality evaluation are presented in Table 2, and Figure 11 presents the *Precision–Recall* curves.

Table 2. Apple detection quality metrics.

No. of Images	No. of Apples	Average No. of Apples Per Image	No. of Detected Apples	No. of Not Detected Apples	No. of Objects Mistaken for Apples	IoU	Precision	Recall	F1	FNR	FPR
878	5142	5.86	4671	471	Whole Set of Images 394	88.9%	92.2%	90.8%	91.5%	9.2%	7.8%
552	4358	7.89	4068	290	Far-View Canopy Images 345	89.7%	92.2%	93.3%	92.8%	6.7%	7.8%
274	533	1.95	446	87	Close-Up Images 30	86.1%	93.7%	83.7%	88.4%	16.3%	6.3%

**Figure 11.** Precision–Recall curve.

Such values of quality metrics are quite acceptable, since both the false positive rate of the algorithm and the share of undetected apples (especially in general images, which determine the route of the manipulator to pick apples) turned out to be quite small. Pre- and post-processing techniques helped to increase the fruit detection rate in comparison with standard YOLOv3 from 9.1% to 90.8%. In general, the system proposed recognizes both red and green apples quite accurately. The system detects apples that are blocked by leaves and branches, green apples on a green background, darkened apples, etc. Manual evaluation of the results shows that there were no multiple detections of the same apple. There also were no splits when one detected box bounds one part of an apple, and another box bounds a different part of the same apple.

The most frequent case, when not all the apples are detected, is when apples form clusters (Figure 12). This is not significant for the robot, since at each step, the manipulator takes out only one apple, and the number of apples in the cluster decreases. It should be noted that this problem arises only when analyzing far-view canopy images presenting several trees with apples. When analyzing images taken in close-up by the camera located on the robot arm, this problem does not occur.

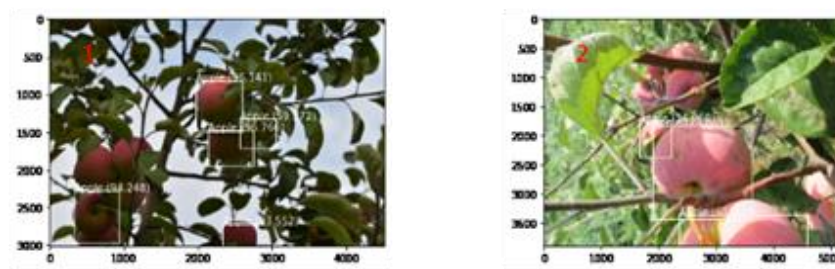


Figure 12. Examples of a partial detection of apples in clusters.

4. Discussion

The results turned out to demonstrate that the YOLOv3 algorithm could be used in harvesting robots in order to detect apples in orchards effectively. However, if we apply this algorithm directly to images taken in real orchards, the detection quality is quite poor. The proposed pre- and post-processing procedures made it possible to adapt the YOLOv3 algorithm for use in apple harvesting robot machine vision system, providing an average apple detection time of 19 ms with a share of not recognized apples at 9.2% and a share of objects mistaken for apples at 7.8%. *Precision* and *F1 Score* are better than in all known similar systems, and the fraction of undetected apples (*FNR*) is better than in most of the known similar systems (Table 3).

Table 3. Comparison to other models.

Model	No. of Images	IoU	Precision	Recall	F1	FNR	FPR
YOLOv3-Dense [67]	480	89.6%	—	—	81.7%	—	—
DaSNet-v2 [70]	560	86.1%	88.0%	86.8%	87.3%	12.0%	13.2%
YOLOv3 [70]	560	85.1%	87.0%	85.2%	86.0%	13.0%	14.8%
YOLOv3 [74]	150	84.2%	—	80.1%	80.3%	9.9%	—
Faster-RCNN [74]	150	86.3%	—	81.4%	81.4%	8.6%	—
LedNet [74]	150	87.2%	—	84.1%	84.9%	5.9%	—
Proposed technique (Whole set of images)	878	88.9%	92.2%	90.8%	91.5%	9.2%	7.8%
Proposed technique (Far-view canopy images)	552	89.7%	92.2%	93.3%	92.8%	6.7%	7.8%
Proposed technique (Close-up images)	274	86.1%	93.7%	83.7%	88.4%	16.3%	6.3%

5. Conclusions

Deep convolutional neural networks combine the ability to recognize objects by color, texture, and shape. In this case, most of the time is spent on training the network, and upon recognition, neural networks significantly outperform classical approaches in time, since recognition is performed by sequentially multiplying matrices in the absence of branches and complex functions.

With some modification, this technique could be applied to detect other spherical fruits such as oranges (Figure 13), tomatoes (Figure 14), et al. The detection quality for oranges is almost the same as for apples, but oranges are not detected when there are some white glares. Some modification of the pre-processing technique could solve this problem. Tomatoes are detected much worse. The problem that prevents the algorithm from recognizing tomatoes is that they are different from apples and oranges by texture and by foliage at the base of fruits. Since YOLOv3 was not trained on tomatoes, to successfully detect them, there is a need to retrain the model on these vegetables.

The concept of transfer learning is currently being developed [75], when trained networks are used as the first layers of new networks. Therefore, it seems essential to conduct further training of the YOLOv3 network to classify recognized apples into healthy apples and apples with various diseases.



Figure 13. Detecting oranges on images by YOLOv3 without pre-processing (a) and with pre-processing (b).



Figure 14. Detecting tomatoes on images by YOLOv3 without pre-processing (a) and with pre-processing (b).

Author Contributions: Conceptualization, V.S.; methodology, A.K., T.M., and V.S.; software, A.K., T.M., and V.S.; validation, A.K.; formal analysis, A.K., T.M., and V.S.; investigation, A.K., T.M., and V.S.; resources, V.S.; data curation, A.K.; writing—original draft preparation, A.K.; writing—review and editing, V.S.; visualization, T.M.; supervision, V.S.; project administration, A.K.; funding acquisition, V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Financial University under the Government of the Russian Federation.

Acknowledgments: The authors thank anonymous reviewers for their attentiveness and valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bechar, A.; Vigneault, C. Agricultural robots for field operations Concepts and components. *Biosyst. Eng.* **2016**, *149*, 94–111. [\[CrossRef\]](#)
2. Sistler, F.E. Robotics and intelligent machines in agriculture. *IEEE J. Robot. Autom.* **1987**, *3*, 3–6. [\[CrossRef\]](#)
3. Ceres, R.; Pons, J.; Jiménez, A.; Martín, J.; Calderón, L. Design and implementation of an aided fruit-harvesting robot (Agribot). *Industrial Robot.* **1998**, *25*, 337–346. [\[CrossRef\]](#)
4. Edan, Y.; Han, S.F.; Kondo, N. Automation in Agriculture. In *Springer Handbook of Automation*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1095–1128.
5. Grift, T.; Zhang, Q.; Kondo, N.; Ting, K.C. A review of automation and robotics for the bio-industry. *J. BioMechatronics. Eng.* **2008**, *1*, 37–54.
6. Kuznetsova, A.; Maleva, T.; Soloviev, V. Detecting Apples in Orchards using YOLO-v3. In Proceedings of the 20th International Conference on Computational Science and Its Applications—ICCSA 2020, Cagliari, Italy, 1–4 July 2020; pp. 1–12.
7. Huang, L.W.; He, D.J. Ripe Fuji apple detection model analysis in natural tree canopy. *TELKOMNIKA* **2012**, *10*, 1771–1778. [\[CrossRef\]](#)
8. Yin, H.; Chai, Y.; Yang, S.X.; Mittal, G.S. Ripe Tomato Extraction for a Harvesting Robotic System. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics—SMC 2009, San Antonio, TX, USA, 11–14 October 2009; pp. 2984–2989.
9. Yin, H.; Chai, Y.; Yang, S.X.; Mittal, G.S. Ripe Tomato Recognition and Localization for a Tomato Harvesting Robotic System. In Proceedings of the International Conference on Soft Computing and Pattern Recognition—SoCPaR 2009, Malacca, Malaysia, 4–7 December 2009; pp. 557–562.
10. Mao, W.H.; Ji, B.P.; Zhan, J.C.; Zhang, X.C.; Hu, X.A. Apple Location Method for the Apple Harvesting Robot. In Proceedings of the 2nd International Congress on Image and Signal Processing—CIPE 2009, Tianjin, China, 7–19 October 2009; pp. 17–19.
11. Bulanon, D.M.; Kataoka, T. A fruit detection system and an end effector for robotic harvesting of Fuji apples. *Agric. Eng. Int. CIGR J.* **2010**, *12*, 203–210.
12. Wei, X.; Jia, K.; Lan, J.; Li, Y.; Zeng, Y.; Wang, C. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optics* **2014**, *125*, 5684–5689. [\[CrossRef\]](#)
13. Zhao, Y.S.; Gong, L.; Huang, Y.X.; Liu, C.L. A review of key techniques of vision-based control for harvesting robot. *Comput. Electron. Agric.* **2016**, *127*, 311–323. [\[CrossRef\]](#)
14. Bulanon, D.M.; Burks, T.F.; Alchanatis, V. Image fusion of visible and thermal images for fruit detection. *Biosyst. Eng.* **2009**, *103*, 12–22. [\[CrossRef\]](#)
15. Wachs, J.P.; Stern, H.I.; Burks, T.; Alchanatis, V. Apple Detection in Natural Tree Canopies from Multimodal Images. In Proceedings of the 7th Joint International Agricultural Conference—JIAC 2009, Wageningen, The Netherlands, 6–8 July 2009; pp. 293–302.
16. Wachs, J.P.; Stern, H.I.; Burks, T.; Alchanatis, V. Low and high-level visual feature-based apple detection from multi-modal images. *Precis. Agric.* **2010**, *11*, 717–735. [\[CrossRef\]](#)
17. Duda, R.O.; Hart, P.E. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **1972**, *15*, 11–15. [\[CrossRef\]](#)
18. Illingworth, J.; Kittler, J. A survey of the Hough transform. *Comput. Vis. Graph. Image Process.* **1988**, *44*, 87–116. [\[CrossRef\]](#)
19. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. [\[CrossRef\]](#) [\[PubMed\]](#)

20. Whittaker, A.D.; Miles, G.E.; Mitchell, O.R. Fruit location in a partially occluded image. *Trans. Am. Soc. Agric. Eng.* **1987**, *30*, 591–596. [[CrossRef](#)]
21. Xie, Z.Y.; Zhang, T.Z.; Zhao, J.Y. Ripened strawberry recognition based on Hough transform. *Trans. Chin. Soc. Agric. Mach.* **2007**, *38*, 106–109.
22. Xie, Z.; Ji, C.; Guo, X.; Zhu, S. An object detection method for quasi-circular fruits based on improved Hough transform. *Trans. Chin. Soc. Agric. Mach.* **2010**, *26*, 157–162.
23. Kelman, E.E.; Linker, R. Vision-based localization of mature apples in tree images using convexity. *Biosyst. Eng.* **2014**, *118*, 174–185. [[CrossRef](#)]
24. Xie, Z.; Ji, C.; Guo, X.; Zhu, S. Detection and location algorithm for overlapped fruits based on concave spots searching. *Trans. Chin. Soc. Agric. Mach.* **2011**, *42*, 191–196.
25. Patel, H.N.; Jain, R.K.; Joshi, M.V. Fruit detection using improved multiple features based algorithm. *Int. J. Comput. Appl.* **2011**, *13*, 1–5. [[CrossRef](#)]
26. Hannan, M.W.; Burks, T.F.; Bulanon, D.M. A machine vision algorithm combining adaptive segmentation and shape analysis for orange fruit detection. *Agric. Eng. Int. CIGR J.* **2009**, *11*, 1–17.
27. Lu, J.; Sang, N.; Hu, Y. Detecting citrus fruits with highlight on tree based on fusion of multi-map. *Optics* **2014**, *125*, 1903–1907. [[CrossRef](#)]
28. OpenCV—Open Source Computer Vision Library. Available online: <https://opencv.org> (accessed on 30 April 2020).
29. Jian, L.; Chengyan, Z.; Shujuan, C. Positioning Technology of Apple-Picking Robot Based on OpenCV. In Proceedings of the 2012 Third International Conference on Digital Manufacturing and Automation, Guilin, China, 31 July—2 August 2012; pp. 618–621.
30. Zhang, Q.R.; Peng, P.; Jin, Y.M. Cherry Picking Robot Vision Recognition System Based on OpenCV. In Proceedings of the 2016 International Conference on Mechatronics, Manufacturing and Materials Engineering—MMME 2016, Hong Kong, China, 11–12 June 2016; pp. 1–4.
31. Zhao, J.; Tow, J.; Katupitiya, J. On-Tree Fruit Recognition Using Texture Properties and Color Data. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 263–268.
32. Rakun, J.; Stajniko, D.; Zazula, D. Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. *Comput. Electron. Agric.* **2011**, *76*, 80–88. [[CrossRef](#)]
33. Kurtulmus, F.; Lee, W.S.; Vardar, A. Green citrus detection using ‘eigenfruit’, color and circular Gabor texture features under natural outdoor conditions. *Comput. Electron. Agric.* **2011**, *78*, 140–149. [[CrossRef](#)]
34. Kurtulmus, F.; Lee, W.S.; Vardar, A. An advanced green citrus detection algorithm using color images and neural networks. *J. Agric. Mach. Sci.* **2011**, *7*, 145–151.
35. Parrish, E.A.; Goksel, J.A.K. Pictorial pattern recognition applied to fruit harvesting. *Trans. Am. Soc. Agric. Eng.* **1977**, *20*, 822–827. [[CrossRef](#)]
36. Sites, P.W.; Delwiche, M.J. Computer vision to locate fruit on a tree. *Trans. Am. Soc. Agric. Eng.* **1988**, *31*, 257–263. [[CrossRef](#)]
37. Bulanon, D.M.; Kataoka, T.; Okamoto, H.; Hata, S. Development of a Real-Time Machine Vision System for Apple Harvesting Robot. In Proceedings of the Society of Instrument and Control Engineers Annual Conference, Sapporo, Japan, 4–6 August 2004; pp. 595–598.
38. Seng, W.C.; Mirisae, S.H. A New Method for Fruits Recognition System. In Proceedings of the 2009 International Conference on Electrical Engineering and Informatics—ICEEI 2009, Selangor, Malaysia, 5–7 August 2009; Volume 1, pp. 130–134.
39. Linker, R.; Cohen, O.; Naor, A. Determination of the number of green apples in RGB images recorded in orchards. *Comput. Electron. Agric.* **2011**, *81*, 45–57. [[CrossRef](#)]
40. Ji, W.; Zhao, D.; Cheng, F.Y.; Xu, B.; Zhang, Y.; Wang, J. Automatic recognition vision system guided for apple harvesting robot. *Comput. Electr. Eng.* **2012**, *38*, 1186–1195. [[CrossRef](#)]
41. Tao, Y.; Zhou, J. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Comput. Electron. Agric.* **2017**, *142*, 388–396. [[CrossRef](#)]
42. Zhan, W.T.; He, D.J.; Shi, S.L. Recognition of kiwifruit in field based on Adaboost algorithm. *Trans. Chin. Soc. Agric. Eng.* **2013**, *29*, 140–146.
43. Zhao, Y.S.; Gong, L.; Huang, Y.X.; Liu, C.L. Robust tomato recognition for robotic harvesting using feature images fusion. *Sensors* **2016**, *16*, 173. [[CrossRef](#)]

44. Zhao, Y.S.; Gong, L.; Huang, Y.X.; Liu, C.L. Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. *Biosyst. Eng.* **2016**, *148*, 127–137. [CrossRef]
45. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems Conference—NIPS 2012, Harrahs and Harveys, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1–9.
46. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations—ICLR 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
47. Williams, H.A.M.; Jones, M.H.; Nejati, M.; Seabright, M.J.; MacDonald, B.A. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [CrossRef]
48. Liu, Z.; Wu, J.; Fu, L.; Majeed, Y.; Feng, Y.; Li, R.; Cui, Y. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. *IEEE Access* **2020**, *8*, 2327–2336. [CrossRef]
49. Mureșan, H.; Oltean, M. Fruit recognition from images using deep learning. *Acta Univ. Sapientiae. Inform.* **2018**, *10*, 26–42. [CrossRef]
50. Fruits 360 Dataset. Available online: <https://github.com/Horea94/Fruit-Images-Dataset> (accessed on 30 April 2020).
51. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
52. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision—ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
53. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
54. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision—ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
55. He, K.X.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition—CVPR 2016, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
56. Zhang, J.; He, L.; Karkee, M.; Zhang, Q.; Zhang, X.; Gao, Z. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). *Comput. Electron. Agric.* **2018**, *155*, 386–393. [CrossRef]
57. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. DeepFruits: A fruit detection system using deep neural networks. *Sensors* **2016**, *16*, 1222. [CrossRef] [PubMed]
58. Bargouti, S.; Underwood, J. Deep Fruit Detection in Orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation—ICRA 2017, Singapore, 29 May–3 June 2017; pp. 1–8.
59. Peebles, M.; Lim, S.H.; Duke, M.; McGuinness, B. Investigation of optimal network architecture for asparagus spear detection in robotic harvesting. *IFAC PapersOnLine* **2019**, *52*, 283–287. [CrossRef]
60. ACFR-Multifruit-2016: ACFR Orchard Fruit Dataset. Available online: <http://data.acfr.usyd.edu.au/ag/treecrops/2016-multifruit/> (accessed on 30 April 2020).
61. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
62. Jia, W.; Tian, Y.; Luo, R.; Zhang, Z.; Zheng, Y. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* **2020**, *172*, 105380. [CrossRef]
63. Gené-Mola, J.; Gregorio, E.; Cheein, F.A.; Guevara, J.; Llorens, J.; Sanz-Cortiella, R.; Escolà, A.; Rosell-Polo, J.R. Fruit detection, yield prediction and canopy geometric characterization using LiDAR with forced air flow. *Comput. Electron. Agric.* **2020**, *168*, 105–121. [CrossRef]
64. Gan, H.; Lee, W.S.; Alchanatis, V.; Ehsani, R.; Schueller, R. Immature green citrus fruit detection using color and thermal images. *Comput. Electron. Agric.* **2018**, *152*, 117–125. [CrossRef]
65. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once Unified, Real-Time Object Detection. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition—CVPR 2016, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.

66. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. YOLOv3 An Incremental Improvement. In Proceedings of the 31th IEEE Conference on Computer Vision and Pattern Recognition—CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1–6.
67. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [[CrossRef](#)]
68. Tian, Y.; Yang, G.; Wang, Z.; Li, E.; Liang, Z. Detection of apple lesions in orchards based on deep learning methods of CycleGAN and YOLO-V3-Dense. *J. Sens.* **2019**, 1–14. [[CrossRef](#)]
69. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition—CVPR 2017, Honolulu, HI, USA, 22–25 July 2017; pp. 1–9.
70. Kang, H.; Chen, C. Fruit detection, segmentation and 3D visualization of environments in apple orchards. *Comput. Electron. Agric.* **2020**, *171*, 105302. [[CrossRef](#)]
71. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [[CrossRef](#)]
72. COCO Common Objects in Context Dataset. Available online: <http://cocodataset.org/#overview> (accessed on 30 April 2020).
73. Ferguson, P.D.; Arslan, T.; Erdogan, A.T.; Parmley, A. Evaluation of Contrast Limited Adaptive Histogram Equalization (CLAHE) Enhancement on a FPGA. In Proceedings of the 2008 IEEE International SOC Conference, Newport Beach, CA, USA, 17–20 September 2008; pp. 119–122.
74. Kang, H.; Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105108. [[CrossRef](#)]
75. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).