agriculture

MDPI

Article

Automated Apple Recognition System Using Semantic Segmentation Networks with Group and Shuffle Operators

Mohd Asyraf Zulkifley [1,*], Asraf Mohamed Moubark [1], Adhi Harmoko Saputro [2] and Siti Raihanah Abdani [3]

[1] Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; asrafmohamed@ukm.edu.my
[2] Faculty of Mathematics and Natural Science, Universitas Indonesia, Depok 16424, Indonesia; adhi@sci.ui.ac.id
[3] Faculty of Humanities, Management and Science, Universiti Putra Malaysia (Bintulu Campus), Bintulu 97008, Sarawak, Malaysia; raihanah.abdani@siswa.ukm.edu.my
* Correspondence: asyraf.zulkifley@ukm.edu.my

Abstract: Apples are one of the most consumed fruits, and they require efficient harvesting procedures to remains in optimal states for a longer period, especially during transportation. Therefore, automation has been adopted by many orchard operators to help in the harvesting process, which includes apple localization on the trees. The de facto sensor that is currently used for this task is the standard camera, which can capture wide view information of various apple trees from a reasonable distance. Therefore, this paper aims to produce the output mask of the apple locations on the tree automatically by using a deep semantic segmentation network. The network must be robust enough to overcome all challenges of shadow, surrounding illumination, size variations, and occlusion to produce accurate pixel-wise localization of the apples. A high-resolution deep architecture is embedded with an optimized design of group and shuffle operators (GSO) to produce the best apple segmentation network. GSO allows the network to reduce the dependency on a few sets of dominant convolutional filters by forcing each smaller group to contribute effectively to the task of extracting optimal apple features. The experimental results show that the proposed network, GSHR-Net, with two sets of group convolution applied to all layers produced the best mean intersection over union of 0.8045. The performance has been benchmarked with 11 other state-of-the-art deep semantic segmentation networks. For future work, the network performance can be increased by integrating synthetic augmented data to further optimize the training phase. Moreover, spatial and channel-based attention mechanisms can also be explored by emphasizing some strategic locations of the apples, which makes the recognition more accurate.

Keywords: apples recognition; deep learning; semantic segmentation; convolutional neural networks

Citation: Zulkifley, M.A.; Moubark, A.M.; Saputro, A.H.; Abdani, S.R. Automated Apple Recognition System Using Semantic Segmentation Networks with Group and Shuffle Operators. *Agriculture* **2022**, *12*, 756. https://doi.org/10.3390/agriculture12060756

Academic Editor: Maciej Zaborowicz

Received: 6 April 2022
Accepted: 25 May 2022
Published: 26 May 2022

1. Introduction

Apples are a popular fruit in the global market, especially in European countries, who have traded more than USD 500 billion worth of apples in 2019 [1]. The shelf life of commercial apples can reach one month if they are stored properly. Even if the apples have been on the shelf longer than the suggested period, they can still be used in the bioconversion process to produce citric acid, which is a very useful organic compound [2]. There are many types of apple, such as Fuji, Empire, Golden Delicious, Braeburn, and Cortland, and the most consumed apple type varies from one country to another. In the United States of America, consumers prefer Red Gala, whereas Australians prefer the Granny Smith version. Because they are one of the most consumed fruits, commercial orchards can be found all over the world. Some of the producer countries focus more on the export market such as China and Italy, which have a total apple export worth more than USD 2.2 billion in 2020 [3]. The largest importer of apples is Russia, which is known to have extremely cold weather that limits the country's capability to produce apples on a large scale.

Due to their big commercial value, many orchards have been planted with apples in China, especially in the Shanxi Province, which has more than 300 kilohectares of apple orchards [4]. Hence, it is crucial to automate the management of the orchards starting from the planting, providing periodic fertilizer, and harvesting of the fruits. Moreover, due to the COVID-19 pandemic, there are traveling restrictions between countries and even between certain provinces that limit the flow of workers from one place to another [5]. In some countries, apple orchards are still managed manually, which requires a lot of hard labor, whereby they are sourced mainly from underdeveloped countries. Hence, due to the lack of workers because of COVID-19, many apples are not harvested on time, which leads to major loss. Therefore, it is important to automate the management process of the apple orchard to reduce dependency on manual labor.

One of the important aspects of managing an apple orchard is to harvest the fruits at the right time, so that the harvested apples are mature enough to be picked and young enough to be transported all over the world [6]. In an automated harvesting system, the apples need to be recognized first before any specialized robotic end-effector can be guided towards them. Therefore, each of the individual apples needs to be recognized from four to six meters away from the respective tree, as the minimum distance between two apple trees is usually around 4 m. Figure 1 depicts the optimal distance between two rows of apple trees in a standard apple orchard. As the recognition process needs to be done from six meters distance, a standard RGB camera is usually used to capture the imaging information, which will include a lot of occluded apples, either full or partial occlusion. It is not unusual for the apples to be heavily occluded by the leaves or other apples, which prompts the automated harvesting system to move around to capture more imaging samples rather than single point imaging. Due to the moving images, some automated systems have even implemented tracking algorithms in order to correlate the images between various captured reference points [7]. Apart from harvesting purposes, it is also important to recognize the apples automatically from a distance, as this information can be used to predict the output yields. An intelligent yield prediction system will allow the manager to better plan the types of apples that need to be planted.

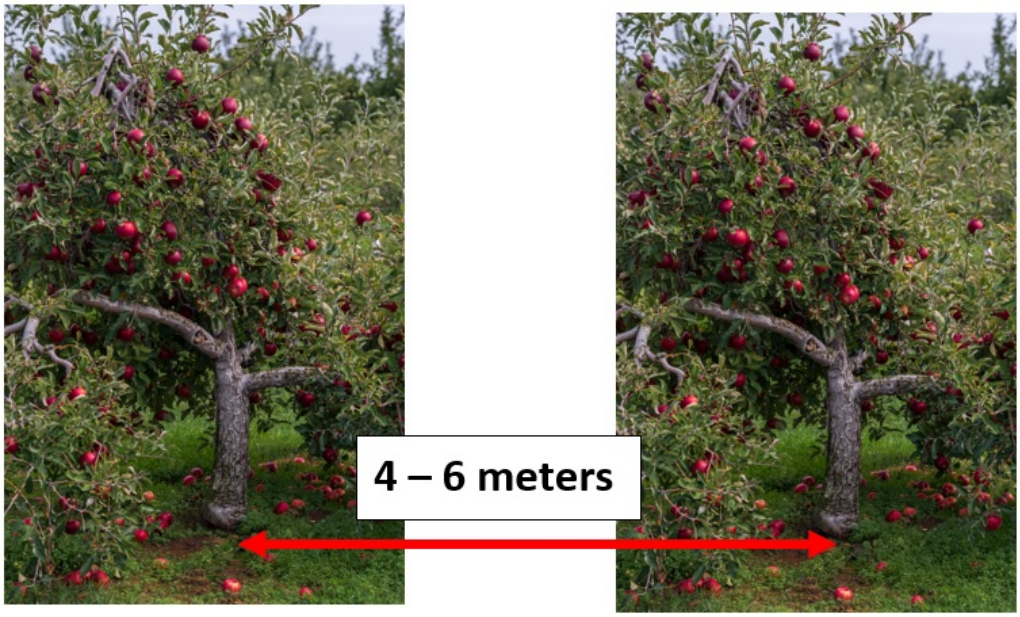

Figure 1. The optimal distance between two rows of apple trees. The distance also depends on the apple types and hence the imaging should be able to capture the apple information from 6 m.

There are various challenges that need to be taken care of in order to design an effective automated recognition of apples in the open orchard. The first challenge is the inconsistent brightness level of the environment. Because the system will be implemented in an outdoor

setting, the captured images will be affected by the changes in the sunlight brightness either due to clouds, rain, or capture time. Hence, the recognition system must be able to recognize the apples regardless of the time when the images were captured. In some computer vision applications, gradual changes in surrounding illumination are solved through a color constancy approach that transforms all input images into a standardized form that mimics the average grey value of the reference image [8]. Furthermore, cast shadows will also induce local brightness changes [9], whereby the affected apples will have a much darker appearance compared to the neighboring apples. Apart from that, another main challenge encountered by the automatic recognition system is occlusion. The system must be able to recognize the location of apples, even if there is only a small fraction of the full size of the apples. Partial occlusion due to the leaves will always be present, as such it needs to be considered as part of the design criteria of the automated system. Another crucial challenge for an automated apple recognition system is the ability to recognize the fruits according to their various growing phases. For most apple types, the young ones will predominantly be green in color, whereas the mature apple will be mostly red in color. Generally, the apple will comprise a mix of colors, and hence, geometric information needs to be extracted concurrently to solve the recognition problem. Distance of the captured images with respect to the tree is also an important design challenge for the automated recognition system. For an RGB imaging modality, distance restriction is not a problem, as the resolution of the imaging camera is good enough even if the images were captured from a mobile phone. However, if the system is embedded with a depth camera such as Kinect, the optimal distance will be limited to less than 2 m [10]. As a consequence, the automated harvesting machine needs to come closer to the trees before any recognition can be done, which severely limits the robustness of the system. Figure 2 shows some samples of challenges that an automatic apple recognition needs to overcome to produce a robust solution.

Figure 2. Some challenges encountered by an image-based automated apple recognition system: (**A**) occlusion, (**B**) various surrounding illumination levels, (**C**) variable distance of apple trees, and (**D**) multiple colors of apple fruits.

In this paper, the recognition of the apples will be done through a single imaging modality, which is based on an RGB camera that will allow a longer effective distance between the camera and the tree. However, the distance is still limited to be less than six meters because of the distance limiting factor between two opposite apple trees. The proposed method will be based on a deep learning semantic segmentation technique that will be able to recognize all the apples in the imaging field of view. The method was chosen because of its ability to detect the apple regions of various growing phases regardless of young or

mature fruits. The semantic segmentation method allows pixel-wise labeling and hence, the occlusion issue will not be a major performance limiter. Moreover, the network will be trained and tested with images of apples that were captured under various outdoor settings so that the system can still be fully functional under varying brightness of the surrounding illumination. The developed system utilizes an end-to-end training philosophy, whereby the required input is just the orchard images taken from the side view without requiring any intervention from the experts to determine the optimal set of features. The features will be learned optimally mainly using convolutional neural networks (CNN) architecture in the form of encoder-decoder network [11]. The output of the network will be the mask that labels each pixel in the image whether it belongs to an apple or not. Additional group and shuffle operators [12] are embedded into the semantic segmentation networks to improve the feature learning capability by urging the network to consider minor features in making the final decision. Without group and shuffle operators, the network tends to learn a set of dominant global features, which put low weights on some minor kernel output. A sample of ground truth masks, which has been labeled manually by the computer vision experts with the help of an agriculture expert, is shown in Figure 3.

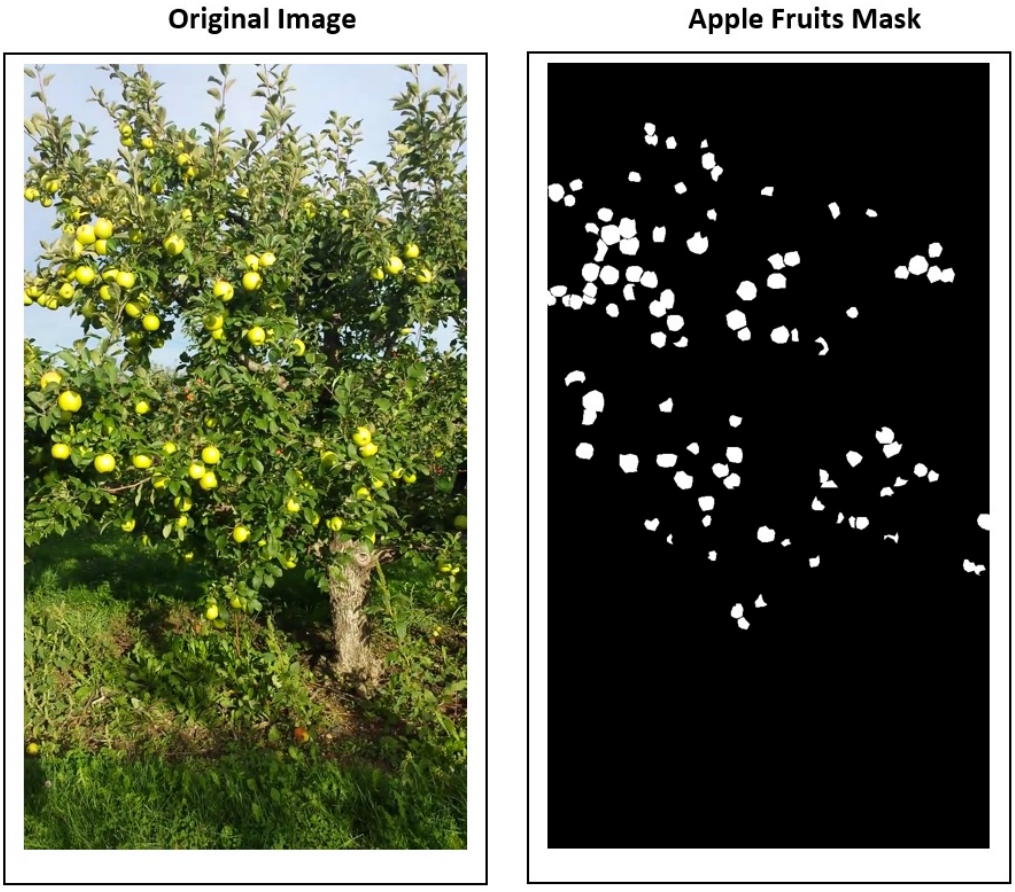

Figure 3. A sample of captured apple trees image and its corresponding ground truth mask of apple fruits.

This paper is organized into five sections that started with an introduction section that explores the challenges in designing an automated apple recognition system. Section 2 then discusses the related work on the previous imaging-based apple recognition systems that cover various approaches of machine vision-based techniques. Section 3 details the proposed semantic segmentation network of the apple recognition system using deep encoder-decoder architecture. In addition, several variants of the networks are proposed by integrating group and shuffle operators. The performance of the apple recognition system is then compared with the state-of-the-art deep semantic segmentation networks.

Section 5 concludes the proposed apple recognition system with a few suggestions for future improvement.

2. Related Work

The most utilized color space in automated apple recognition systems is the RGB color model. One of the earliest studies using the RGB color model was designed by Ji et al. [13], which used a set of medium resolution of Fuji apple images with 640 × 480 pixels that were captured from an orchard in Jiangsu Province, China. All input images were first enhanced through a vector median filter to reduce the random noise. A region growing segmentation technique was used to produce a preliminary apple mask by pivoting the region seeds based on the highest probability pixels that might belong to an apple class. If the number of pixels in the growing region achieved a minimum of 20 pixels, one mask of an apple was fully generated. If the total pixel in one cluster is less than 20, another isolated cluster that has been screened to belong to an apple class will be combined together to produce one full mask. A threshold-based method was used to determine the cluster combination decision through the Euclidean distance metric. Several input features, which are variance, ellipse variance, tightness, and segmented area ratio, were extracted to determine the final class of each cluster by using a support vector machine (SVM) classifier. They have also explored three kernel functions that include poly kernel function, radial basis function (RBF), and sigmoid function, and the results showed that RBF produced the best overall apple recognition performance. Instead of using the RGB color model directly, the work in [14] utilized the difference values between the red, green, and blue channels. They proposed the automated recognition system based on high-resolution images of 1704 × 2272 pixels of Gala apple images, which were taken in Bonn, Germany. In order to segment each pixel, a simple threshold-based method was applied by utilizing the difference in channel values. There are two calculated channel differences, whereby the first value is the difference between red and blue channels, and the second value is the difference between green and red channels. The emphasis on the red channel for the intelligent agriculture systems is in stark contrast to the general biomedical systems that usually assign greater weights to the green channel [15]. These two gradient values were chosen to detect both red and green apples. Median, erosion, and dilation filters were also applied to remove the noise, especially on the apple boundaries. Connected pixel components were then used to link the separated apple pixels, and a threshold of 400 pixels was used to segregate the various clusters of the initial apple masks.

In the apple recognition system proposed by Jiang and Zhao [16], they transformed the RGB input images into a single channel model by putting more emphasis on the red channel contribution. The weight of the red channel is set to be twice larger than for the green and blue channels. They also assumed that their system is only usable for red apples, therefore they have set the transformed color channel to zero if the contribution of twice the red channel value is less than the combination of green and blue pixel values. Each pixel will be labeled deterministically using the threshold-based method. Otsu adaptive approach is used to determine whether or not a pixel belongs to an apple. Morphological operations through dilation and erosion were then applied to smooth out the original segmented apple boundaries. Sobel edge detector was used to extract the edges information, which becomes the input to the Hough circle transform to extract the final boundaries. All tested images do not contain any occlusion situations, which is rare in real-life applications. In addition, Syal et al. [17] also tested their automated system using a simplified zoomed-in apple image. Before the images were processed, a Gaussian low pass filter was applied to normalize the original RGB pixel ranges, which were then converted to CIELAB color space. From then, only two channels were utilized, which were the a and b channels. Using the information from these two channels, pixel-wise segmentation was performed deterministically based on the mean values of a and b channels. Euclidean distance was then used to measure the channel value differences between the tested value of each pixel and the pre-chosen apple

model. Once the raw segmented masks of the apples were generated, circle fitting was performed to cluster a set of pixels that potentially belong to a specific apple.

Apart from the CIELAB color space, Tanco et al. [18] suggested using HSV color space to process the images for automated recognition of apples. A set of 266 high-resolution HSV images from Canelones, Uruguay were analyzed to identify the apple's pixels using deterministic ranges of hue and saturation channels. If any pixel values fall within the ranges, they were flagged as the initial candidate for the apple class. Sobel edge detector then was used to extract the edge information, whereby an assumption based on texture density was utilized to screen the initial candidates. Three types of the classifier were explored that include decision tree, SVM, and k-nearest neighbor (KNN) to determine the possibility of a pixel belonging to an apple class based on the extracted color and texture features. Morphological operations that include erosion, dilation, and circular fitting were then applied to produce the final segmented mask of the apples. KNN classifier produced the best recognition accuracy, which has been validated using a difficult dataset that contains heavy occlusion.

Rather than using RGB image as the sole input to the automated system, Nguyen et al. [19] integrated the depth image as additional information to recognize the apples. The recognition is heavily dependent on the redness level of a particular pixel, which was validated with images from an apple orchard in Sint-Truiden, Belgium. Once an early decision for each pixel has been generated, Euclidean clustering was then used to group the possible sets of apples. The circular Hough transform was then fitted to identify the ideal apple shape so that the goodness score of each cluster, whether it belongs to an apple or not, can be determined. The primary weakness of this method appears when the apples are occluded with other apples or leaves. The authors have also attached a canopy to the proposed machine, so that all imaging was taken under homogenous lighting, which will limit the robustness of the system. Another method that has utilized RGB-D sensors as an input to automated apple recognition was proposed by Yongting and Jun [20]. They utilized Kinect V2 by Microsoft to capture the input images in two different resolutions. The RGB image was captured in a high-resolution format of 1920 $\times$ 1080 pixels, but the depth image was captured in a medium-resolution format of 512 $\times$ 424 pixels. The effective range of the sensor was also limited; it needed to be less than 2 m and hence limited the robustness of the system for real-life application. Three-dimensional geometric features in the form of point cloud representation were used as input to a region growing segmentation technique to cluster the possible apple regions. For each selected seed point, Euclidean distance of the color and depth features was used to determine the decision to absorb the candidate pixels or not. To finalize the decision, the clustered regions were passed to an optimized SVM-genetic algorithm classifier to determine whether they belong to the apple, branch, or leaf class.

In real-life applications, some apple orchards face the problem of strong winds, especially during harvesting season, which make the apples sway a lot. Hence, Yang et al. [7] embedded a tracking algorithm to improve the performance of automated apple recognition. The initial mask of the apple regions was generated using pixel-based labeling through the Otsu method by emphasizing more on the red channel. The generated mask had many disconnected components, which were then combined using the Hough circular transform. Each of the clustered apple regions was tracked based on the corner point of interest, which was matched between the frames using optical flow matching. The location of the apple regions was then tracked with regard to the fixed camera position. The tracking update was done through template matching of cross-correlation algorithm. As the apples were assumed to be continuously swaying, the relative size of the captured apples also vary slightly from frame to frame. Therefore, the RANSAC algorithm was used to normalize the scale by standardizing the size with respect to the pivot frame. Finally, a parallel processor was also used to speed up all the algorithm computation to produce a real-time output. Awad et al. [21] utilized a field programmable gate array to process a unique YCbCr image for the apple recognition system. They selected this color model because of the limitation

of the RGB color model in handling constant changes in the surrounding brightness level. Each pixel was labeled according to the possibility of a particular pixel: either it belonged to an apple or not. K-mean clustering was then used to combine the positively labeled pixels to generate the apple mask automatically.

3. GSHR-Net

The proposed method aims to help the automated harvesting machine recognize the apples on the trees by providing the location mask. The recognition process will rely solely on the basic RGB image input without utilizing the depth image, so that the effective range of the apple recognition system is wider. The output of the recognition process is the mask of all apple locations on the image, which will be represented by a high-pixel value, whereas the background will be labeled as a low-pixel value. A binary semantic segmentation approach [22] is utilized in this work; as such, each pixel will be labeled as either belonging to an apple or not. No instance labeling will be performed to distinguish between two apples because of the high occlusion severity in the tested dataset. In addition, the automatic yield prediction and harvesting guide mechanism do not require exact distinction of the apples, especially when the images were taken from 4 to 6 m with respect to the apple trees.

In this work, the deep learning approach to semantic segmentation is explored because of its high segmentation accuracy compared to conventional machine learning techniques. In addition, the deep learning approach enables an optimal feature extraction process through recursive learning without relying on the expert's opinion in selecting the final set of features of interest. Moreover, the training and inference processes will be executed through an end-to-end mechanism, whereby the required input is just the RGB image, and the network will directly produce the output in the form of the apple location mask. Therefore, the architecture of the deep network has been optimized to produce the most accurate apple segmentation mask through group and shuffle operators (GSO). Several variants of the GSO architecture have been devised so that the final set of features does not depend heavily on only a few dominant filters. Through the shuffling process, the flow of the features map will be altered and hence reduces the possibility of one dominant flow. Moreover, the utilization of group convolution will enable each smaller group of convolution filters to contribute to the whole decision process effectively rather than only a few primary filters. HRNet [23] is chosen as the base segmentation network, whereby the GSO module will replace the original basic convolution block. Figure 4 shows the overall architecture design of the proposed deep learning segmentation network, which has been named GSHR-Net.

The network consists of four scales, whereby three smaller-scale feature maps are generated gradually while retaining feature maps with the original size. The original input size is 512 $\times$ 512 pixels, which is relatively higher than most semantic segmentation networks (FC-DenseNet [24] with 224 $\times$ 224 pixels, DeepLab [25] with 321 $\times$ 321 pixels, and UNet [26] with 224 $\times$ 224 pixels). During the first stage, two layers of standard convolution operators are applied to initialize the network flow with 64 convolutional kernels with the size of 3 $\times$ 3 pixels. In the proposed network, each convolution operator will be coupled with a batch normalization layer and a rectified linear unit (ReLU) activation function. Batch normalization that acts as a regularization step will expand the range of the feature maps output according to the batch data, and ReLU will transform the feature maps into a non-linear representation for more effective feature extraction. After that, a bottleneck block will be executed that consists of a series of four skip connection modules. For each module, a skip connection branch will be spawned from the input layer and fed to the output of three convolutional layers with the same filter number of 64. At the end of the first stage, the feature maps, which are now 128 $\times$ 128 pixels, will be passed to the first transition block to create a parallel branch with a smaller scale of 64 $\times$ 64 pixels. The number of convolutional channels is smaller for the larger-scale feature maps compared to the smaller-scale feature maps.

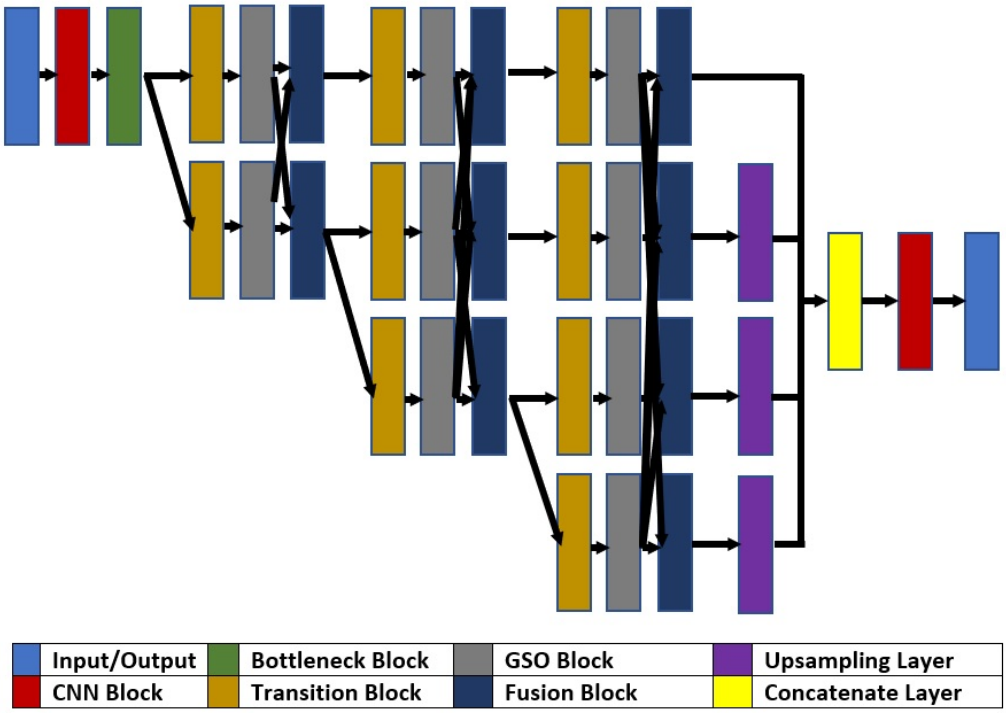

Figure 4. The full network architecture of the proposed GSHR-Net.

It is interesting to note that GSHR-Net retains the original high-resolution feature map throughout the networks. In the subsequent stages, the original resolution feature maps will be maintained together with the down-sampled feature maps. The lower resolution feature maps are obtained through a convolution operator with a larger stride size, not through the down-pooling process. After the new branch with a smaller scale has been created, each of the separate paths will be passed through the GSO block before the feature maps of various sizes are fused together. There are three variants of the GSO block that have been developed, as detailed in Figure 5.

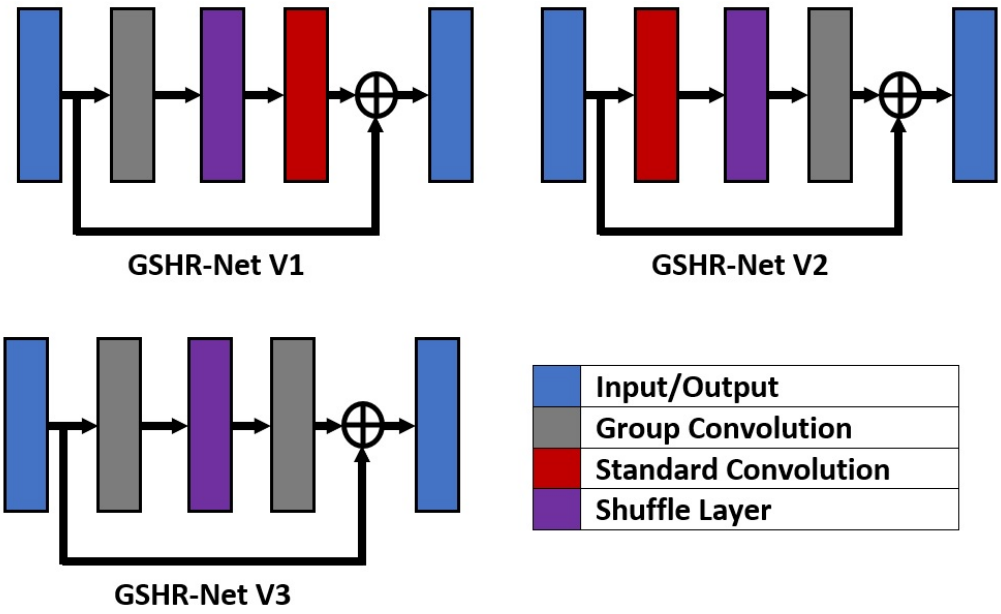

Figure 5. Three variants of group and shuffle operators network architecture.

The first variant, GSHR-Net V1, consists of a sequence of group convolution, shuffle operator, and standard convolution, which is then combined with a skip connection from the input layer. The second variant, GSHR-Net V2, consists of a sequence of standard convolution, shuffle operator, and group convolution, which is then combined with a skip connection from the input layer. For these two variants, only a single layer of group convolution is implemented. The last variant, GSHR-Net V3, comprises a sequence of group convolution, shuffle operator, and another group convolution, before a skip connection from the input layer is added together at the end of the network block. There are three group settings, which are 2, 4, or 8 groups used in the group convolution process. Each of the settings will be tested for all GSHR variants to find the best segmentation architecture. The shuffling process will mix the incoming channels from different groups to alter the network flow so that the smaller branches can be optimally updated.

After the GSO block, the fusion block will be executed so that the features of various resolution paths can be concatenated together to increase the richness of information. For the larger scale path, feature maps of the smaller scale path will be up-sampled through bilinear interpolation. For the smaller scale path, feature maps of the larger scale path will be down-sampled using convolution operator with a stride size of 2. Once the size of the feature maps has been adjusted, the addition operator is used to combine the feature maps for each parallel path. In order to prevent exponential channel expansion when more parallel paths are added, the addition operator is utilized in the fusion block, rather than the concatenation operator. A similar pattern observed in the GSO block can be traced in the fusion block, whereby more convolution filters are used for the smaller scale feature maps during the up-sample and down-sample processes. At the end of the fusion block, a new additional path of a smaller scale is added, whereby a lower resolution path is spawned again through a convolution operator with a stride size of 2.

The process will be repeated until the network has processed four parallel paths of the feature maps that comprise four scales, which are 128 $\times$ 128, 64 $\times$ 64, 32 $\times$ 32, and 16 $\times$ 16 pixels. After the last fusion block, the feature maps from four parallel paths will be combined through a concatenation operator, whereby the three smaller scale paths will be up-sampled first to match the size of the largest feature map size using the bilinear interpolation process. After the feature maps have been concatenated together, two layers of convolutional filter are added to complete the optimized semantic segmentation network architecture. For the first layer, a total of 256 filters are used, whereas for the second layer, only two filters are used that correspond to the two targeted output classes, which are apple regions and background. To be specific, only two complementary segmentation maps will be produced in the last layer, as the goal of this work is to produce a mask of the apples and a mask of the background information. To produce the output maps, the softmax function is used as the final layer activation function. In addition, this work utilizes three configurations for the group convolution operator that depends on the total number of groups per layer; 2, 4, and 8-group schemes were devised for each variant of GSHR-Net, which results in a total of nine GSHR-Net variants. The suffix at the end of the GSHR-Net indicates the variant and the total number of groups used in the group convolution.

4. Results and Discussion

The images used to validate the proposed GSHR-Net performance were downloaded from a public online database, MinneApple (http://rsn.cs.umn.edu/index.php/MinneApple, accessed on 1 February 2022), which were collected and maintained by University of Minnesota researchers using a standard mobile phone camera, Samsung Galaxy S4. There are 670 annotated images for the segmentation category, whereby the data are divided into test and training subsets according to the ratio of 1:4. The images were saved in the portable network graphics (PNG) format with a resolution of 720 $\times$ 1280 pixels. This dataset contains apples of various colors that include green, red, orange, and a mix of all these colors. The data were gathered by the researchers walking around the apple orchard, hence some apples are far away from the camera, whereas others are relatively close to the

camera. Even more, data collection was done at various times of the day, which can be observed through cast shadows spawned from various angles.

Keras-Tensorflow library is deployed to execute the GSHR-Net using parallel coding in Python. The processing speed of the proposed GSHR-Net depends on the machine used to execute the algorithm. For reference, the inference phase of GSHR-Net can achieve an average of 1.742 frames per second by using Intel i9 central processing unit. In contrast, using a Nvidia RTX 2080 Ti graphics processing unit (GPU), the GSHR-Net can be processed at a much faster rate of an average of 11.514 frames per second. Therefore, the GPU is used to speed up the computation of 80 epochs for each tested algorithm. Based on the utilized GPU with 11 GB of memory, a batch size of four images can be processed per cycle. Default optimizer settings are used to update the parameters through the Adam backpropagation method [27]. For GSHR-Net, the original input image is resized to 512 × 512 pixels to produce the output mask of apple locations, which is also in 512 × 512 pixels resolution. Two performance metrics are used to measure the capability of the proposed method, which are accuracy (Acc) and intersection over union (IoU). Both of the metrics are assessed through pixel-wise operation of a combination of true positive (Tr_{+ve}), true negative (Tr_{-ve}), false positive (Fa_{+ve}), and false negative (Fa_{-ve}) labels. For each tested pixel, it will be labeled as Tr_{+ve} if both GSHR-Net and annotated masks show an apple label, and it will be labeled as Tr_{-ve} if both GSHR-Net and annotated masks show a background label. For Fa_{+ve}, a pixel is detected by GSHR-Net as an apple class but the real annotated label is background. Finally, Fa_{-ve} is a situation where the pixel is detected by GSHR-Net as the background class but the real annotated label is the apple class.

$$Acc = \frac{\sum Tr_{+ve} + \sum Tr_{-ve}}{Tot_{pix}} \tag{1}$$

$$IoU = \frac{\sum Tr_{+ve}}{\sum Fa_{+ve} + \sum Fa_{-ve} - \sum Tr_{+ve}} \tag{2}$$

where Tot_{pix} is the total number of pixels in an image. The average of the Acc and IoU tested images, which are denoted as $MAcc$ and $MIoU$, respectively, will be the final reported performance of each tested algorithm. There are 11 state-of-the-art deep learning semantic segmentation algorithms that have been tested as the performance comparison benchmark: PSPNet [28], FCN [29], FC-DenseNet [24], SegNet [11], DeepLab V1 [30], DABNet [31], DeepLab V2 [25], HRNet [23], UNet [26], TernausNet [32], and DeepLab V3+ [33].

In the first phase of validation, nine variants of GSHR-Net have been validated to analyze the effect of various GSO block configurations. There are three variants of the GSO block, and each variant will have three settings that differ according to the number of groups in the group convolution. In theory, a more convolutional group will result in less dependency between various convolutional filters, whereby the relationship is restricted to a few input feature maps only. In addition, by utilizing more group operations, each of the filters is more likely to contribute to effective feature extraction. However, if only a single group is used, the convolutional filters can be trained by using all incoming feature maps that allow them to be updated with the most informative input. Yet, there is also a possibility in which only a few dominant filters contribute to the decision making. Therefore, there is a fine performance balance that must be chosen in extracting the best features with regard to the optimal number of groups.

According to Table 1, the best GSHR-Net configuration is GSHR-Net V3-2, which uses two sets of group convolution per one GSO block. GSHR-Net V3-2 returned the best $MIoU$ of 0.8045 with a relatively good $MAcc$ of 98.835%. In this work, the $MIoU$ metric is given the priority over the $MAcc$ metric in ranking all the proposed variants, because a good $MIoU$ is more important for apple segmentation, as such it is better to have a good overlapping mask between the predicted output and the annotated ground truth mask. However, if the $MIoU$ performance is similar for any two variants, then the $MAcc$ performance will supplement the ranking value by analyzing the best true detection for both positive and negative labels. It is also important to note that pixels

that belong to the apple regions are relatively fewer compared to the pixels that belong to the background. Therefore, the algorithm might return a good *MAcc* performance if it is trained bias for background detection. Hence, in terms of *MAcc*, the best performance is returned by the GSHR-Net V3-8 variant, which divides the feature maps into eight sets of group convolutions that were applied to both convolution layers. The recorded highest *MAcc* value by the GSHR-Net V3-8 variant is 98.861%, but its *MIoU* performance is the lowest among the variants, with only 0.7974. However, if the main concern is the model size, GSHR-Net V3-8 is the best candidate, as it requires the lowest memory capacity of just 5,506,304 parameters. A smaller model size is obtained by GSHR-Net V3-8 since the network applied the highest number of group convolutions, and the connections between the filters are limited to each particular group only. However, all variants can be considered as a lightweight network in general according to the criterion used in [34], whereby only GSHR-Net V2-2 uses more than 10 million parameters. Its total number of parameters is 10,621,184, which is slightly more than the minimum threshold of a lightweight network of 10 million parameters. Table 1 also concurred with our previous argument, whereby if more convolutional groups are utilized in the GSHR-Net, the network will require a lower number of total parameters. A lower set of parameter connections can be observed if the convolution kernels are separated into more groups, which explains the reduction in the number of network parameters.

For the first variant, the best segmentation performance was obtained by using eight groups of convolution, whereas for the second variant, the best segmentation performance was obtained by using four groups of convolution kernels. The third variant produced the best performance if only two groups of convolution are utilized. Hence, there is no clear pattern that can be deduced according to the optimal number of convolutional groups used in the GSHR-Net, as it depends on the variant itself, whereby each variant requires a different set of total convolution groups for the optimal performance. However, there is a relationship between a lower number of group convolutions and the total number of parameters that can be observed for the third variant of GSHR-Net. It is observable that the number of parameters reduces significantly as more groups are utilized because this third variant employed two layers of group convolution operator per GSO block compared to the other variants with just one layer. As a result, if a more convolutional group is employed, the number of parameters will also reduce by a larger margin compared to other variants. Therefore, a significantly less complex network will be produced to map the input image to the output mask of apple locations can be observed for GSHR-Net V3-8 compared to GSHR-Net V3-2. For the other variants, only one layer of group convolution is employed, and hence, the reduction in the total number of parameters is not too sudden. In summary, the proposed GSHR-Net still can be considered as a lightweight deep network in general.

Table 1. Performance comparison between variants of GSHR-Net.

Method	Trainable Parameters	*MAcc* (%)	*MIoU*
GSHR-Net V1-2	10,621,184	98.821	0.8026
GSHR-Net V1-4	9,598,208	98.859	0.7974
GSHR-Net V1-8	9,086,720	98.849	0.8029
GSHR-Net V2-2	10,621,184	98.809	0.8027
GSHR-Net V2-4	9,598,208	98.844	0.8029
GSHR-Net V2-8	9,086,720	98.850	0.8001
GSHR-Net V3-2	8,575,232	98.835	0.8045
GSHR-Net V3-4	6,529,280	98.847	0.8013
GSHR-Net V3-8	5,506,304	98.861	0.7974

For the second phase validation, the performance of the optimized GSHR-Net is compared with the other state-of-the-art semantic segmentation algorithms. According to the performance results in Table 2, GSHR-Net V3-2 returns the best $MIoU$ compared to the other benchmarked algorithms. Furthermore, GHSHR-Net V3-2 uses the lowest memory space compared to the other networks, with just 8,575,232 parameters. The performance improvement of GSHR-Net can be pointed out clearly when it is compared to the HRNet, another model that uses the same design philosophy of a high-resolution semantic segmentation network. GSHR-Net V3-2 recorded an improvement of 2.71% in terms of $MIoU$, which is contributed by the addition of the GSO block. The group operation has forced all the convolutional filters in the GSHR-Net to learn the required feature in a more effective manner, while the shuffle operation breaks off any dominant flow between the group, which results in better segmentation performance. In fact, among all the benchmarked methods, only GSHR-Net V3-2 exceed the threshold of 0.8 for $MIoU$, coupled with the best $MAcc$.

Table 2. Apple recognition performance comparison between the proposed method and state-of-the-art benchmarks.

Method	Trainable Parameters	$MAcc$ (%)	$MIoU$
PSPNet	27,838,400	27.830	0.1465
FCN	134,393,428	97.490	0.5855
FC-DenseNet	14,594,658	97.999	0.5963
SegNet	29,444,166	97.591	0.6426
DeepLab V1	28,890,946	98.016	0.6513
DABNet	25,753,565	98.086	0.6802
DeepLab V2	71,419,720	98.137	0.6869
HRNet	12,667,136	98.671	0.7827
UNet	31,032,834	98.563	0.7867
TernausNet	22,927,426	98.750	0.7879
DeepLab V3+	41,051,088	98.778	0.7903
GSHR-Net V3-2	8,575,232	98.835	0.8045

The worst performance was returned by PSPNet with an $MIoU$ of 0.1465. The low performance can be attributed to the ineffective pyramid pooling scheme, as the size of the apple is relatively low compared to the imaging size. Hence, only the lowest resolution feature maps will be able to extract the features effectively, whereas the other bigger parallel paths do not learn many unique attributes of the apples. Furthermore, PSPNet also tends to require more training data in fitting the model optimally during the training phase because its network design is deep in nature through the usage of ResNet as the encoder module. There are four networks that are able to produce $MIoU$ performance of more than 0.7: DeepLab V3+, TernausNet, UNet, and HRNet. These four networks also produced relatively good $MAcc$ of approximately 98%, but all of them are not in the lightweight network category. The biggest network among them is DeepLab V3+, which is 4.7 times bigger compared to GSHR-Net V3-2, whereas the smallest network among them is HRNet, which is 1.4 times bigger compared to GSHR-Net V3-2. It is interesting to note that the reduction in model size between GSHR-Net V3-2 and HRNet is due to the usage of group and shuffle operators, whereby the total number of convolutional filters is the same for both networks.

Therefore, apart from having the best $MIoU$ and relatively good $MAcc$, the proposed GSHR-Net V3-2 requires the least memory capacity, which is perfect for a mobile-based application. Among the tested algorithms, DeepLab V2 requires the largest memory capacity with 71,419,720 parameters, but only produced an $MIoU$ of 0.6869. Even though more parameters will allow the network to establish a more complex relationship mapping between the input image and output mask, the network design architecture also plays an important role in order to extract the best mapping. Hence, a larger model such as DeepLab V2 does not necessarily return a good segmentation mask compared to a much lighter

model. Therefore, the usage of GSO block inside the proposed algorithm managed to reduce mapping dependency on a few dominant filters by allowing all filters to contribute effectively for a better segmentation output. DABNet also suffers from the same problem as PSPNet, whereby the multi-scale module does not really play an effective role in extracting the apple regions. The same issue of small apple size relative to the frame size that causes the up-sampled scale through atrous convolution does not provide a good feature extraction scheme. For FCN, the segmentation performance is low with an $MIoU$ of just 0.5855, which can be attributed to the simple network design that does not extract the feature effectively. It is just a stack of convolutional operators on the encoder side with a few up-sampling processes on the decoder side. Hence, not many complex relationships can be learned effectively by the network to extract the apple regions. Figure 6 shows some output masks of the recognized apples for top six algorithms.

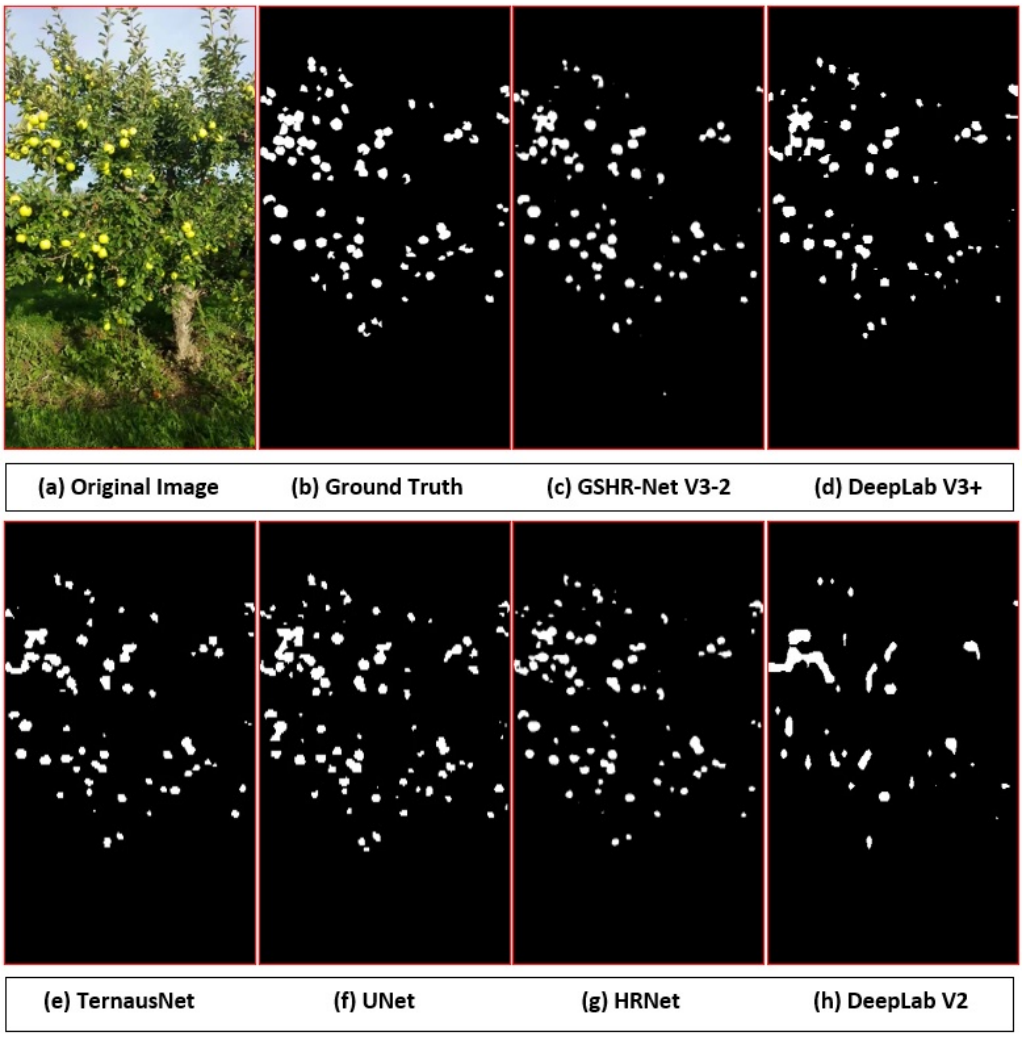

Figure 6. Three variants of group and shuffle operator configurations.

5. Conclusions

In conclusion, the proposed GSHR-Net has managed to deliver the best performance in recognizing the apple locations automatically. GSHR-Net V3-2 with two sets of group convolution applied to both layers in the GSO block is the best variant with the highest $MIoU$ of 0.8045. The proposed network can also be categorized as a lightweight model with a total parameter count of just 8,572,232, which makes it suitable for mobile-based applications. The integration of the GSO block managed to reduce the probability of a few dominant filters, whereby the feature extraction process was processed in a smaller-size group, which is then shuffled to break the dominant patterns. As a result, each group

of filters will be able to contribute more effectively to extracting the unique features of the apple regions that have been captured in various imaging conditions. For future work, the efficiency of an automated apples harvesting system can be further improved by enhancing the capability of the deep network architecture to cater to more challenging scenes. The multi-scale unit can be embedded to improvise the network capability in handling apples of various sizes and colors, whereas the attention mechanism can be integrated to give more weight to the regions that are more probable to be the apple trees, rather than the sky regions. Training dataset diversity can also be improved through the usage of the conditional generative adversarial network, which can create synthetic data that cover a wider spectrum of probable cases.

Author Contributions: Conceptualization, M.A.Z. and S.R.A.; software, M.A.Z. and S.R.A.; formal analysis, M.A.Z. and S.R.A.; writing—original draft preparation, M.A.Z., A.M.M. and A.H.S.; writing—review and editing, M.A.Z., A.M.M. and A.H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universiti Kebangsaan Malaysia under Dana padanan Kolaborasi with a grant number DPK-2021-012 and Ministry of Higher Education Malaysia with grant number FRGS/1/2019/ICT02/UKM/02/1.

Data Availability Statement: All images and the corresponding annotated ground truth masks can be obtained from http://rsn.cs.umn.edu/index.php/MinneApple, accessed on 1 February 2022 .

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

Abbreviations

The following abbreviations are used in this manuscript:

RBF	Radial Basis Function
GSO	Group and Shuffle Operators
ReLU	Rectified Linear Unit
KNN	K-Nearest Neighbor
MIoU	Mean Intersection over Union
MAcc	Mean Accuracy
CNN	Convolutional Neural Networks
SVM	Support Vector Machines
ANN	Artificial Neural Networks
GPU	Graphics Processing Unit
PNG	Portable Network Graphics

References

1. Fruit, Edible; Apples, Fresh Imports by Country in 2019. 2020. Available online: https://wits.worldbank.org/trade/comtrade/en/country/ALL/year/2019/tradeflow/Imports/partner/WLD/product/080810 (accessed on 25 March 2022).
2. Akhtera, F.; Azizb, S.; Jalalc, F. Effective Bioconversion of Locally obtained Apple Waste into Citric Acid using Aspergillus Niger (NRRL 567). *J. Kejuruter.* **2022**, *34*, 317–323.
3. Workman, D. Apples Exports by Country. 2021. Available online: https://www.worldstopexports.com/apples-exports-by-country/ (accessed on 25 March 2022).
4. Wu, T.; Wang, Y.; Yu, C.; Chiarawipa, R.; Zhang, X.; Han, Z.; Wu, L. Carbon sequestration by fruit trees-Chinese apple orchards as an example. *PLoS ONE* **2012**, *7*, e38883. [CrossRef] [PubMed]
5. Tougeron, K.; Hance, T. Impact of the COVID-19 pandemic on apple orchards in Europe. *Agric. Syst.* **2021**, *190*, 103097. [CrossRef]
6. Bertone, E.; Venturello, A.; Leardi, R.; Geobaldo, F. Prediction of the optimum harvest time of 'Scarlet' apples using DR-UV–Vis and NIR spectroscopy. *Postharvest Biol. Technol.* **2012**, *69*, 15–23. [CrossRef]
7. Yang, Q.; Chen, C.; Dai, J.; Xun, Y.; Bao, G. Tracking and recognition algorithm for a robot harvesting oscillating apples. *Int. J. Agric. Biol. Eng.* **2020**, *13*, 163–170. [CrossRef]
8. Zulkifley, M.A.; Mustafa, M.M.; Hussain, A. On improving CAMSHIFT performance through colour constancy approach. In Proceedings of the 2012 International Conference on Computer & Information Science, Kuala Lumpur, Malaysia, 12–14 June 2012; pp. 375–378.
9. Zulkifley, M.A.; Moran, B.; Rawlinson, D. Robust Foreground Detection: A Fusion of Masked Grey World, Probabilistic Gradient Information and Extended Conditional Random Field Approach. *Sensors* **2012**, *12*, 5623–5649. [CrossRef] [PubMed]

10. Humadi, A.; Nazarahari, M.; Ahmad, R.; Rouhani, H. In-field instrumented ergonomic risk assessment: Inertial measurement units versus Kinect V2. *Int. J. Ind. Ergon.* **2021**, *84*, 103147. [CrossRef]
11. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
12. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
13. Ji, W.; Zhao, D.; Cheng, F.; Xu, B.; Zhang, Y.; Wang, J. Automatic recognition vision system guided for apple harvesting robot. *Comput. Electr. Eng.* **2012**, *38*, 1186–1195. [CrossRef]
14. Zhou, R.; Damerow, L.; Blanke, M.M. Recognition Algorithms for Detection of Apple Fruit in an Orchard for early yield Prediction. *Precis. Agric.* **2012**, *13*, 568–580. [CrossRef]
15. Karim, R.A.; Zakaria, N.F.; Zulkifley, M.A.; Mustafa, M.M.; Sagap, I.; Md Latar, N.H. Telepointer technology in telemedicine: A review. *Biomed. Eng. Online* **2013**, *12*, 21. [CrossRef] [PubMed]
16. Jiang, G.Q.; Zhao, C.J. Apple recognition based on machine vision. In Proceedings of the IEEE International Conference on Machine Learning and Cybernetics, Xi'an, China, 15–17 July 2012; Volume 3, pp. 1148–1151.
17. Syal, A.; Garg, D.; Sharma, S. Apple fruit detection and counting using computer vision techniques. In Proceedings of the 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2014; pp. 1–6.
18. Tanco, M.M.; Tejera, G.; Di Martino, M. Computer Vision based System for Apple Detection in Crops. In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Madeira, Portugal, 27–29 January 2018; pp. 239–249.
19. Nguyen, T.T.; Vandevoorde, K.; Kayacan, E.; De Baerdemaeker, J.; Saeys, W. Apple detection algorithm for robotic harvesting using a RGB-D camera. In Proceedings of the International Conference of Agricultural Engineering, Zurich, Switzerland, 6–7 October 2014.
20. Yongting, T.; Jun, Z. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Comput. Electron. Agric.* **2017**, *142*, 388–396.
21. Awad, F.H.; Fadhel, M.A.; Alheeti, K.M.A.; Al-Shamma, O.; Alzubaidi, L. Enhancing Apple Maturation Recognition Performance Based on Field Programmable Gate Array Implementation. *J. Southwest Jiaotong Univ.* **2019**, *54*. [CrossRef]
22. Abdani, S.R.; Zulkifley, M.A.; Zulkifley, N.H. Group and Shuffle Convolutional Neural Networks with Pyramid Pooling Module for Automated Pterygium Segmentation. *Diagnostics* **2021**, *11*, 1104. [CrossRef] [PubMed]
23. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef] [PubMed]
24. Jegou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1175–1183. [CrossRef]
25. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
26. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
27. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2014.
28. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239. [CrossRef]
29. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
30. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
31. Li, G.; Yun, I.; Kim, J.; Kim, J. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv* **2019**, arXiv:1907.11357.
32. Iglovikov, V.; Shvets, A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv* **2018**, arXiv:1801.05746.
33. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 833–851.
34. Zulkifley, M.A.; Abdani, S.R.; Zulkifley, N.H. COVID-19 Screening using a Lightweight Convolutional Neural Networks with Generative Adversarial Network Data Augmentation. *Symmetry* **2020**, *12*, 1530. [CrossRef]