Table of Contents

Section 1: Introduction

Fruit counting is an imperative part of the planting industry to estimate agricultural production. However, manual counting takes longer time and more labor, it is needed to apply machine vision to save expenses and time. Due to various kinds of fruits, we choose apples to implement the counting. Conventional and machine learning methods are applied to count the number of apples. Image processing is the core of these methods and is proven to be more effective and accurate than manual counting. Thus, it is necessary to apply these methods in a reasonable and productive way to achieve great identification and counting fruit for better understanding of image processing.

For the conventional part, it is based on fundamental image processing to extract the features to identify the fruit without deep learning. Firstly, we used the color thresholding to separate apple from the background. Then, turning the image into grey scale makes it simple to process and use the closing and contour detection to eliminate extra small pixels. Finally, the CHT (Circular Hough Transform) was applied to fit circle for every individual apple to count the numbers. In a word, based on the color and shape features of the apple, we use image segmentation, morphological operation, and feature extraction to implement the conventional method.

For the machine learning method, we present our approach by using the Convolutional Neural Network (CNN) to classify apple images and count the number of apples. To achieve this, the model was trained on a dataset of labeled apple images from MinneApple (Haeni, Roy and Isler, 2019) using ResNet18 as a base model. After that, we evaluated its performance on a separate dataset of images and found that it was able to determine the number of apples with high accuracy. In this report, we will discuss the details of our approach, including the dataset we used, the training and testing procedures, and the results of our experiments. The comparison of the apple counting task between the conventional method and machine learning was also conducted and shown in the Results section.

Section 2: Related works

Section 2.1 - Conventional Method

Even though image processing is a precise way to identify and count the apple, it still has many challenges that affect the result：

- Occluded apple may be ignored by machine
- Overlapped apple may be counted less
- False detection, recognizing leaves or branches as apple

With the development of the machine vision, there is much research that has implemented better methods of image processing to achieve more accuracy about fruit recognition.

In a recent study (Behera *et al.*, 2018), they presented a paper on apple fruit detection and counting using color thresholding and CHT. They applied L*a*b color space to extract apples from the background and converted it into a binary image, then used CHT to find and count apples. The results had 93.10% accuracy in densely occluded apples and 100% accuracy in non-occluded and partially occluded apples.

Patel, Jain and Joshi, (2012) have researched automatic segmentation and yield measurement of counting fruit. He applied Gaussian low pass filter and converted image to L*a*b color space for pre-processing. And he used morphological operation to remove noise and edge detection which can be fitted circle to detect individual fruit. The results can achieve high accuracy, but the algorithm ignored some small apples and counted fruits clusters as one fruit in some cases.

Rong Zhou proposed a color difference algorithm for apple recognition and used two different color models to segment mature apple images. In the green stage when the apple is not mature, the R-B (red-blue) algorithm is used to identify the apple; in the mature stage, the R-G (red-green) algorithm is used to identify the apple. The mature apple fruit is larger, and the color difference between red leaves and green leaves is larger, so the algorithm can identify mature apples more accurately. Because a large number of apples in the image are obscured by leaves or fruit overlap is serious, there is a difference between the number of fruits detected by fruit counting algorithm and the number of fruits counted manually.

Section 2.2 – Machine Learning

Machine vision has been used in an agricultural field for yield estimation. The manual method of yield estimation is “time-consuming, labor-intensive, and inaccurate” (Wang *et al.*, 2013). With accurate and automatic yield estimation, it can help “growers improve fruit quality and reduce operating cost” (Wang *et al.*, 2013). The uses of machine vision in agriculture are, for example, “object recognition, classification, and counting” (Rahnemoonfar and Sheppard, 2017a). Regarding the use of deep learning, the accuracy and success “largely depend on the availability of a large amount of training samples” (Rahnemoonfar and Sheppard, 2017a). The challenges of a fruit counting task by computer vision include illumination variance, fruit occlusion, and overlapping between fruits. In the study by Rahnemoonfar and Sheppard, 2017, the deep learning network which was developed by using the Inception-ResNet model was proposed to “[count] objects without detecting them”

After assessing and analyzing different models, we selected the ResNet18 model for the apple counting task in our study. ResNet18 is a convolutional neural network (CNN) architecture which was introduced in the paper "Deep Residual Learning for Image Recognition" (He *et al.*, 2015). The paper describes the ResNet18 as a variant of the ResNet architecture consisting of 18 layers, which is known for its ability to train very deep neural networks. The model is widely adopted due to several reasons:

Depth: it is a relatively deep neural network, which means it is capable of learning complex relationships in the data.

Skip connections: The use of skip connections allows the network to bypass one or more layers and directly propagate the input to the output alleviating the problem of vanishing gradients.

Performance & Ease of use: The model has shown good performance on a variety of tasks and is relatively easy to implement and fine-tune for specific tasks.

The research by Rahnemoonfar and Sheppard (2017) applies an Inception-ResNet CNN to directly enumerate red fruits from simulated data. They employed synthetic data to train a network on 728 classes to count red tomatoes.

Section 3: Data acquisition and datasets

Section 3.1 – Conventional Method

The data of apples was collected from Google and Baidu Image. To make sure the authentication of the test, the apple images were all on-tree to match the situation of harvest for the planting industry. And the aim of apples species which need counting were ripe and red (we only counted red ones due to the limitation of the conventional method).

We separated the images into three categories: non-occluded/partially occluded apples, densely occluded apples, and apples clusters to test whether the conventional method has high accuracy on these images. The illustration about these categories is shown below:

- Non-occluded/partially occluded apples image: there are 10 different images which are the simplest level to test the method's fundamental function. Some apples in the image were not/partially obstructed by leaves, branches, or other apples, which may cause error of identification.
- Densely occluded apples: there are 5 different images which are harder to identify and count. Most apples in the image were densely obstructed by the background or other apples, which may reduce the accuracy of the counting furthermore. This category is to test whether the method can detect hiding apples which do not have a perfect round shape and avoid the false detection of the background.
- Apples clusters: there are 5 different images which are challenging to identify and count. Most apples were overlapped to form a cluster, which can be very confusing to the machine. This category is to test whether the method can count overlapped apples in a cluster.

Section 3.2 – Machine Learning

The data set in this paper is from Haeni, Roy and Isler (2019) in the Horticultural Research Center (HRC) at the University of Minnesota. Overall, there are 1526 apples and 26 trees in this data set, the ripeness of the apple is reflected in the color of the skin, which changes from green to red that indicates ripeness. However, due to limitations in lighting conditions and time dimensional, the color of the apple images will be displayed differently than under standard conditions. The dataset contains rich information on color dimensions, this perhaps generates some errors. In the experiment by Häni, Roy and Isler (2020), the yellow leaves sometimes can be detected as an apple by networks. In addition, in the oil palm maturity test (Septiarini *et al.*, 2021), the uneven color variation also has an impact on the classification of half-ripe and ripe. Thus, this may have an impact on the accuracy of the recognition after the final model training. Through comparing and training four network models (VGG-16, ResNet50, Inceptionv3, EfficientNet), the residual network model is used to implement apple counting. After training, the images will be grouped into seven categories. This is the possible number of apples in each image, from zero to six. Moreover, there are three types of data sets amount the training: training set, verification set, and test set. The training set has 64590 images, and 3395 images are in the verification set. This is because to have a result as accurate as possible, the model needs more extensive training. In addition, the training will continue for 10 rounds.

Section 4: Methodology

Section 4.1 – Conventional Method

A. Color thresholding and segmentation

Color is the most intuitive feature of the apple. And red color is easy to distinguish from the background. Therefore, we need to choose a color space to extract the red color via thresholding. There are three common color spaces: RGB (red, green, blue), HSV (hue, saturation, value) and L*a*b color space. To achieve better performance, we chose HSV color space which contains more detailed information about the color.

There are three channels in HSV space which are illustrated in Figure 1: HSV color spaceFigure 1. The hue means all the color in the entire spectrum. As for the saturation, it means the purity and strength of the color (higher means brighter). For the value, it stands for the brightness of the pixels. Thus, HSV space can detect red apples in different brightness and strength to make the segmentation more accurate.

Figure 1: HSV color space

We also need to convert it into binary image using Gaussian blur to reduce the noise and Otsu threshold to separate black and white pixels. As a result, we can get a binary image whose black area stands for apples and white area means the background.

B. Morphological operations

There are two parts about this section: Closing and small object removal. As for the closing part, the dilation followed by erosion to refill pixels into black area. Then, we get the contour of the apples, calculating the maximum size and average size of the contours. The smallest contours must be beyond 10% of maximum size. If the contour is less than 10% of the average size, it will be removed to reduce the false count of the extra pixels.

C. CHT

We apply the Circular Hough Transform as a feature extraction to identify the circle shape. This method can detect multiple unknown radius of different circles. The CHT (Circular Hough Transform) will produce accumulator metrics which are contributed by iterations of possible radius. The steps of the CHT are illustrated below:

Step 1: Set the maximum and minimum value for possible radius.

Step 2: Draw the contours and fit the circle to them.

Step 3: Screen out the qualified circle which is beyond the 10% of average radius.

Step 4: Repeat the above steps and update the new circle until it finishes all the contours.

In summary, the convention method is divided into few steps below:

Step 1: Pre-process the image with color thresholding and Gaussian blur into binary image.

Step 2: Apply closing the binary image to fill the small blank of apples and remove small black pixels to avoid redundant counting.

Step 3: Use the CHT to fit the circle to the contour of apples and count the numbers of circles with qualified radius.

Section 4.2 – Machine Learning

Deep learning models can facilitate the automated calculation process in yield estimation and can precisely count fruits subsequent to being trained. A variety of deep learning models can be used to leverage transfer learning techniques by reutilizing the pre-trained model as a base model in our task. In order to ascertain the most appropriate model for our apple enumeration task, we assessed and contrasted the performances of four distinct deep learning models:

- VGG-16
- EfficientNet
- Inceptionv3
- ResNet18

All of these four pre-trained Image Classification models are considered to be state-of-the-art (SOTA) and are widely used in industry. Each of these models has its own set of advantages and disadvantages depending on the specific characteristics of the dataset and the desired trade-offs between accuracy, efficiency, and speed. After data preprocessing and data slitting of the MinneApple dataset, we trained and fitted all four models to evaluate their accuracy and associated loss metrics. Table 1 shows the recorded metrics for these models.

Table 1: Accuracy metrics of four models

Approach	Accuracy %
VGG-16	84.7
EfficientNet	80.0
Inceptionv3	60.1
ResNet18	97.7

Firstly, VGG-16 was tested which has a relatively simple architecture and is comparatively rapid to train. However, the accuracy it yielded in our trial was approximately 84.7%. Secondly, EfficientNet which is a family of models that combines various architectures, it requires more computational resources to train and execute than VGG-16, and the accuracy it scored in our trial was close to 80%. Inceptionv3 has a more complex architecture than VGG-16 and demands more computational resources to train and run. The accuracy metric it achieved in our trial was approximately 60%. ResNet-18 is a deep residual neural network with an intricate architecture. Despite the high computing power needed to train and run it, it showed the highest accuracy of around 97.7% in our trial. After evaluating and comparing the metrics of these four models, our investigation determined that the ResNet18 model presented the most promising results for the apple counting task in our study. Based on our analysis of the results we could deduce that as the depth of the neural network increases, the training set loss increases. The introduction of the residual component enables the neural network to become more profound, and thus more efficient. Compared to the serial architecture of a standard network, the residual unit adds a skip mapping technique, which adds the input and output together, making up for the feature information lost during the convolution process.

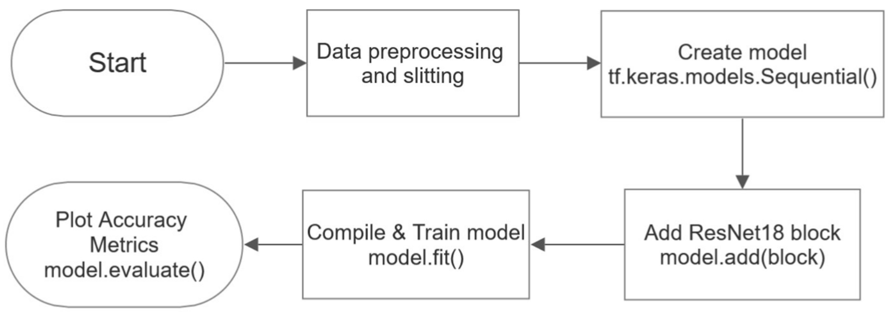


Figure 2: Deep learning method workflow

In order to train the ResNet18 model to count apples, we utilized the MinneApple dataset of images that contain apples from an assortment of sources. Figure 2 illustrates the workflow our approach implements. Initially, we divided the pictures from the dataset into various classes depending on the quantity of apples the image contained to be utilized as training data. Subsequently, the dataset was split into a training set and a validation set which were then utilized to train the ResNet18 model via the Adam optimizer and a learning rate of 0.0001.

Section 5: Experiment and Implementation

Section 5.1 – Conventional Method

A. Data Collection

Due to some limitations, we were unable to take pictures of apple trees in the field, so we selected a total of 20 high-resolution images from reliable sources for the Apple Counting experiment. These images can be roughly divided into four types: non-occluded, partially occluded, densely occluded and cluster. There are two bases for classification: one is whether there is a distance between the fruits and whether they are close to each other; the other is whether the fruits are blocked and whether the blocked parts occupy a large proportion. We know that the larger the part of the fruit is covered, the more difficult the detection will be and the higher the probability of error will be, so the detection results of densely occluded images can best show the accuracy of the algorithm.

B. Image Processing

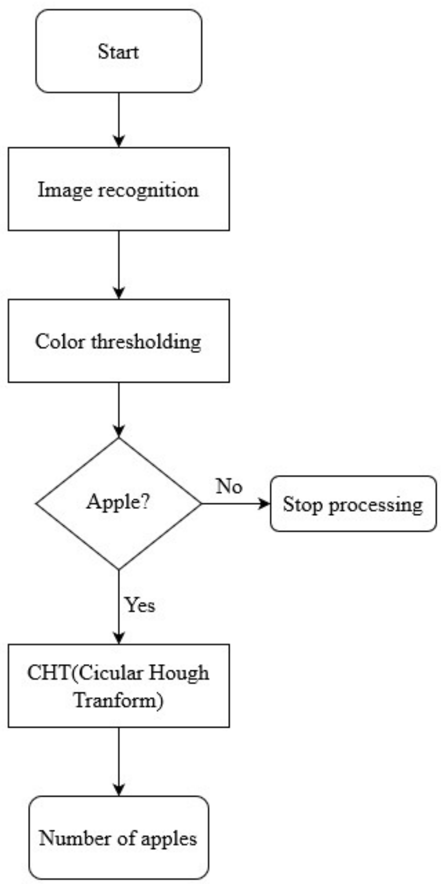


Figure 3: Counting process of apple fruit

We developed an image processing algorithm and tested it with 20 images from the collected images. Figure 3 is the flow chart of the image processing algorithm we developed.

First, after importing the image, convert the color model used in the image from RGB to HSV, that is, Hue, Saturation, and Value, where Value refers to the brightness of the color. The reason for the conversion to the HSV color model is that under the HSV color model, the contrast between the red of the apple and the green of the leaf is stronger, so that the probability of recognition error in the algorithm is relatively reduced when processing the image. Then customize the lower limit and upper limit of the color of a red mask, and use the bitwise_and() function to filter out the blue and green pixels in the image, leaving only red pixels. However, this operation also tends to increase noise. In order to suppress this noise, after screening, the processed image is converted into a grayscale image, and then processed by Gaussian blur, which can also be regarded as using a low-pass filter to filter the noise. After filtering the noise, these grayscale images are converted into binary images by Otsu's threshold method, which determines the best threshold for separating image pixels into foreground and background classes with the minimum intra-class variance. In order to remove the unwanted small objects that are divided into foreground, we use the contourArea() function to get the contour area of the image, and then filter out the regions whose contour area is less than 0.1 times the contour area of the image obtained by the function, and then use the mask to filter out these areas from the image, so as to reduce the interference in the process of apple detection.

If the fruit is not screened from the image, the image processing is stopped; if the fruit is screened, the next step is to apply Circular Hough Transform (CHT) iteratively within a specified radius in order to identify circular objects that may be apples. Finally, we get the number of apples in the image.

Section 5.2 – Machine Learning

A. Python Environment for Image Classification

Table 2 provides details on the PC, Python IDE/code editor, and core python packages used in the report.

Table 2: Python Environment for Image Classification

PC Hardware	Central Processing Unit (CPU)	Intel Core i5-10400F
	RAM	16 GB
	Graphics Processing Unit (GPU)	GeForce NVIDIA RTX 2060
	Operating System (OS)	Windows 10
Python IDE/Code Editor	Visual Studio Code	VS code is used to run Python code in a local machine
Python Packages/Libraries	Tensorflow	Tensorflow is the main library to create and train data for image classification
	Matplotlib	Matplotlib visualizes the accuracy of the CNN model in terms of 2D graph plotting
	pathlib	pathlib is used to deal with the file path in a local machine
	random	Random is used to shuffle data when training the model to enhance the generalization and prevent overfitting
	Pandas	Pandas is used to read CSV-formatted label files

B. Data Preprocessing

Images from the MinneApple dataset do not have the same dimension, therefore, we first need to resize all the images into 64x64 pixels which have a value between 0 and 255. Then, the rescaling process is done to reduce the amount of data and speed up the optimization process. The value is divided by 255 resulting in a range of numbers between 0 and 1. The preprocessing step is written in the "load_preprocess_image" function following the flowchart as shown in Figure 4.

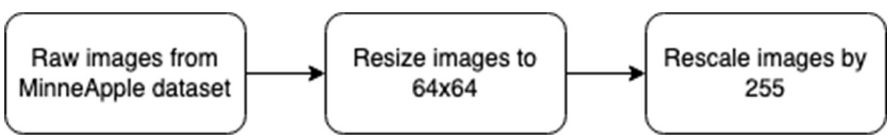


Figure 4: Preprocessing Steps for Image Classification

C. Data Splitting

In the experiment, the dataset is divided into three parts: training dataset, validation dataset, and testing set. Each dataset must have images and labels since the method used in this report is supervised learning. The MinneApple dataset provides images and labels for training and validation datasets. The labels are given in the form of a txt file. "train_ground_truth.txt" and "val_ground_truth" are file names provided by MinneApple with CSV format which can be read by the read_csv method from Pandas library. The information contains the image file name with the correct number of apples for that image.

However, the testing dataset does not come with a label file. Therefore, we created the testing dataset by selecting images and labels from the validation dataset provided by MinneApple.

D. Convolution Neuron Network Model

The model used in the experiment is based on Resnet18 architecture. The dataset is divided into seven classes as we categorized the images into 0, 1, 2, 3, 4, 5, and 6 apple classes according to the ground truth labels. Therefore, the output layer contains seven neurons. The input shape is 64x64 with 3 channels since most of the original images have the dimension of 64x64 pixels. Adam optimizer is used to train the model because Adam automatically adjusts the learning rate instead of manually fixing it. The loss function used along with the Adam optimizer is the Sparse Categorical Cross Entropy. The Sparse Categorical Cross Entropy loss function is often used and suitable for an image classification task in which each image contains a single label. The number of epochs is the number of times the model is trained over the training set. The greater number of epochs usually yield a better result. However, the performance will stop or increase at a lower rate which should be the optimum number of epochs.

Section 6: Results and Evaluation

Section 6.1 – Conventional Method

A total of 20 images were selected for apple counting experiment, including 5 images of non-occluded, partially occluded, densely occluded and cluster respectively. Convert the input RGB image to HSV color space for threshold segmentation, then convert the image into a binary image, and use CHT to fit the circle of a single fruit in different visual environments.

Figure 5: Non-occluded apple & CHT image

Figure 6: Partially occluded apple & CHT image

Figure 7: Densely occluded apple & CHT image

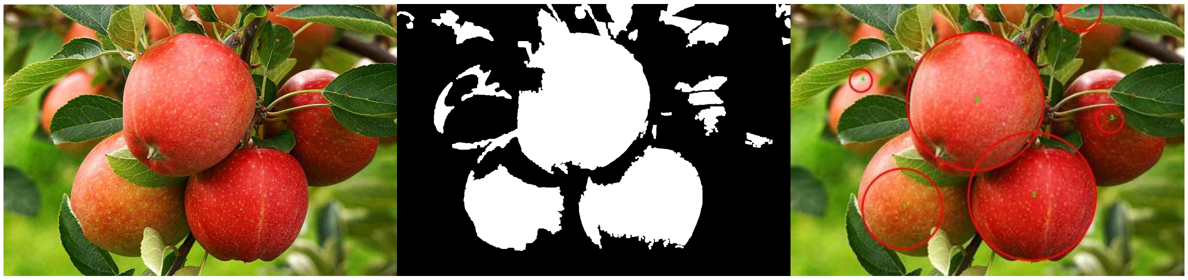


Figure 8: Cluster apple & CHT image

In the first case, the apples are not shielded from each other, there is a distance between the apples, and the apples are not obscured by leaves. In this case, the number of apples calculated by the algorithm is no different from the manual count, and the algorithm can achieve 100% accuracy.

In the second case, apples will be in contact with each other, individual apples are obscured by other apples or branches, leaves, etc., covering about 20% to 30% of the apple area. In this case, the algorithm also achieves a relatively high accuracy, although there will be errors, but the error is not large, in the acceptable range.

In the third case, apples will encounter each other, and some or even most apples will be obscured by other apples or branches and leaves, accounting for 50% and 60% of the apple area. In this case, the accuracy of the algorithm is also considerable, although sometimes there will be 2-3 errors, the larger occluded area of the apple will not be detected, but the overall error is also relatively small.

In the fourth case, the apples are in close contact with each other, showing a dense clump shape, which will partially block each other, but the occlusion does not account for a large part of the apple area, which can be regarded as the second special case. In this case, the results given by the algorithm are also in line with expectations.

Table 3: Accuracy for detection & counting

Input image	Type of image	No. of apples count manually	No. of apples count by the algorithm	Percentage of accuracy	Accuracy as per visual situation	Accuracy
1	Non-occluded	2	2	100%	100%	96.07%
2		4	4	100%		
3		5	5	100%		
4		3	3	100%		
5		4	4	100%		
6	Partially occluded	6	5	83.33%	94.64%	
7		8	9	88.88%		
8		3	3	100%		
9		4	4	100%		
10		5	5	100%		
11	Densely occluded	6	5	83.33%	92.98%	
12		14	15	93.33%		
13		4	4	100%		
14		4	4	100%		
15		15	17	88.24%		
16	Cluster	5	5	100%	96.67%	
17		4	4	100%		
18		6	5	83.33%		
19		7	7	100%		
20		8	8	100%		

Table 4: False detection of occluded apple

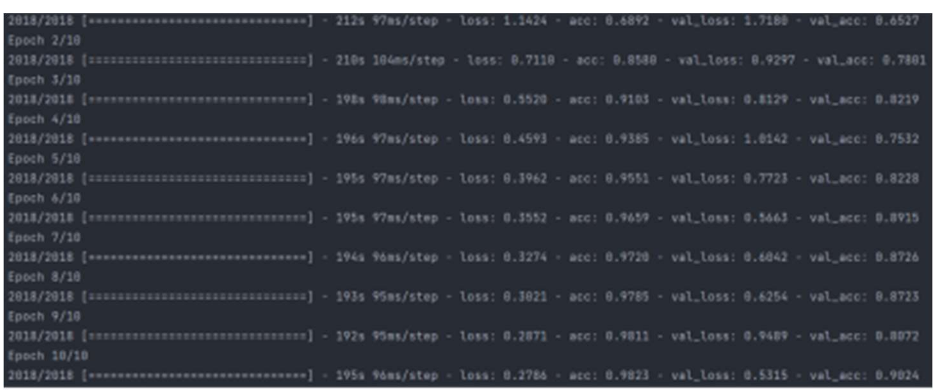
Input image	Manual count	Automatic count	No. of false detection	Error in %	Average error in %
1	6	5	1	16.67	5.53
2	8	9	1	12.5	
3	3	3	0	0	
4	4	4	0	0	
5	5	5	0	0	
6	6	5	1	16.67	
7	14	15	1	7.14	
8	4	4	0	0	
9	4	4	0	0	
10	15	17	2	13.33	
11	5	5	0	0	
12	4	4	0	0	
13	6	5	1	16.67	
14	7	7	0	0	
15	8	8	0	0	

The algorithm is applied to four different apple images, and the experimental accuracy of apple counting can reach 96.07%, as shown in Table 3. Table 4 shows all the apple count results under occlusion, although not 100% accurate, but the error is small, controlled at 5.53%.

This algorithm can detect most apples with protruding circular edges. In most cases, a single apple in the cluster can be accurately detected, and the error mainly occurs when the apple is highly obscured by another apple, and the highly obscured apple may not be recognized. This error is due to the limitation of CHT, that is, when the distance is very close, the identified apples are not counted.

Section 6.2 – Machine Learning

In the training and validation of a neural network model, three data sets are first used: the training set, the validation set, and the test set. The training set is used to train the model and usually has a very large amount of data. It contains 64590 images. The data in the validation set is to be similar to the test set and is usually obtained in the same way. However, the amount of data in the validation set is much smaller than that in the test set, and there are only 3395 images in this experiment. By comparing the changes in the training and validation sets during the training process, we can derive a picture of how the model is trained. In this experiment, we used the ResNet18 model and performed a total of 10 rounds of training which means the epoch of the training is 10. In addition, another parameter called batch size also affects the training effect, here the batch size is set to 32. The detailed training results and training set, and the variation in accuracy of the validation set are shown below in Figure 9.



```
2018/2018 [==============================] - 212s 97ms/step - loss: 1.1424 - acc: 0.6892 - val_loss: 1.7180 - val_acc: 0.6527
Epoch 2/10
2018/2018 [==============================] - 210s 104ms/step - loss: 0.7110 - acc: 0.8580 - val_loss: 0.9297 - val_acc: 0.7801
Epoch 3/10
2018/2018 [==============================] - 198s 98ms/step - loss: 0.5520 - acc: 0.9103 - val_loss: 0.8129 - val_acc: 0.8219
Epoch 4/10
2018/2018 [==============================] - 196s 97ms/step - loss: 0.4593 - acc: 0.9385 - val_loss: 1.0142 - val_acc: 0.7532
Epoch 5/10
2018/2018 [==============================] - 195s 97ms/step - loss: 0.3962 - acc: 0.9551 - val_loss: 0.7723 - val_acc: 0.8228
Epoch 6/10
2018/2018 [==============================] - 195s 97ms/step - loss: 0.3552 - acc: 0.9659 - val_loss: 0.5663 - val_acc: 0.8915
Epoch 7/10
2018/2018 [==============================] - 194s 96ms/step - loss: 0.3274 - acc: 0.9720 - val_loss: 0.6042 - val_acc: 0.8726
Epoch 8/10
2018/2018 [==============================] - 193s 95ms/step - loss: 0.3021 - acc: 0.9785 - val_loss: 0.6254 - val_acc: 0.8723
Epoch 9/10
2018/2018 [==============================] - 192s 95ms/step - loss: 0.2871 - acc: 0.9811 - val_loss: 0.9489 - val_acc: 0.8072
Epoch 10/10
2018/2018 [==============================] - 195s 96ms/step - loss: 0.2786 - acc: 0.9823 - val_loss: 0.5315 - val_acc: 0.9024
```

Figure 9

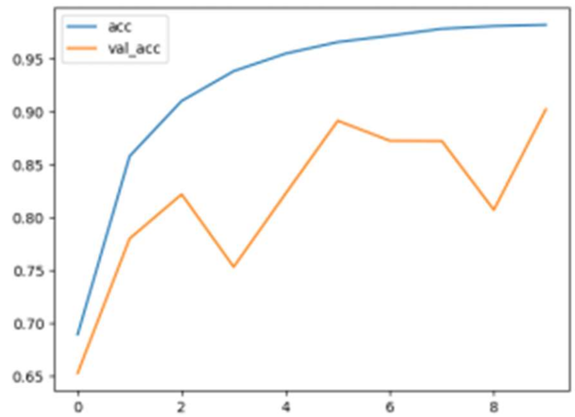


Figure 10: Result of ResNet

The results of the validation set are a test of the generalizability of the model. If in the image, the training set loss decreases and the validation set loss increases or remains unchanged, the model has identified additional features and is in a state of overfitting. If the loss of both data sets is reduced to a certain level and cannot be reduced any further, a bottleneck in learning has occurred. A normal training situation would be for both data sets to decrease in loss and increase in accuracy, as shown in Figure 10. Therefore, during the training process, analysing the change in loss for each training round can be used to infer the current state of the model and make adjustments.

The other dataset used in this experiment is the test set. After training, the simulation of the test set can be treated as a practical application of the model and gives the most intuitive picture of the training effect of the model. This experiment is a multi-classification problem where the model needs to classify images into one of seven different broad categories. Here the test results are shown in Figure 11.

```
2019/2019 - 59s - loss: 0.2943 - acc: 0.9767 - 59s/epoch - 29ms/step
0.9766700267791748
```

Figure 11

The test set consisted of 105 images, and the results in Figure show that the model recognized these 105 images correctly at a rate of 97.667%. This shows that the model was trained very well.

Section 6.3 – Conventional Method VS Machine Learning

As can be seen from the results of Table 5, the accuracy of conventional methods will decline when using machine learning datasets for testing. Because the above conventional methods cannot count the green apple, the test results of the green apple image are all zero, and because the resolution of the image is very low, the algorithm also has an error in counting the red apple. Comparing the results of the two methods, we can see that the machine learning method in this paper can count green apples, which cannot be done by the conventional method in this paper, and the machine learning method can maintain high accuracy in the state of low image resolution. This is also one of its advantages.

Table 5: Accuracy for machine learning dataset using conventional method

Input image	No. of apples count manually	No. of apples count by the algorithm	Percentage of accuracy	Accuracy as per visual situation	Accuracy
1	0	0	100%	100%	76%
2		0	100%		
3		0	100%		
4		0	100%		
5		0	100%		
6	1	1	0%	80%	
7		0	100%		
8		1	100%		
9		1	100%		
10		1	100%		
11	2	2	100%	80%	
12		2	100%		
13		2	100%		
14		2	100%		
15		0	0%		
16	3	3	100%	80%	
17		3	100%		
18		0	0%		
19		3	100%		
20		3	100%		
21	4	4	100%	70%	
22		0	0%		
23		3	75%		
24		4	100%		
25		3	75%		
26	5	5	100	52%	
27		4	80%		
28		0	0%		
29		4	80%		
30		0	0%		
31	6	0	0%	70%	
32		6	100%		
33		4	66.66%		
34		6	100%		
35		5	83.33%		

Section 7: Conclusions and Future works

Section 7.1 – Conventional Method

Aiming at the detection and counting of fruits on apple trees, we propose an image segmentation scheme. Detection and counting work propose a machine vision algorithm to identify and count the number of apples in the image. In this paper, the concepts of CHT and color threshold are introduced, and the image is preprocessed by contourArea () and bitwise_and (). Finally, CHT is used to count apples. This method is suitable for the detection and counting of round apples under normal light conditions, and the accuracy is 96.07%. The main advantage of this method is the successful counting of occluded and overlapping apples with a small error of 5.53%. Overall, it is a reliable method with low errors counting the number of apples in different circumstances.

According to the result, the conventional method has 100% accuracy counting the non-occluded apples, which means the method of thresholding and CHT are effective. In another hand, when counting the occluded and clustered apples, the accuracy is decreasing with the increasing of difficulty which includes more objects occluded and overlapping the apple. Thus, it is obvious to know the weakness of this method that the CHT cannot detect or ignore some non-circle shapes which are caused by obstructions or different shooting angles. Moreover, due to the color thresholding, the method can falsely detect the yellow leaves or other background whose color is within the threshold as a part of the apples, which affects the CHT to recognize the real one. In summary, the conventional method still has its limitations when it comes to different counting tasks. Without deep learning and training, the method is not flexible enough to cope with multiple circumstances which is hard to generalize.

However, the algorithm can only detect and count red apples, but cannot deal with green apples which have the same color as most leaves due to the color thresholding, and if the shape of the apple is not round, the algorithm can cause more errors. The future work can be extended to detect and count green apples with another method to extract non-color features. And it is needed to improve the method to detect non-round shape apples in the whole environment.

Section 7.2 – Machine Learning

In this experiment, we use deep learning to identify and count the number of apples. A large training set of data can be used to enable the computer to recognize the features of an object and thus to identify it. This approach eliminates the need for manual processing of features and allows the extraction of as many features as possible by improving the loss function.

In this experiment, firstly, the results show that although our test results are 97% accurate, the analysis of the training and validation curves shows that the validation set curve fluctuates greatly. This indicates that there is a problem with the generalizability of the model, and the fluctuations in the validation set curve should be eliminated in the future by adjusting the parameters so that the validation set curve overlaps with the training set curve as much as possible. In addition, in terms of model adoption, the residual network itself is an improvement on the classical network. However, there are also neural network models with more layers such as ResNet50. There are also improvements to residual networks such as DenseNet. In future experiments, we can try out more network models to test their impact on the experimental results.

References

Behera, S.K., Mishra, N., Sethy, P.K. and Rath, A.K. (2018) *On-Tree Detection and Counting of Apple Using Color Thresholding and CHT*In: *2018 International Conference on Communication and Signal Processing (ICCSP)*. [online]. 224–228.

Haeni, N., Roy, P. and Isler, V. (2019) MinneApple: A Benchmark Dataset for Apple Detection and Segmentation. [online]. Available from: http://conservancy.umn.edu/handle/11299/206575.

Häni, N., Roy, P. and Isler, V. (2020) A Comparative Study of Fruit Detection and Counting Methods for Yield Mapping in Apple Orchards. *Journal of Field Robotics*. [online]. 37 (2), pp.263–282. Available from: http://arxiv.org/abs/1810.09499.

He, K., Zhang, X., Ren, S. and Sun, J. (2015) *Deep Residual Learning for Image Recognition* [online]. arXiv. Available from: http://arxiv.org/abs/1512.03385.

Patel, H., Jain, R.K. and Joshi, M. v (2012) Automatic segmentation and yield measurement of fruit using shape analysis. *International Journal of Computer Applications in Technology*. 45, pp.19–24.

Rahnemoonfar, M. and Sheppard, C. (2017a) Deep Count: Fruit Counting Based on Deep Simulated Learning. *Sensors*. [online]. 17 (4), p.905. Available from: https://www.mdpi.com/1424-8220/17/4/905.

Rahnemoonfar, M. and Sheppard, C. (2017b) Deep Count: Fruit Counting Based on Deep Simulated Learning. *Sensors*. [online]. 17 (4), p.905. Available from: https://www.mdpi.com/1424-8220/17/4/905.

Septiarini, A., Sunyoto, A., Hamdani, H., Kasim, A., Utaminingrum, F. and Hatta, H.R. (2021) Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features. *Scientia Horticulturae*. [online]. 286, p.110245.

Wang, Q., Nuske, S., Bergerman, M. and Singh, S. (2013) *Automated Crop Yield Estimation for Apple Orchards* Desai, J.P., Dudek, G., Khatib, O. and Kumar, V. (eds.) [online]. 88, Heidelberg, Springer International Publishing, 745–758. Available from: http://link.springer.com/10.1007/978-3-319-00065-7_50.