Comparison of different Convolutional Neural Network models on Fruit 360 Dataset

Kevin Liu [1, *]

[1] Department of Mathematics, University of California San Diego, San Diego, United States of America

* Corresponding Author Email: kkliu@ucsd.edu

Abstract. Numerous Convolutional Neural Networks emerged in the past decade, each varies in accuracy, speed, and architecture. From AlexNet to ResNet, CNN models have been developing rapidly, and the architecture of the models become more complicated. These models are known for their accuracy on ImageNet, so the topic of this research is to explore how CNN models can perform differently on the Fruit 360 dataset. A model constructed specifically in this research and three significant models developed in the past decade are applied to the Fruit 360 dataset for result comparison: VGG-16, ResNet-50, MobileNet, and SC-3. The models are selected to be different in architecture for comparison. The models are modified in similar ways and are trained on the same device under the same circumstances. The research found that ResNet-50 obtains the best prediction accuracy. MobileNet takes the least time to train and predict. The ResNet-50 and MobileNet models can be considered for real-world application, whereas VGG-16 and SC-3 are either inefficient or fails to generalize to complicated situations.

Keywords: Convolutional Neural Network, VGG-16, ResNet-50, MobileNet, Image Classification.

1. Introduction

Convolutional Neural Networks (CNNs) have been widely adopted in image classification tasks. The first CNN model LeNet-5 [1] was developed in 1998, it was used for handwritten digit recognition. Further advancements in improving image classification model accuracies and developing new CNN architectures were made when ImageNet [2] started a classification challenge called 'ImageNet large scale visual recognition challenge (ILSVRC)' [3] in 2010. The challenge measures the performance of a model in Top-1 accuracy and Top-5 accuracy. In 2010, the winner of the challenge had a Top-5 accuracy of 71.8%, but the method did not implement CNN. It was until 2012 that the first winner implemented CNN architecture emerged, and the model is known as AlexNet [4]. It improved the Top-5 accuracy to 84.6%, a great advancement at the time. After AlexNet, more CNN models have been developed to complete the challenge. Among VGGNet [5], GoogleLeNet [6], ResNet [7], and MobileNet [8], GoogleLeNet was the ILSVRC winner in 2014 with a Top-5 accuracy of 93.3%. ResNet was the winner in 2015 with a Top-5 accuracy of 96.4%. These models all made significant improvements in the accuracy of image classification. Some of the architectures are still being widely used in the present day, and notably affect models developed after it.

In this paper, 4 models will be applied to the Fruit 360 dataset [9, 10], and then compare the results obtained. The selected 4 models are VGG-16, ResNet-50, MobileNet, and a simple CNN model constructed by the author of this paper which will be called SC-3. Firstly, the pros to the VGG-16 model are that it is easy to construct and understand. The cons to this model are that it has a large number of training parameters, resulting in the model being large in size, and also the model has a long training time. Secondly, the pros to the ResNet-50 model are that it allows the network to go deeper in depth and achieve higher accuracy, and the training parameters are less than a normal CNN model with 50 layers. The con to this model is that it is hard to understand and implement. Thirdly, the pros to the MobileNet model are that it has fewer training parameters and is significantly smaller in size compared to other models, and it also trains faster than most models. The con to this model is that it has a lower accuracy compared to other models with the complicated architectures. At last, the

pros of the SC-3 model are that it is extremely easy to implement and fast to train. The con to this model is that it lacks the ability to extract complicated features due to its simple structure.

Section 2 is a description of the methodology adopted in this paper. Section 3 records, compare, and analyze the results. Section 4 will be a conclusion that summarizes the results obtained from this paper and discuss potential implementation and improvements of the results. Section 5 lists all the references in this paper.

2. Methodology

2.1. Data description

The dataset used in this paper is Fruit 360 dataset. The Fruit 360 dataset contains 131 classes of fruits and vegetables with a total number of 90483 images. There are 67692 images in the training dataset and 22688 images in the testing dataset, and each image contains one fruit or vegetable. All images in the dataset have a size of 100x100 pixels. Due to the different lighting conditions, the fruits and vegetables are extracted from the background to improve the accuracy of classification.

In order to have a working model with good accuracy, both the training and testing datasets must have an equal distribution of data. By comparing the two datasets, it can be observed the ratio in the distribution of the images across all classes is similar. This can be verified by calculating the ratio of some classes in the two datasets.

Consider the Grape Blue and Ginger Root classes. The Grape Blue class contains the most images in both the training and testing dataset, with a number count of 984 images in the training dataset and 328 images in the testing dataset. Through calculation, the ratio of the Grape Blue class in both the training and testing datasets is around 1.45%. Contrastingly, the Ginger Root class contains the least images in both the training and testing dataset, with a number count of 297 images in the training dataset and 99 images in the testing dataset. Through calculation, the ratio of the Ginger Root class in both the training and testing datasets is around 0.45%. This situation can be generalized to all other classes and proves that the training and testing datasets are balanced.

2.2. CNN models

All models are compiled using Adam [11] as the optimizer and categorical cross entropy as the loss function.

2.2.1 VGG-16

The VGG-16 model is a convolutional neural network constructed with 16 weight layers, consisting of convolution layers and max-pooling layers. The model has a size of 528 MB [12], and 138 million training parameters. A visualization of the VGG-16 model is shown in Figure 1.

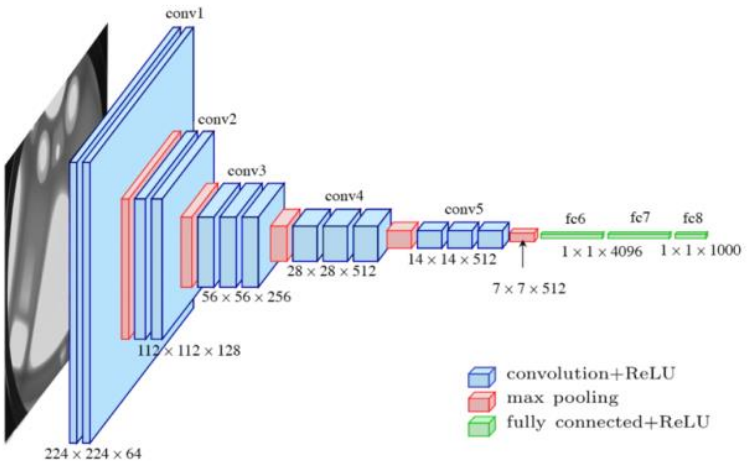

Figure 1. Visualization of the VGG-16 Model [13]

VGG-16 is a plain CNN network that does not implement any special techniques in its architecture. The network is simply convolution layers and max pooling layers stacked together. Transfer learning is applied to the VGG-16 model, and the weights are pre-trained on ImageNet. The input size is scaled down from 224x224 to 100x100, and the top 3 layers are removed from the model, which are fc6, fc7, and fc8 in Figure 3. The 3 layers are replaced by a dropout layer of rate 0.5 and a fully connected dense layer consisting of 131 units using the softmax activation function for predictions.

2.2.2 ResNet-50

The ResNet-50 model is a modified convolutional neural network 50 layers deep, consisting of convolution layers, max-pooling layers, and average pooling layers. The model has a size of 98 MB [12], and 25.6 million training parameters [12].

ResNet-50 employs residual blocks in Figure 2, where a shortcut connection that performs the identity mapping is introduced to the plain 50-layer convolutional neural network. The residual block finds the difference of the output from the input instead of the direct mapping of the input, which allows convolution neural networks to reach a greater depth while avoiding the vanishing gradient problem. The residual block of ResNet-50 contains 3 weight layers.

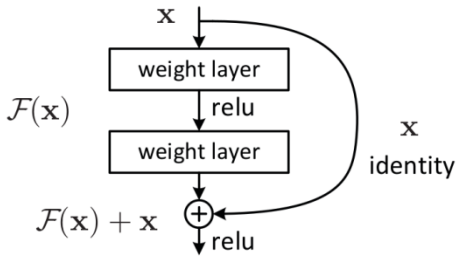

Figure 2. Shortcut Connection [7]

Similar to VGG-16, transfer learning is also applied to the ResNet-50 model, the weights are handled the same as in the VGG-16 model. The input size is also scaled down to 100x100 pixels, and the top layers are removed from the model. A dropout layer with of rate 0.5 and a dense layer with 131 units are added on top of the ResNet-50 model.

2.2.3 MobileNet

MobileNet model is a convolutional neural network 28 layers deep. The model has a size of 16 MB [12], and 4.2 million training parameters [8]. The training parameters of MobileNet are significantly less than other models. The original MobileNet contains 4.2 million parameters while being 28 layers in depth, compared to the original VGG-16 model with 138 million parameters while being only 16 layers in depth.

The depthwise separable convolutions used by MobileNet were first put forth in the work of SIfre [14]. Depthwise convolution layers and pointwise convolution layers make up depthwise separable convolutions. The depthwise convolution operates on the channels of the input. An image has 3 channels, which are red, green, and blue channels. Therefore 3 convolutions will be performed individually on each input channel. The pointwise convolution is an ordinary convolution with a kernel size of 1x1. The pointwise convolution combines the features obtained by the depthwise convolution.

The MobileNet model in this paper is constructed in a similar way as the VGG-16 model and ResNet-50 model. Transfer learning is applied, and input images are scaled down to 100x100 pixels. The top layers of the MobileNet model are removed. A dropout layer with of rate 0.5 and a dense layer of 131 units are added on top of the model.

2.2.4 Self-constructed CNN (SC-3)

A simple CNN model (SC-3) constructed in this paper is also applied to the Fruit 360 dataset. The size of SC-3 is 13 MB, and the model contains 3.3 million parameters. The architecture of the SC-3 model is simple, consisting of only 3 convolution layers.

Transfer learning is not employed in the SC-3 model, all weights are randomly initialized. The input size takes in images of size 100x100 pixels. The first 2 convolution layers extract 32 filters and have a 3x3 kernel size. The third convolution layer extracts 64 filters and has a 3x3 kernel size. The top layers of SC-3 are constructed using the same method as the 3 other models in this paper, consisting of a dropout layer of rate 0.5 and a dense layer with 131 units on top of the 3 convolution layers.

2.3. Training process

A validation split is performed on the original training set of Fruit 360. The original training set is split into a new training set and a validation set. The new training dataset is used for training in each epoch and the validation dataset is used to check the accuracy of the model after each epoch. The new training set takes 80% of the original training set for training, in a total of 54190 images. The validation set takes 20% of the original training set for validation, in a total of 13502 images. Data Augmentation is applied to prevent overfitting in the models.

The 4 models are trained on the same device under the same conditions for 25 epochs each on the new training set. Images are fed into the models in a batch size of 128 in each step until the entire training set is exhausted. The model records the loss and accuracy during the training and validation at the end of each epoch.

Two callback functions are utilized in the training process, EarlyStopping and ReduceLROnPlateau [12]. The EarlyStopping function is assigned to monitor the validation accuracy value. The EarlyStopping function stops the training process once the monitored validation accuracy values stop improving by a minimum difference of 0.5% for 4 epochs. The weights that give the best validation accuracy will be recorded during the training, and the weights will be restored to the model when the training terminates. The ReduceLROnPlateau function monitors the validation loss value. The ReduceLROnPlateau function will reduce the current learning rate by a factor of 0.1 when the validation loss ceases to decrease for 2 epochs. The starting learning rate is 1e-3 and the minimum learning rate that can be reduced to is 1e-5.

3. Results

3.1. Testing and prediction accuracy

The models are tested on the 22688 images in the testing set. The prediction will output an array of shape as (22688, 131), including predictions out of the 131 classes for all 22688 images. Only the top-1 accuracy is accounted for, and the average accuracy over the entire testing set is recorded as the final accuracy. Table 1 provides the detailed information about the accuracy and training time of each model.

Table 1. Accuracy and Training Time

Model	Accuracy (%)	Training Time (minutes)
VGG-16	96.90	39
ResNet-50	98.85	28
MobileNet	97.95	25
SC-3	97.81	34

Figure 3 is a visualization of the accuracy of each model.

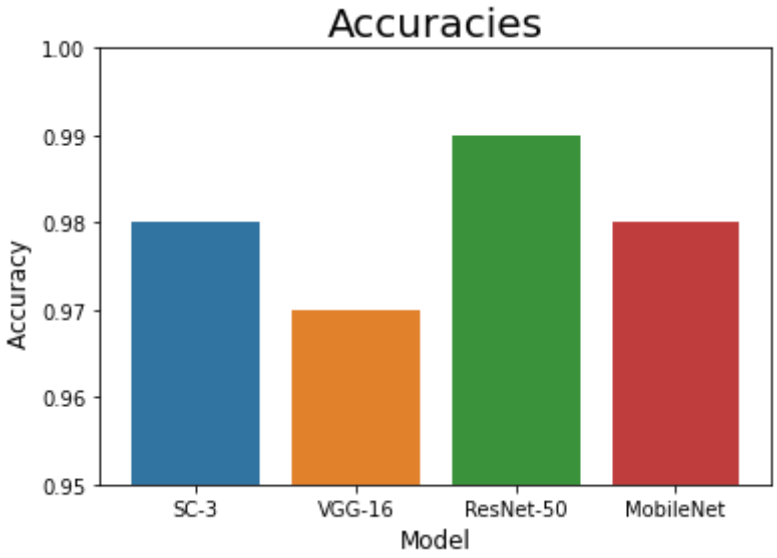

Figure 3. All Model Prediction Accuracies

3.2. Visualization

3.2.1 Accuracy and loss

Figure 4 shows the graphs of training accuracy, training loss, validation accuracy, and validation loss recorded during the training process.

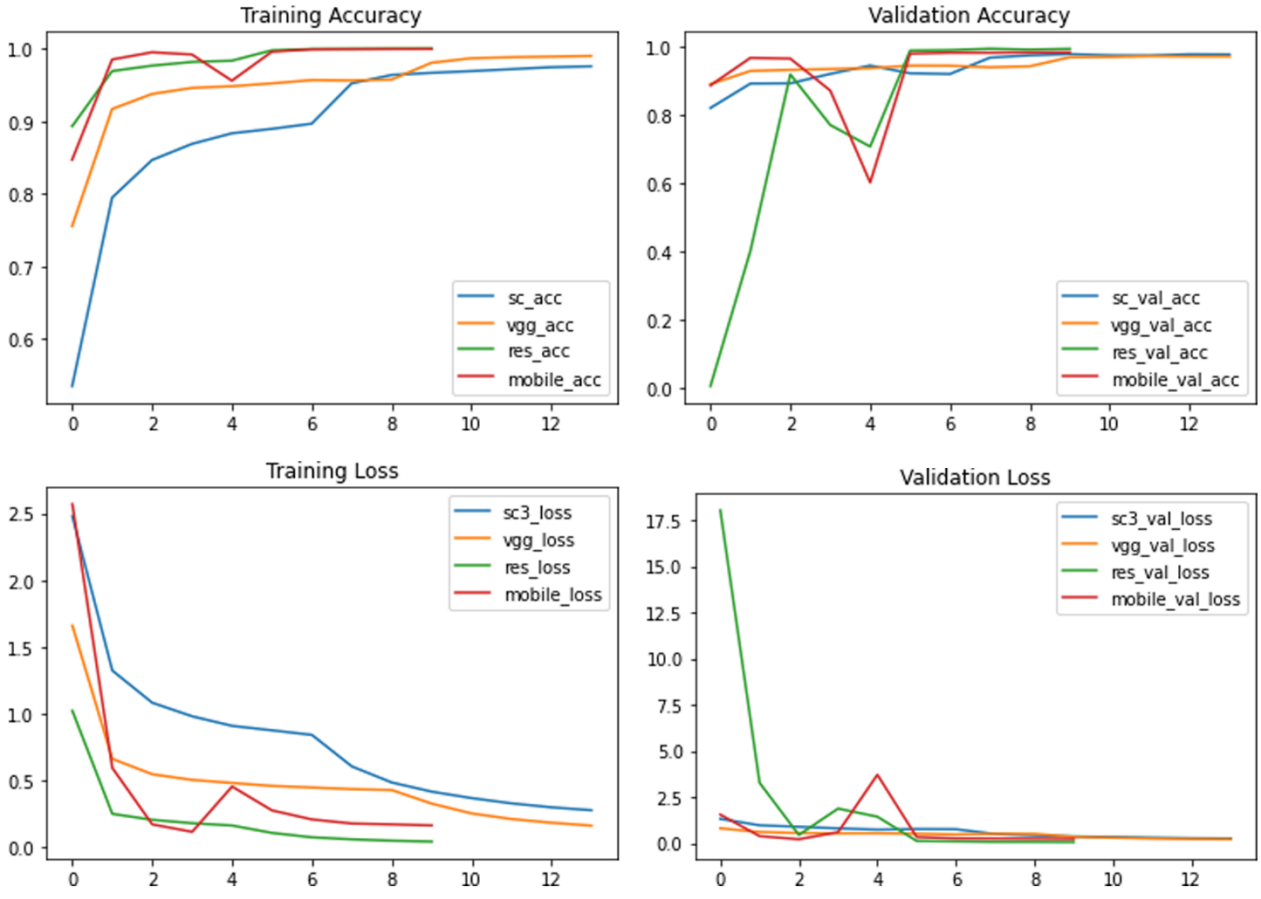

Figure 4. Accuracy and Loss in Training and Validation Sets

By examining Figure 4, it can be established that ResNet-50 and MobileNet take fewer epochs to train than VGG-16 and SC-3 while having a higher validation accuracy and a lower validation loss.

3.2.2 Confusion matrix

Displaying the entire confusion matrix for all 131 classes of fruits and vegetables in the dataset will be extremely hard for reading. Instead, 5 classes are selected for display to give a clear visualization of the model's prediction. The 5 classes with the lowest accuracy are chosen for analysis.

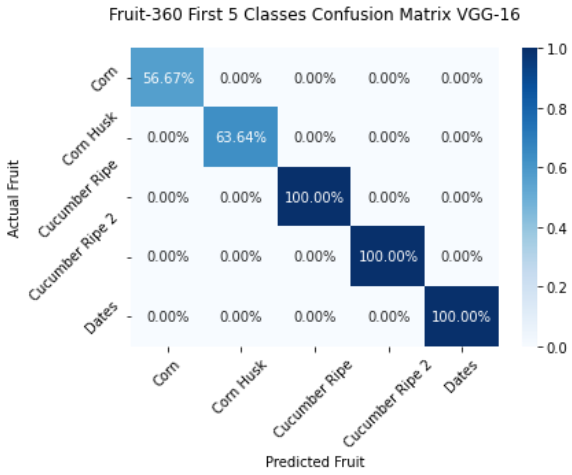

Figure 5. Confusion Matrix VGG-16

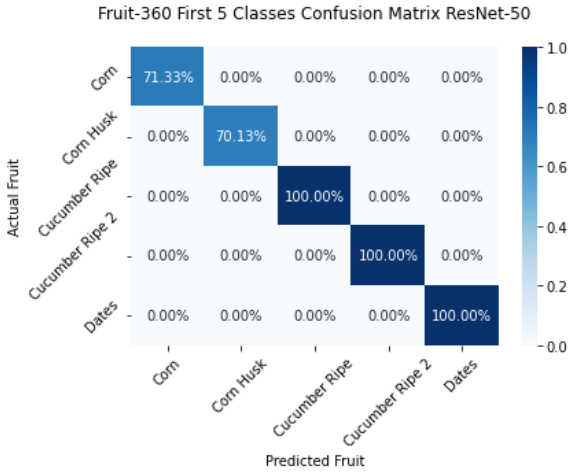

Figure 6. Confusion Matrix ResNet-50

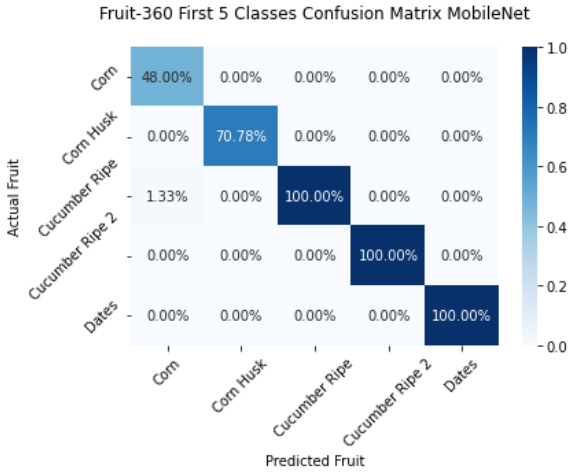

Figure 7. Confusion MobileNet

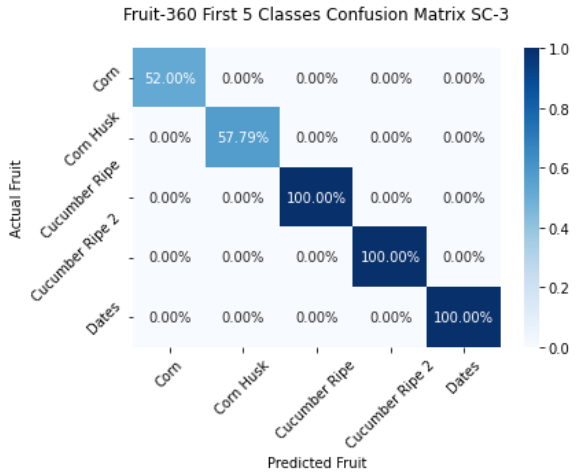

Figure 8. Confusion Matrix SC-3

Figure 5, Figure 6, Figure 7, and Figure 8 are the confusion matrices. It can be observed that all the models' prediction accuracy reaches 100% for each class, with exceptions for the Corn and Corn Husk classes. The two classes tend to have low prediction accuracy below 75% in all 4 models.

3.3. Result analysis

The VGG-16 model is originally trained with trainable ImageNet weights like other models. However, the results obtained are exceedingly poor in accuracy, achieving only 89.68%. By making the ImageNet weights untrainable in the VGG-16 model, leaving only the last convolution layer trainable, the model is able to train faster and obtain a higher prediction accuracy of 96.90%.

The ResNet-50 model gives the best prediction accuracy of 98.85% and takes the second shortest time to train. ResNet-50 has more FLOPs when compared to MobileNet and SC-3, therefore it trains slower than the two models. Under the setup of this paper, ResNet-50 and VGG-16 have around the same training time because not all weights are trained in the VGG-16 model, and the top layers of both models are modified from the original setup. Under ordinary setups, ResNet-50 should train faster than VGG-16. This is previously explored in the study of K. He et al [7].

By examining Figure 4, spikes in validation accuracy and loss can be found for ResNet-50 and MobileNet. This is caused by the use of mini-batches during training. The images are fed into the model in batch sizes of 128. The gradient calculated by the optimizer on some batches cannot be applied to the entire training set, causing there to be noises in the training process. These spikes can help the optimizer escape some local minima in the training process. Therefore, the spikes are expected and do not cause any deficiency in the model.

SC-3 takes the least time to train each epoch because its architecture is the simplest, containing only 3 convolutional layers and 1 dense layer. SC-3 reaches an average prediction accuracy of 97.81% over the entire Fruit 360 dataset. This is because the size of the Fruit 360 dataset is small and fruits and vegetables lack complicated features. If SC-3 is trained and tested on ImageNet, the accuracy will decrease significantly since it cannot extract enough features with 3 convolutional layers. This lack of feature extraction also exists when applied to the Fruit 360 dataset but does not significantly affect the prediction accuracy.

The Corn and Corn Husk classes are good examples. By observing the confusion matrix (Figure. 8) of SC-3, the model only achieves a prediction accuracy of 52.00% and 57.79%. The other models also have poor average accuracy for the Corn and Corn Husk classes but perform better than the SC-3 model. By the fact that SC-3 is an extremely simple CNN with only 3 convolution layers, it can be concluded that insufficient features are extracted. ResNet-50 and MobileNet contain significantly more layers compared to SC-3. With the more convolution layers, the other models have the capability to extract more features from the image, therefore resulting in more accurate predictions compared to SC-3.

By observing the distribution of the misclassified classes in all 4 models, it can be identified that Cantaloupe 2, Onion White, and Apple Golden 1 are the most misclassified classes, as shown in Table 2.

Table 2. Distribution of Predictions for Corn in all models (The most prominent 2 classes are displayed)

Model	Classes	Percentage (%)
VGG-16	Corn	56.67
	Cantaloupe 2	25.33
ResNet-50	Corn	71.33
	Onion White	19.33
MobileNet	Corn	48.00
	Onion White	47.33
SC-3	Corn	52.00
	Apple Golden 1	21.33

The misclassified classes appear similar in color compared to Corn, as shown in Figure 9.

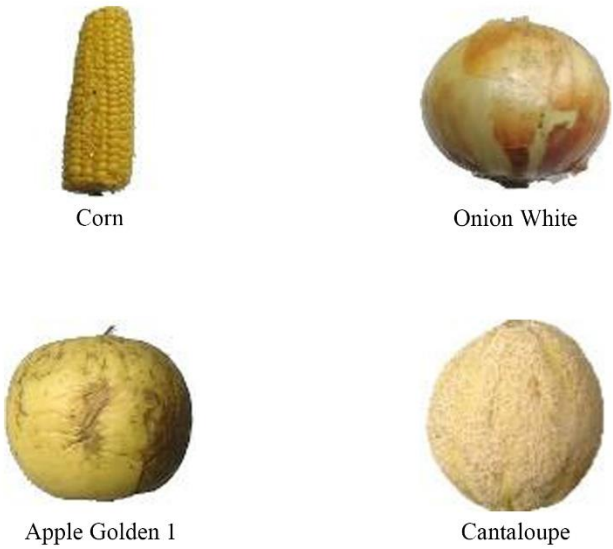

Figure 9. Visualization of Corn, Cantaloupe, Onion White, and Apple Golden 1

Therefore, it can be concluded that the SC-3 model out of the 4 models, lacks most of the ability to extract a detailed feature map in order to classify fruits and vegetables that appear similar physically.

The image of the Corn in Figure 9 does not appear similar in shape compared to the misclassified classes. By examining all images of the Corn class in the dataset, some images taken from the top of the Corn can be found, where the shape and color both appear in similar ways when compared to the misclassified classes, which may account for the reason for the misclassification.

Although the models predict poorly on certain classes, they have a reasonable accuracy over the entire dataset, meaning the models has high accuracy when predicting classes that do not appear similar physically. It can be proven by examining Figure 10.

Figure 10. Visualization of Cucumber Ripe, Cucumber Ripe 2, Dates

In Figure 10, classes Cucumber Ripe, Cucumber Ripe 2, and Dates are shown. The prediction accuracy for these 3 classes by all 4 models is 100%, as shown in Figure 5, Figure 6, Figure 7, and Figure 8. There is a remarkable difference in the appearance of each class, even though they are the same type of apple. Thus, the SC-3 model can differentiate between each class and reach an accuracy of 100%.

4. Conclusions

The 4 models applied to the Fruit 360 dataset all reached an acceptable accuracy above 95%. The ResNet-50 model obtains the highest prediction accuracy of 98.85%, then MobileNet with 97.95%, SC-3 with 97.81%, and VGG-16 with 96.90%. The average time needed to train each epoch varies for each model. SC-3 takes 2.43 minutes to train each epoch, taking the least time, and trains 14 epochs in total. MobileNet takes 2.50 minutes for each epoch and trains 10 epochs in total. ResNet-50 and VGG-16 both take around 2.80 minutes to train each epoch, with ResNet-50 training for 10 epochs and VGG-16 training for 14 epochs.

It can be concluded that ResNet-50 performs the best on the Fruit 360 dataset as it has the highest accuracy and takes the second least total time to train. The MobileNet model can also be considered for mobile fruit and vegetable recognition applications as it takes the least total time to train and still has relatively high accuracy. SC-3 and VGG-16 can be disregarded in real-world practice. SC-3 lacks the ability to extract the detailed features, and there is a high possibility to have a cluttered background in real-world practice, therefore it should not be considered. VGG-16 takes the longest to train and gives the poorest accuracy, ResNet-50 and MobileNet should both be considered before using VGG-16 as they give better accuracy, take less time to train, and are smaller in size.

The research conducted in this paper can demonstrate how different models perform on a relatively small dataset with simple features. The results of this paper can also be applied to real-world applications when fruit and vegetable image classification is needed. It will help with the selection of models. Large scale machines can select the ResNet-50 model for its high accuracy in classification, and mobile machines can select the MobileNet model for its advantage in small size and fast prediction.

The research conducted in this paper cannot be generalized to all fruit and vegetable classifications, as the fruits and vegetables are extracted out of the original image to avoid cluttering backgrounds and different lighting conditions. The situation where multiple fruits are in one image is also not tested in this paper. Future improvements can be training and testing different models with images of fruits and vegetables under the more complicated circumstances, like cluttering background, contrasting lighting scenarios, and detecting multiple fruits in one image, in order to generalize the predictions to real-life applications.

References

[1] Yann Lecun, Leon Bottou, Yoshua Bengio and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[3] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 2012, vol. 25. [Online].

[5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv, 2014. doi: 10.48550/ARXIV.1409.1556.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[8] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv, 2017. doi: 10.48550/ARXIV.1704.04861.

[9] Kaggle. 2017. Fruit 360. https://www.kaggle.com/datasets/moltean/fruits.

[10] Horea Muresan, Mihai Oltean. Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.

[11] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. arXiv, 2014. doi: 10.48550/ARXIV.1412.6980.

[12] F. Chollet and others, Keras. 2015. Keras. https://keras.io.

[13] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee, and Kincho H. Law. Automatic localization of casting defects with convolutional neural networks. 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 1726–1735. doi: 10.1109/BigData.2017.8258115.

[14] Laurent SIfre and Stéphane Mallat. Rigid-Motion Scattering for Texture Classification. arXiv, 2014. doi: 10.48550/ARXIV.1403.1687.