Experimental Evaluation of Convolutional Neural Network and Transfer Learning for Fruit Classification

Xuan-Y T. Dam - Hoang-Vu Vo - Le-Quang Nguyen
Faculty of Information Technology
University of Science
Vietnam National University, Ho Chi Minh City, Vietnam
19120160@student.hcmus.edu.vn, 19120727@student.hcmus.edu.vn, 19120121@student.hcmus.edu.vn

***Abstract*—Several methods of fruit classification have been proposed, and each is used for a different purpose. With the rapid development of computer vision and robot technology, Convolutional Neural Network(CNN) has been proposed and quickly become an extremely popular method. Many CNN models architectures have been built, from simple to complex models. Our goal is to build a fast-running CNN model, with short-term training but high efficiency for the well-preprocessed dataset. We focus on two methods: self-built CNN and CNN transfer learning from the VGG-16 model. We use the Fruits-360 dataset with 8132 images for training and 2715 images for testing from 15 different categories. We have intentionally chosen categories with similar shapes and colors, some from the same family, to emphasize the precise proportions of the two methods. We use various combinations of hidden layers, batch size, epochs, and dropout probability to achieve the best training results for both methods. The experiment shows that the CNN model gives better results with 100% training accuracy and 99.71% testing accuracy.**

***Index Terms*—Convolutional Neural Network, VGG-16, Fruit classifications, Fruits-360 dataset**

I. Introduction

For centuries, the fruit has played an important role in the human food chain. It provides many nutrients, fiber, vitamins, and so on. As a result, it helps people stay healthy both physically and mentally, reducing the risk of diseases such as cancer, heart disease, diabetes, etc. Nowadays, with the development of technology and cultivation techniques, fruit production is increasing day by day. The recognition and classification of fruit names in large quantities for preservation, export, or scientific research. Thanks to that technology, we can teach robots how to recognize and automatically sort fruit into storage or choose the right fruit to buy in the supermarket. Automation helps to increase productivity and limit manual errors because not everyone understands all kinds of fruits.

Many classification algorithms have been researched and developed before, with high efficiency and great significance in making the premise for this paper. First of all, PL. Chithra, et al., proposed a new method to classify fruits by using image processing techniques [1] with a nearly 100% accuracy rate. Another one is Fruit Classification for Retail Stores Using Deep Learning [2], which was introduced by Jose Luis Rojas-Aranda, et al..The result shows that the accuracy gets 95% for fruit with no plastic bag and 93% for fruit in a plastic bag. Deepika Bairwa, et al., have introduced a classification method of Fruits Based on Shape, Color, and Texture using Image Processing Techniques [3]. From the results, it is proved that color-texture-based classification gives the highest rate of accuracy: 97.2%, while color feature gives 90% and 89.60% with texture basis. Moreover, Yudong Zhang and Lenan Wu researched the classification of fruits using computer vision and a multiclass support vector machine [4]. They tested 3 different multi-class SVMs, including MWV-SVM, WTA-SVM, and DAG-SVM. As a result, they found that the MWV-SVM method with GRB kernel gave the best accuracy result of 88.2%. Jean A.T.Pennington and Rachel A. Fisher applied a mathematical clustering algorithm in order to classify fruits and vegetables [5].

We have researched and applied over 5 different classification methods to solve the fruit sorting problem in the most effective way. Regarding the dataset, we initially mixed images from Kaggle with images downloaded from google with the ratio of 60% and 40% respectively. However, the training accuracy is not high (85.7%) because there is no initial image processing step (detection and background separation of the image). Then when the initial image processing step is added, the classification is more accurate (92%), but there is still a lot of confusion since the images of each category are cropped from the video in different ways as well as different points of view. Therefore, the photos of the same category on google are really too diverse in shape, color, and context compared to the pictures taken from the fruit on Kaggle. Finally, we decided to just use the dataset from Kaggle and apply two methods: using a CNN [6] and using the Transfer Learning Model from VGG-16 to it. We apply the two training models with the study of changing some training parameters to obtain the highest possible results. In addition, we use several tools to visualize the results more, which are presented in section III of the paper.

The rest of this paper is described as follows. Section II

contains the explanations of every method and every necessary step to build the model. This section also analyses the dataset as well as its pros and cons. Section III shows the detailed results of these methods such as tables, figures, training/testing accuracies, confusion matrix...Section IV discusses the results, some strengths and weaknesses of our 2 models in detail. Finally, section V sums ups the content and also shows the applications of this study in the future or for another study.

II. THE DATASET AND METHODOLOGY

A. The dataset

The Fruits-360 dataset [7] contains 90483 images of 131 categories of fruits, where 75% samples are used for the training and the remaining 25% for testing. A white paper is placed behind the fruits as a background. Therefore, a flood fill type algorithm was proposed to extract the fruit from the background. After removing the background all the fruits are scaled down to 100×100 pixels of standard RGB fruit images. From the Fruit-360 dataset, we picked 10847 images from 15 different categories which 8132 images for training and 2715 images for testing. In the training dataset, we extract 20% of images for the validation randomly. Figure 1 below shows 15 distinct images, each one is represented for different categories and the figure 2 illustrates the numbers of images for training and testing in the dataset that we picked. The fruits picked in the dataset have similar shapes and colors, some fruits are in the same family so it can be a challenge for the model to accurately identify the category correctly.

Fig. 1. Display all fruit Categories

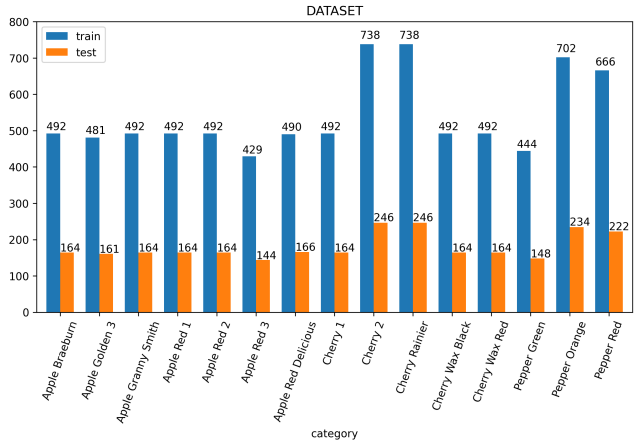

Fig. 2. Dataset Statistical Chart

B. The methodology

1) Approach 1: 4-layers Convolutional Neural Network Model: The developed methodology consists of four main steps. In the first step, we will build a CNN model for training & testing. In the second step, the input data must be pre-processed to have the same size and color space. Thirdly, the main step of this method is training and testing the model. Finally, we will get the result of the prediction.

a) Building a CNN model: This network contains four Convolutional layers which are used for the feature extraction from input data, each of them followed by a max-pooling layer. The input data is (100,100,3) that represents an image with the size of 100x100 pixels and color form is RGB (red, green, blue). Therefore, the input layer has 30000 neurons as input data.

The convolutional layers have 16 filters, 32 filters, 64 filters, and 128 filters sequently. Each of them has a kernel size of 2x2 pixels, the zero-padding is implemented which keep the dimension of the output image the same as the input image, and Rectified Linear Units (ReLU) function [8] is used as an activation function. When we use the ReLu function, if the output is positive, it is unchanged. Otherwise, the ReLu function will set it to zero. Because of this benefit, ReLu has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance. In the pooling layer, max-pooling is used with a pool size of 2x2 along with the stride length of 2 pixels. It means we will filter all 2x2 cells of the input image and take the maximum value of this cell. Then the filter jumps 2 pixels at a time as we slide around them. The equation (1) describes the ReLU function generally.

$$f(x) = max(0, x) \tag{1}$$

After calculating and evaluating, we find that layer often drops out with a probability of 0.3 is used to reduce overfitting as well as improving the performance of the network by making it more robust. After the Dropout step, a flatten layer is used to convert all the resultant 2-dimensional arrays into a single long continuous linear feature vector before entering into the fully connected layers. The Hidden layer has 150 neurons with a dropout probability of 0.4 and the ReLU function. Again a dropout with the probability of 0.4 is used between the output layer and the last hidden layer in order to prevent the model from overfitting. The output layer (the last layer) consists of 15 neurons where softmax classifier activation is used to predict the output of the model and represents 15 different categories classes of fruits. The equation (2) describes softmax function generally

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))} \tag{2}$$

In compiling the CNN model, the optimizer plays an important role and helps to decrease the model's error function. Adam optimizer can optimize weight decay value. Hence, for choosing the optimal number of weights, there are various

algorithms of Stochastic Gradient Descent but the most efficient one is Adam so that is the reason why we use Adam optimizer. Another vital parameter is the loss function, and here we choose 'categorical_crossentropy' as loss function because this function is suitable for multi-label classes. This loss function provides one-hot vectors representation to specify different classes. The final parameter is metrics which is a list of metrics to be evaluated by the model and here we choose the accuracy metrics.

TABLE I
THE 4-LAYERS CNN ARCHITECTURE

Layer type	Filter used size, strides, padding	Output shape	Activation	Params.
Input	-	100 x 100 x 3	-	0
Conv2D	2x2 / 1x1 / same	100 x 100 x 16	relu	208
MaxPooling2D	2x2 / 2x2 / valid	50 x 50 x 16	-	0
Conv2D	2x2 / 1x1 / same	50 x 50 x 32	relu	0
MaxPooling2D	2x2 / 2x2 / valid	25 x 25 x 32	-	2080
Conv2D	2x2 / 1x1 / same	25 x 25 x 64	relu	0
MaxPooling2D	2x2 / 2x2 / valid	12 x 12 x 64	-	8256
Conv2D	2x2 / 1x1 / same	12 x 12 x 128	relu	0
MaxPooling2D	2x2 / 2x2 / valid	6 x 6 x 128	-	32896
Dropout	-	6 x 6 x 128	-	0
Flatten	-	4608	-	0
Dense	-	150	relu	691350
Dropout	-	150	-	0
Dense	-	15	softmax	2265

b) Pre-processing data : Input images can have different size and color space. Pre-processing refers to eliminate the noise and correct the distorted or data. Each image must be resized to the same size is 100x100 and represented in the form of RGB pixels. Then we will convert them to NumPy array and manipulate the data which each element of the array belong to [0,1] in order to make the calculation and manipulation become more convenient and efficient.

c) Training the model : In this process, we set the value of batch size as 30 and epochs as 30. The training process can be described in the 4 steps below generally:

- Step 1: All filters are initialized with random weights.
- Step 2: Input image goes through the forward propagation step which consists of convolution, ReLU function, and max-pooling operation along with forwarding propagation in the fully connected layer. Due to the softmax function [9], every class will have an output probability respectively. In the first training, the outputs probabilities are random because the weights of filters are randomly initialized.
- Step 3: Adam optimizer calculates the total error in the output layer and updates the weights suitably for the next training process. Therefore, the later training often has better accuracy than the previous one.
- Step 4: Repeat step 2 and step 3 with all images in the training set.

d) Result and making the prediction: In this step, we will evaluate the training accuracy and testing accuracy after training our model. We will choose randomly 16 images from the testing data and predict them.

2) VGG-16 transfer learning architecture: The ImageNet [10] Large Scale Visual Recognition Challenge (ILSVRC) is an annual computer vision competition. Each year, teams compete on two tasks. The first is to detect objects within an image coming from 200 classes, called object localization. The second is to classify images, each labeled with one of 1000 categories, called image classification. VGG 16 was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University in 2014 in the paper "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION" [11]. This model won first and second place in the above categories in the 2014 ILSVRC challenge [12]. This model achieves 92.7% top-5 test accuracy on the ImageNet dataset which contains 14 million images belonging to 1000 classes.

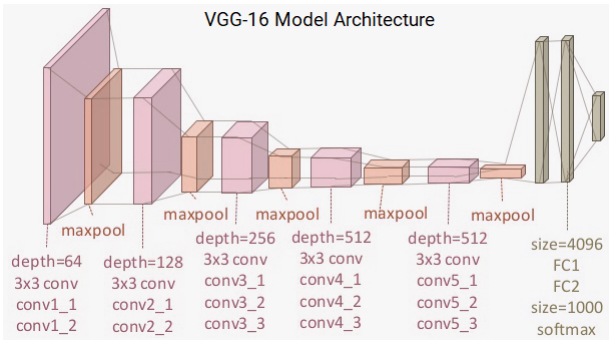

Fig. 3. VGG-16 model architecture

This approach can summarize in 6 process below:

a) Pre-processing input data: the same as the pre-processing step in approach 1

b) Building the model using transfer learning : In our model, the weight parameter value is "imagenet".The second parameter is include_top which is set to False, in which case the fully connected output layers of the model used to make predictions are not loaded. The input shape = (100,100,3) means the input image has the size of 100x100 pixels and color form is represented by RGB.

c) Freeze the top layer and add some custom dense layers: In this step, we customize a convolutional layer with 1024 filters with a kernel size of 3x3, zero-padding is implemented, and this layer is followed by a max-pooling layer with a pooling size of 2x2. ReLU function is used as the activation function.

A regularization layer dropout with a probability of 0.3 is used to randomly switch off 30% of the neurons in the layer during training to reduce overfitting as well as improving the performance of the network by making it more robust.

After the Dropout step, a flatten layer is used to convert all the resultant 2-dimensional arrays into a single long continuous linear feature vector before entering into the fully connected layers.

The Hidden layer has 150 neurons with a dropout probability of 0.4 and the ReLU function. Again a dropout with the probability of 0.4 is used between the output layer and the last hidden layer in order to prevent the model from overfitting. The output layer (the last layer) consists of 15 neurons where softmax classifier activation is used to predict the output of the

model and represents 15 different categories classes of fruits. This table II describes the model architecture using VGG-16.

TABLE II
THE MODEL ARCHITECTURE FOR TRANSFER LEARNING

Layer type	Filter used size, strides, padding	Output shape	Activation	Params.
VGG16	-	3 x 3 x 512	-	14714688
Conv2D	3x3 / 1x1 / same	3 x 3 x 1024	relu	4719616
MaxPooling2D	2x2 / 2x2 / valid	1 x 1 x 1024	-	0
Dropout	-	1 x 1 x 1024	-	0
Flatten	-	1024	-	0
Dense	-	150	relu	153750
Dropout	-	150	-	0
Dense	-	15	softmax	2265

d) Compile the model: Same as 4-layers Convolutional Neural Network in approach 1, we choose Adam optimizer, the loss function is 'categorical_crossentropy' and the value of the metrics is 'accurray'

e) Training model: training process with batch size is 30 and epochs is 30 and It is similar to the training process in approach 1.

f) Result and Predict: Evaluating and calculating the training & testing accuracy.

Selecting randomly 16 images from the test data set and make a prediction to get the result.

III. RESULT

This section will detail the experimental results obtained when training the model using the two mentioned methods.

First, this table III is a comparison of training and test rates:

TABLE III
TEST AND TRAINING ACCURACY

	Train accuracy	Test accuracy
CNN	1	0.9971
VGG16	1	0.9937

Therefore, we can see that the accuracy of the train and test of both models is very high. Moreover, the accuracy of the test using CNN is better than using VGG-16 (Transfer Learning) and is almost 100% (0.9971).

Second, the following Fig. 4 are graphs showing the dependence of Accuracy/Loss (including train and validation rates) on the increment of epochs of using the CNN method.

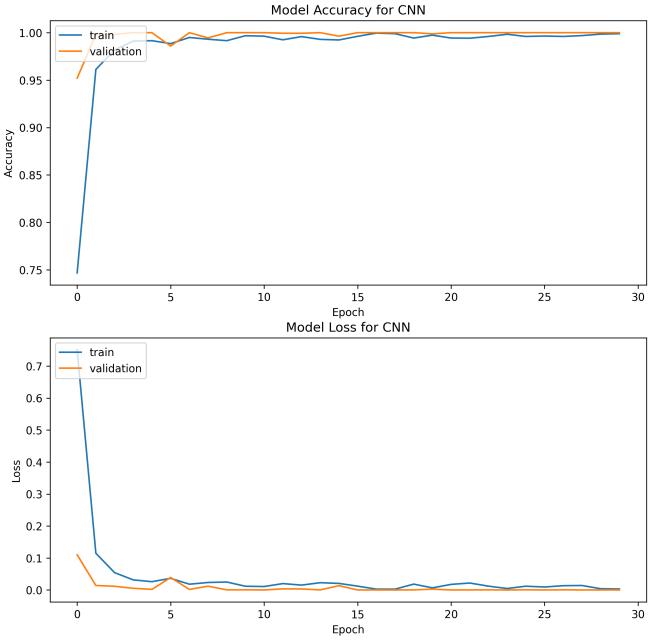

Fig. 4. Accuracy and Loss of CNN

Fig. 5 belongs to Transfer Learning:

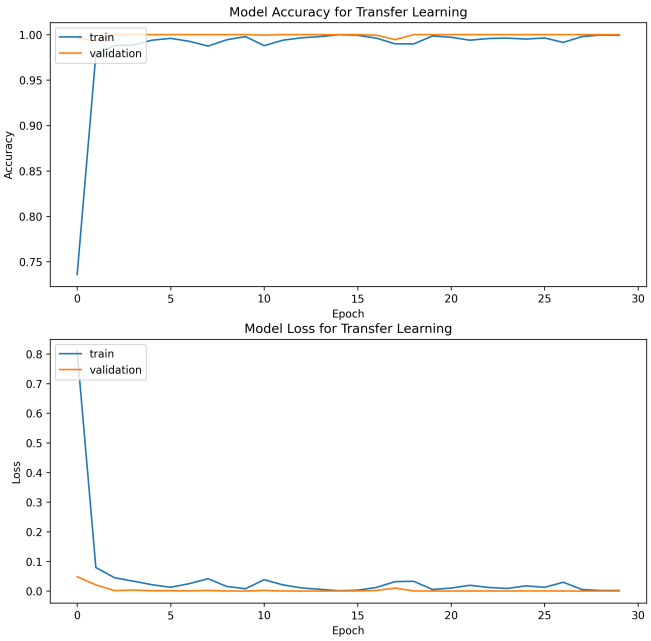

Fig. 5. Accuracy and Loss of Transfer Learning

To better illustrate the classification, we applied the model to predict and printed the prediction results with 16 random fruit images from the test set. When the image is classified correctly, the predicted label and its correct label (printed in parentheses) will be green, otherwise red.

This Fig. 6 is the result predicted by the first method (CNN) with all correct predictions:

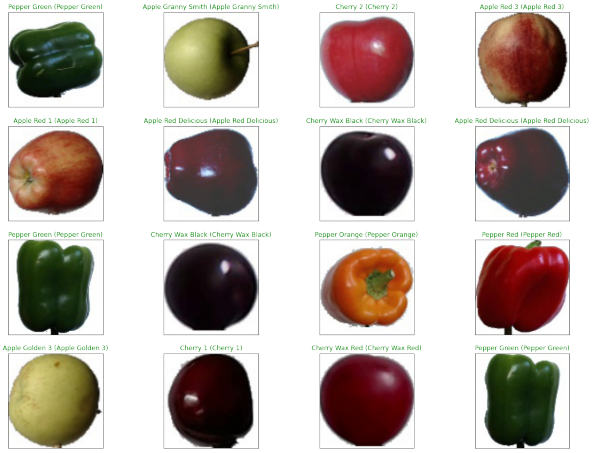

Fig. 6. Result predicted by CNN

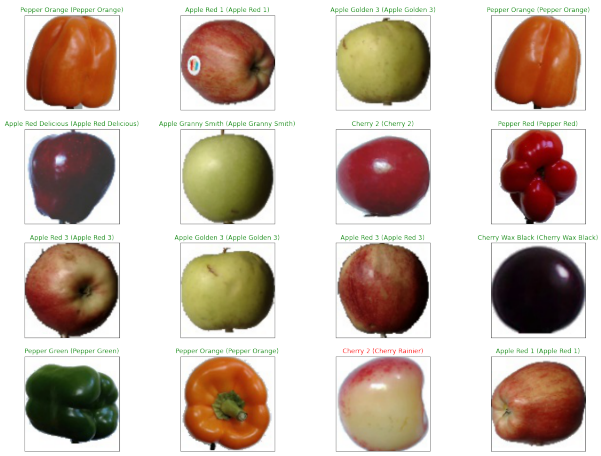

Fig. 7. Result predicted by Transfer Learning

With the second method (VGG-16), from Fig. 7, the result shows that there is an inaccurate prediction case: Cherry Rainer was wrong to predict Cherry 2. Finally, the following charts show the confusion matrix for all approaches:

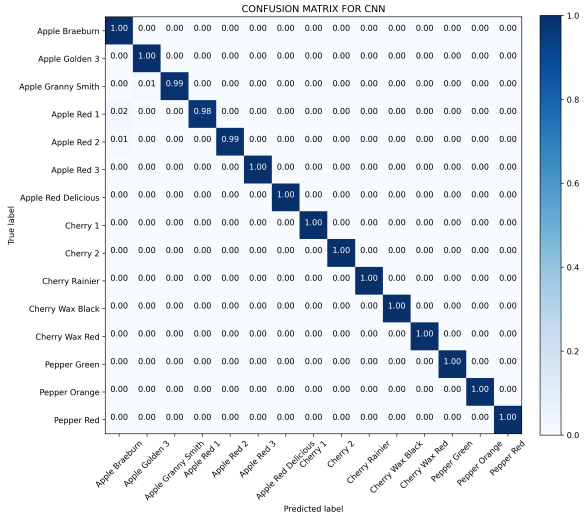

Fig. 8. Confusion Matrix For CNN

From Fig. 8, it is evident that the most predicted fruits with 100% accuracy except Apple Granny Smith, Apple Red 1, and Apple Red 2 (nearly 100%).

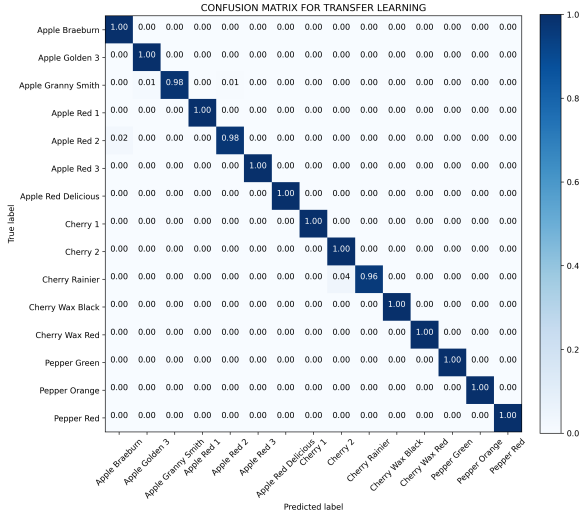

Fig. 9. Confusion Matrix For Transfer Learning

With Transfer Learning, through Fig. 9 we can see that 3 fruits have been mistakenly predicted, including Apple Granny Smith, Apple Red 2, Cherry Rainier with correct rates of 0.98, 0.98, and 0.96, respectively. Cherry Rainier is the most confused with a 0.04 rate as Cherry 2.

IV. Discussion

In this study, we use two methods (CNN and VGG16) to classify fruits with a dataset of 15 fruits. Compared with previous studies, we have achieved almost absolute accuracy (99.71% of CNN and 99.37% of VGG16) with a dataset that is difficult to distinguish even from humans, higher than previous studies with datasets including many common fruits. The results are also consistent with previous studies because they all yield high accuracy rates (over 90%).

In both models, through the Accuracy/Loss graphs over each epoch, we can see that in general the accuracy is increasing and the loss is decreasing to the highest acceptable extend. It can be concluded that the models gradually learn the features, get smarter, and predict more accurately. With high results in both methods, it means that we have successfully researched and applied these methods of fruit classification, achieving the set goal.

Overall, this study has the following great advantages:

- It has high applicability, minimizing errors in fruit classification in practice.
- Successfully extracting characteristics such as shape (from one fruit family to another), color (pepper has the same shape but different color), texture (Fruit-360 dataset is taken from multiple views of the fruit).
- Success in distinguishing fruits from the same family as they are quite similar, humans are still capable of making mistakes in this.
- The high prediction accuracy rate.
- The model is not too complicated, the training process is also fast.

However, it also has some concerns:

- The dataset is not diverse in the number of fruits (only tested on 15 fruits belonging to 3 different families), so there may be errors if the practical application with thousands of fruits.
- This model classifies well with properly preprocessed dataset. Otherwise, when the image has a complex background, is partially obscured or cropped, has foreign objects, the model may not be effective.
- There are still some errors in the image prediction process among the fruits of the same family.

Through the above analysis, it is easy to see that the CNN model gives better results, and the training process is also faster and saves resources compared to Transfer Learning. Therefore, with a well-processed dataset like Fruit-360, we should prefer the simpler CNN model.

V. Conclusion and future scope

This paper explores the classification of similar fruits based on two approaches: neural networks accumulating and transfer learning from VGG-16. With well-preprocessed datasets like the Fruit-360 that we used, the CNN method offers greater efficiency both in terms of training time and accuracy. However, if extended with more complex data (with background, leaves..), we believe that the transfer learning method will bring better performance.

In the future, we may expand our research to sliced, dried, and canned fruit. Furthermore, it is hoped that new characteristics can be drawn to distinguish ripe fruit from raw fruit, and delicious fruit from unpalatable fruit. This will help improve the pattern of fruit cultivation and processing.

References

[1] P. Chithra and M. Henila, "Fruits classification using image processing techniques," *Int J Comput Sci Eng*, vol. 7, no. 5, pp. 131–135, 2019.

[2] J. L. Rojas-Aranda, J. I. Nunez-Varela, J. C. Cuevas-Tello, and G. Rangel-Ramirez, "Fruit classification for retail stores using deep learning," in *Mexican Conference on Pattern Recognition*. Springer, 2020, pp. 3–13.

[3] D. Bairwa and G. Sharma, "Classification of fruits based on shape, color and texture using image processing techniques," *Int. J. Eng. Res.*, vol. 6, pp. 110–114, 2017.

[4] Y. Zhang and L. Wu, "Classification of fruits using computer vision and a multiclass support vector machine," *sensors*, vol. 12, no. 9, pp. 12 489–12 505, 2012.

[5] J. A. Pennington and R. A. Fisher, "Classification of fruits and vegetables," *Journal of Food Composition and Analysis*, vol. 22, pp. S23–S31, 2009.

[6] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*. Ieee, 2017, pp. 1–6.

[7] A. Kausar, M. Sharif, J. Park, and D. R. Shin, "Pure-cnn: A framework for fruit images classification," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018, pp. 404–408.

[8] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[9] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, vol. 181. Citeseer, 1997, p. 185.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

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

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.