Convolutional Neural Network for Fruit Image Classification

Ng Yao Rong
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp058809@mail.apu.edu.my

Cheong Yew Kien
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp058031@mail.apu.edu.my

Mohammed Omer
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp064706@mail.apu.edu.my

How Yan Han
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp064365@mail.apu.edu.my

Zailan Arabee Abdul Salam
School of computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
zailan@apu.edu.my

***Abstract* — Fruit image classification is one of the many things we can achieve when dealing with an artificial neural network, as it is very handy in helping many groups in society. This technology can improve the productivity rate, save a lot of expenses, and improve the competitivity of the worldwide fruit producer's market. We use various technologies such as deep convolutional neural networks (DCNN) for deep learning which is the pinnacle of technology (Sharma, 2021). The problem with DCNNs is that they require the necessities of high calculation and capacity assets, thus, deny the utilizations of DCNNs on asset restricted conditions like programmed reaping robots. Consequently, we want to pick a lightweight neural network to accomplish the equilibrium of asset limits and recognition precision. This paper will dive into the details of the processes that were done to showcase how the system is able to recognize fruits with the results.**

Keywords — Convolution Neural Network, Fruit Classification, Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN)

I. INTRODUCTION

With the development of cameras, many people now use them to take pictures for their memories, and some post them on social media to share with others. In the comments section, we can see that people are always asking for or searching for places or names in the photos themselves, and this is where image classification comes in handy.

With the advent of Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), machines have made great breakthroughs in the field of object detection and scene classification, especially CNN. Image classification is a supervised learning problem as well as the extraction of photographic features. There are many such extraction techniques, such as Scale-invariant feature transform (SIFT) (Lowe, 2004), histogram of oriented gradients (HOG) (Dalal & Triggs, n.d.), Local binary patterns (LBP) (Cheung & Deng, 2014), Content-Based Image Retrieval (CBIR) (Khan et al., 2012), etc. When the features of the image are extracted, the classification will be done by using classifiers (Sharma et al., 2018). There are two types of image classification which are single -label classification, and multi label classification. The major difference between them is that single-label classification has only one single label is present for each image, and the output has only one value or prediction. Multi-label classification is the opposite and is widely used in the medical imaging domain, e.g., to detect multiple diseases by X-rays (Bandyopadhyay, 2022).

CNNs are constructed to imitate the human or animal brain. Our brain contains 86billions of neurons (Herculano-Houzel, 2009), and CNNs are built to simulate the neurons in the brain. CNNs have a wide range of uses, for example in computer vision (Fang et al., 2020), speech processing (Palaz et al., 2019), face recognition (Li et al., 2020) and so on. Its main advantage is that it can identifies relevant features on its own, without human supervision (Gu et al., 2018). CNN architecture consists of 3 layers and 2 functions, which are convolutional layer, pooling layer, activation function, fully connected layer, and loss functions (Alzubaidi et al., 2021). There are many fruits in the world and even the same fruit has different characteristics, for example apple can be divided into red apple, golden apple and so on. In this paper, we focus on fruit classification, so that the machine can recognize which fruit the image is. The dataset we use is Fruits 360 from Kaggle, it is a dataset with 90380 images of 131 fruits and vegetables. The methodology is based on the MobileNetV2 architecture. We will use Google Colab to conduct this experiment. The rest of this paper is organized as follows: Section II Background, Section III Algorithms and Approaches, Section IV Methodology, Section V Results and Discussion. (Abdo et al., 2022) proposed obstacle avoidance robot using convolutional Neural Network to predict the object detected and to find a path of avoidance the detected object.

A. *Literature Review*

In the work of Liu et al. (Liu et al., 2019), the image is processed using frequency domain enhancement and the color model for basic image processing. they also used homomorphic filtering to improve the appearance of the Actinidia arguta orchard, highlight its distinctive features, and lessen the impact of background noise on the ability to identify the Actinidia arguta trunk. besides, they improved fruit location recognition accuracy by using a binocular stereo vision technology. In the end, they succeeded in separating ripe and immature kiwifruit with an accuracy of 99.16%.

In order to perform fruit classification, Feng et al. (Feng et al., 2019) used a more complex algorithm. This algorithm is based on the pseudo-color and texture data from MSX images for locking potential fruit regions and detect fruit. This algorithm can remove the non-target regions and extract the texture characteristics to improve the target regions' support judgement. The result is the complete fruit regions had recognition precision and sensitivity above 92%, and no less than 72% for incomplete fruit regions.

Geng et al., (2021) used the combination of hyperspectral and Temporal Convolutional Network (TCN) to classify apples. the Hyperspectral image acquisition system uses spectral data collected from different varieties of apples. In order to classify apple varieties, they used a double-branch structure in their recognition model, one with a TCN network and the other with two long and short-term memory networks (LSTM) modules. The results obtained from this module were the classification accuracy of seven kinds of apples reached 99.74%.

II. BACKGROUND

A. *Image Database*

The dataset we chose is from Kaggle.com which is fruits 360. Fruit 360 is a dataset of fruits and vegetables with a total of 90380 images of 131 different fruits and vegetables. The pictures were acquired by recording the natural products while they are turned by an engine and afterward separating outlines. Organic products were established in the shaft of a low-speed engine (3 rpm) and a short film of 20 seconds was recorded. Behind the organic products we put a white piece of paper as foundation. After downloading the dataset, we uploaded all the pictures into google drive as we used Google collab to compile and run the code and view the images. Uploading the dataset and using collab was the idea, due to Google collab and Google drive being corelated, as Google collab can access cloud services such as Google drive for datasets. The dataset is separated into 2 subdirectories which are training images and test images.

B. *Image Classification*

Convolutional Neural Network goes under the subdomain of AI which is Profound Learning. Calculations under Deep Learning process the data the same way the human cerebrum does, yet clearly for a tiny scope, since our mind is excessively complicated (our mind has around 86 billion neurons) (Xiang et al., 2019).

Picture arrangement includes the extraction of highlights from the picture to notice a few examples in the dataset. Involving an ANN with the end goal of picture arrangement would turn out to be expensive as far as calculation since the teachable boundaries become incredibly huge. For instance, assuming we have a 50 X 50 picture of a cat, and we need to prepare our customary ANN on that picture to group it into a canine or a feline the teachable boundaries become - (50*50) * 100 picture pixels increased by stowed away layer + 100 inclination + 2 * 100 result neurons + 2 predisposition = 2,50,302. We use channels while utilizing CNNs (Shahi, 2022). Channels exist of various kinds as per their motivation.

Thus, after importing the dataset with the help of TensorFlow, we allow our system to recognize and classify fruits and vegetables, after training and testing the images.

III. ALGORITHMS AND APPROACHES

A. *Artificial Neural Network*

Artificial Neural Network (ANN) algorithms were implemented to assist in fruit image classification with pre-trained image datasets that were downloaded from the Kaggle online website. The source codes used in this research belonged to MeAmarP from GitHub and were written in Java. The ANN method consisted of manmade neurons known as “nodes” that were inspired by the structure of the human brain which led researchers to develop the algorithm model. The algorithm model has three layers as shown in Fig 1, which are the input layer for accepting image data input, the "hidden layers" that consisted of neurons to process the image data to generate outputs via an activation function, and the output layer for displaying the final data output with a transfer function (Gill, 2022). The neurons are comprised of weights and biases that are interconnected, where the parameters can be adjusted to provide a higher level of accuracy. After the image data output results are displayed, backpropagation is initiated if the desired output has a low accuracy so that the model can learn from previous predictions to produce a more accurate result (Marr, n.d.).

The ANN algorithm used in this study was trained on different fruit images from fruit datasets and classified based on identical features for each of the fruits, which were then distributed according to their fruit classes. The training results were produced in the form of graphs to display the level of accuracy and loss for both training and validation of the fruit images after going through 20 epoch levels. Fig 1. shows the architecture of Artificial Neural Network.

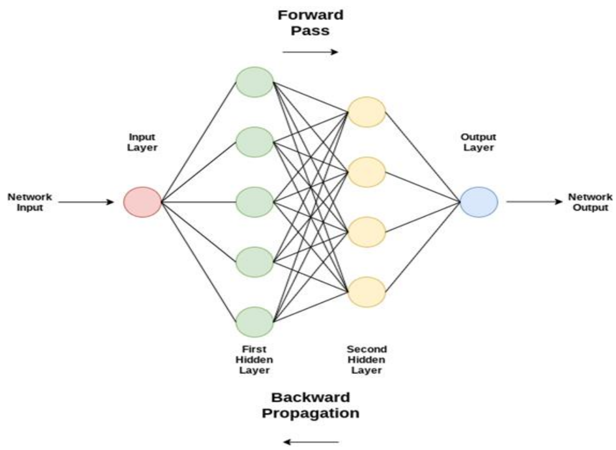

Fig. 1. Architecture of Artificial Neural Network.

B. *Convolutional Neural Network*

Convolutional Neural Network (CNN) is a type of Artificial Neural Network that specializes in data processing of graticule two-dimensional images. A digital image is comprised of binary numbers to produce complete visual data and consists of a series of pixels arranged in a grid-like pattern that denotes which pixels should have a higher illumination and what colors are assigned to each pixel (Mishra, 2020). The architecture of the CNN model trained input images from a source such as an image dataset, assigned weights, and biases across various aspects of the images to differ from one another, and produced output results known as feature maps that are displayed in arrays (Saha, 2018). The CNN model has three layers, which are the convolutional layer, pooling layer, and fully connected layer as shown in Fig 2.

In this research, the CNN algorithm model is used to perform an analysis of the fruit image datasets that were downloaded from the Kaggle online website for object recognition, segmentation, classification, and processing of images. The image data from the datasets are fed in large portions to the CNN model to recognize the various patterns for each image and noise was removed while the model enhances and smoothens the images before resizing them. The image segmentation process starts by converting the RGB color of the images to grayscale to partition them into several parts, after which the image representations are determined and changed for easier analysis. Image segmentation can be categorized into two parts which are similarity (represented by regions) and discontinuity (represented by boundaries). Following the process of image segmentation, the fruit images are processed to determine their color, shapes, sizes, and textures, which is known as feature extraction. The feature extraction was performed to remove redundant information while retaining those that are relevant for image recognition, making image detection to be much easier. The classification process is then finally carried out to classify the fruit images based on their similar extracted features and placed according to their fruit classes (Sari et al., 2021).

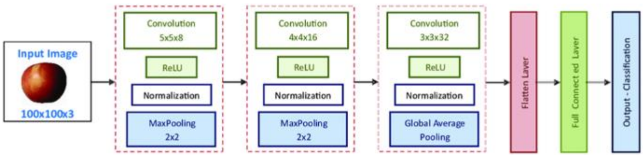

Fig. 2. Architecture of Convolutional Neural Network.

C. *MobileNetV2*

Mobile NetV2 is a part of the Convolutional Neural Network model that was developed by Google, which enabled image classification to take place. The Mobile NetV2 model is an improved version of the Mobile NetV1, in which the number of parameters in the deep neural network algorithm was significantly reduced, enabling it to perform tasks in a fast and efficient manner. The enhancement of the model resulted in it being lightweight which enabled it to be used for systems that are embedded and more convenient for mobile devices to consume less computational power to operate the model. The Mobile NetV2 was used as a pre-trained model for training large image datasets, saving development time by eliminating the need to train the deep neural network from the start (Nganga, 2022).

The fruit image datasets used in this study were downloaded from the Kaggle website and implemented in Google Colab, where TensorFlow is utilized as a base model from the Mobile NetV2 model. According to the Mobile NetV2 input requirements, the fruit image size was adjusted to 224x224 for better image representation, so that the image output results are not pixelated or blurry. The pre-trained Mobile NetV2 model was then used to create the base model for the fruit image classification, with the convolutional layers frozen to enable the base model to be utilized as a feature extractor. Following that, the layers in the model are routed through a sequential model, enabling the neural network to be built layer upon layer by hidden layers stacked on top of each other. The sequential method was implemented to modify the deep neural network so that it could recognize and perform tasks with greater accuracy. The parameters that were included in the layers of the model are as follows, the flatten layer, the dropout layer that manages model overfitting, and the dense layer which is the output layer and has Softmax as the activation function, which assists with multi-label classification. Following the sequential process, the image classification model will be compiled and trained to evaluate the performance of the training dataset, with the accuracy and loss results plotted in graph representations. If the output results were not satisfactory, the base model will need to be unfrozen and the layers must be frozen in advance of a certain layer to tweak the parameters to provide a better accuracy result (Sahoo, 2021). After the parameters have been fine-tuned, the process of freezing the base model, going through the sequential process, compiling, and training the model will be repeated until the desired output is achieved.

IV. METHODOLOGY

A. *Environment of Experiments*

In this research, a dataset which contains images of different fruit types has been used to train as a model for future predictions. The programming language that has implemented in this research is Python. The environment that we are using is Google Colab as libraries and requirements are provided from this platform. Besides, Google Colab is the only environment that we are using so that deviations such as different specifications in software and hardware can be avoided. The hardware specifications provided by Google Colab are listed in TABLE I.

TABLE I. HARDWARE SPECIFICATIONS FOR EXPERIMENTS

Hardware	Specification
GPU Model	Nvidia K80
GPU Memory	12GB
CPU Model	Intel(R) Xeon(R)
Number of CPU Cores	2
Available RAM	12GB
Disk Space	25GB

B. *Experiments*

We are going to apply MobileNetV2 pre-trained model as the feature extractor. The first step that we had in our experiment is that we used TensorFlow to create a base model with MobileNetV2 model that has been pre-trained on the ImageNet by Google. We have chosen one of the most common practices for the feature extraction, which is using the output of the layer before the flattened layer, also known as the “bottleneck layer” as it will retain more generality. Then we will add a new layer below the flattened layer with the default activation "ReLU", and dense with the value of 512. And finally, we will have the very last layer with "Softmax" activation and dense with the value just as the number of different fruit classes that we are going to train. The experiment will be conducted with Adam optimizer, batch size of 16, a learning rate with le-5, and 20 epochs. This will help us to determine the differences of the output so that we can implement the better option to get a better result.

$$f(x) = sigmoid(x) = \frac{1}{1+e^{-x}} \quad (1)$$

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; for \; j = 1, \dots, K \quad (2)$$

C. Applying Different Parameters

In order to find out the best approach, we have conducted numbers of experiments by implementing different parameters. The parameters that we are going to modify will be the learning rate, number of epochs, and the activation for the last layer.

Learning rate is a hyperparameter in neural network where it will manage the numbers of weights to be updated each time during the training, it is usually within the range of 0.0 to 1.0. In our experiment, the default learning rate that we had used is le-5. However, we do not know whether it will give us an optimal output. Thus, we trained the model with different learning rates to find out which of them will produce the optimal result. The first learning rates that we are using are le-3. It is a larger learning rate compared to the default value. After getting the result from le-3, we and modify the learning rate to a relatively larger at le-10 to see what will happened. And finally, we tried to find out the outputs with the learning rate at le-7 and le-6.

Epoch indicates the number of training iterations that passes through the entire dataset. We are assuming more epochs give us higher accuracy. Thus, we will only conduct one experiment with more epochs. The number of 20 epochs were defined for the first experiment. We have modified it to 30 epochs to see whether it will give us a better result.

Activation is a function in artificial neural network that helps the network to learn the intricate patterns in the data. There are two activation functions included in the experiment, ReLU (Rectified Linear Unit) and Softmax. The ReLU activation was used in the hidden layer, this provides a better performance, and it will help in avoiding the problem of vanishing gradient. On the other hand, Softmax where we have used it in the last layer will help us on handling different classes. Nevertheless, we will change the activation function in the last layer from Softmax to Sigmoid, both of the activation functions produce different results with their formula.

V. RESULTS AND DISCUSSION

A. Results

After running multiple times of the experiment, we have all the results from different parameters. The final loss, final accuracy, validation loss, validation accuracy, and time taken from different learning rate are stated in TABLE II. The number of epochs and activation function applied in these experiments are the default value which are 20 epochs, and Softmax activation function.

The next experiments were carried out with different number of epochs to show the difference between different epochs. We have changed back the learning rate to le-5 so that we are able to determine the effects brought by epochs. The results with default number of 20 epochs and the modified 30 epochs are record in TABLE III.

The activation functions were being investigated in the last experiment. Although we have mentioned that ReLU is not that suitable to be applied at the last layer, we are still going to examine its output. The default activation function was Softmax, and we have run the experiment by changing it to ReLU, and Sigmoid, the epochs are remained as 20. TABLE IV holds the results produced by the different activation function.

TABLE II. RESULTS OF DIFFERENT LEARNING RATE

Learning Rate	Training loss	Training Accuracy	Validation Loss	Validation Accuracy	Time Taken (h)
le-5 (default)	0.2604	0.9307	0.1488	0.9765	2.13
le-3	0.2186	0.9438	0.1104	0.8922	3.35
le-10	3.9653	0.0318	3.6786	0.0218	2.43
le-7	3.5916	0.0624	3.1676	0.0982	1.56
le-6	1.8616	0.4710	0.4418	0.7097	1.5

TABLE III. RESULTS OF DIFFERENT EPOCHS

Epochs	Training loss	Training Accuracy	Validation Loss	Validation Accuracy	Time Taken (h)
20 (default)	0.2604	0.9307	0.1488	0.9765	2.13
30	0.1385	0.9655	0.0654	0.9912	2.32

TABLE IV. RESULTS OF DIFFERENT ACTIVATION FUNCTION

Activation Function	Training loss	Training Accuracy	Validation Loss	Validation Accuracy	Time Taken (h)
Softmax (default)	0.2604	0.9307	0.1488	0.9765	2.13
ReLU	1.5622	0.6061	0.7987	0.8404	4.63
Sigmoid	0.2321	0.9402	0.1135	0.9820	3.23

Note that all the hyperparameters are at default values other than the hyperparameters in the caption. Fig 3-12 shows the results produced by the Convolutional Neural network.

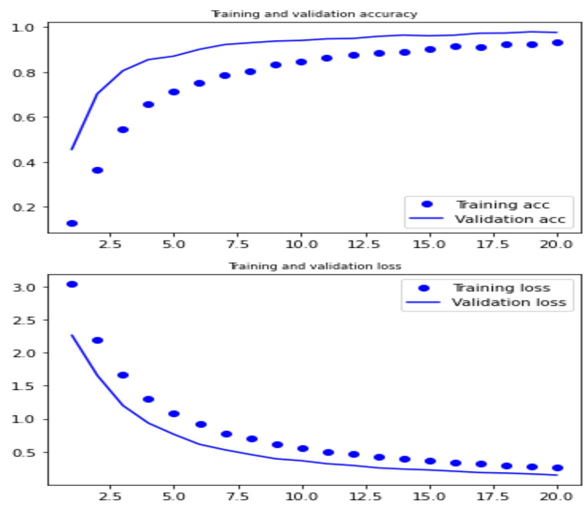

Fig. 3. Plot result of le-5.

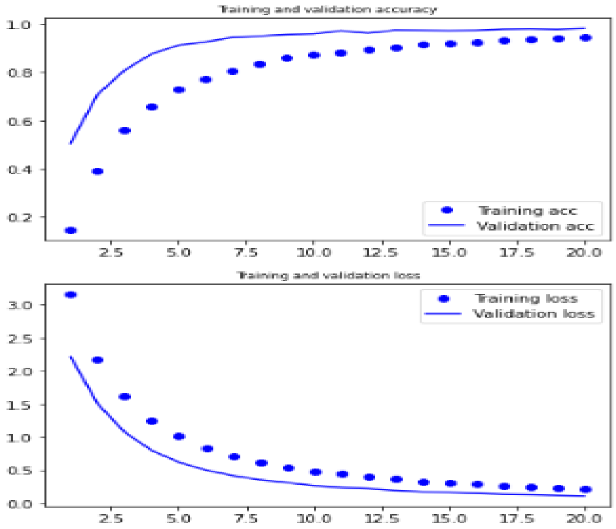

Fig. 4. Plot result of le-3.

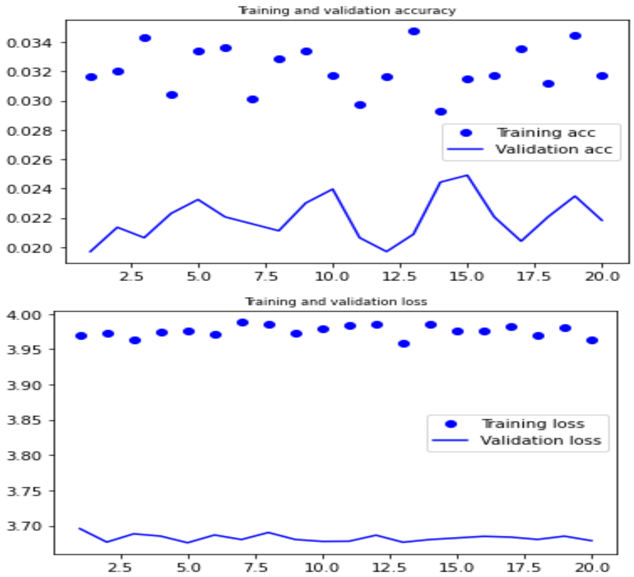

Fig. 5. Plot result of le-10.

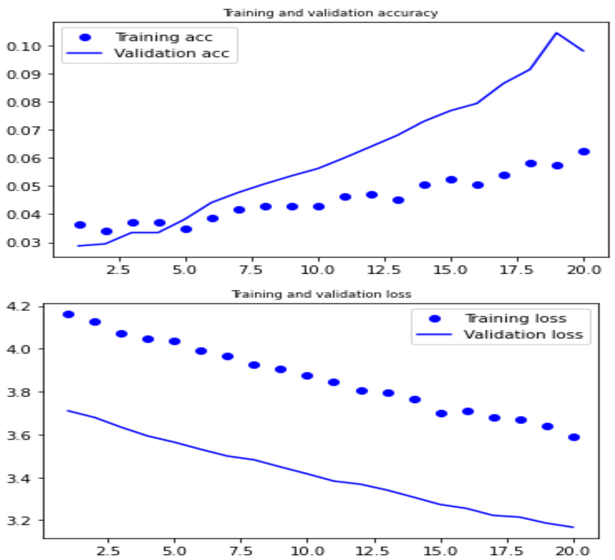

Fig. 6. Plot result of le-7.

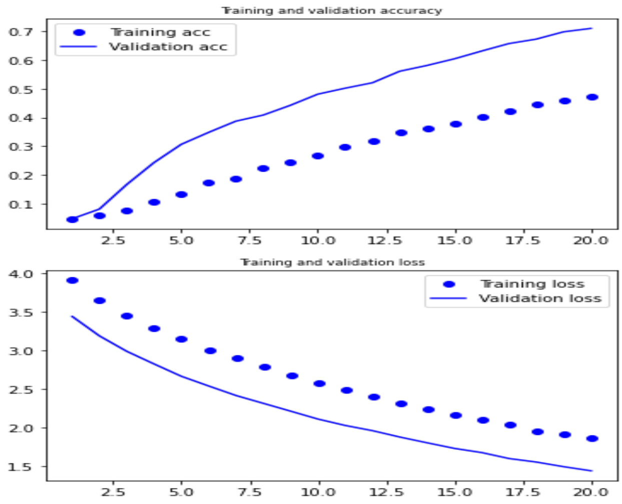

Fig. 7. Plot result of le-6.

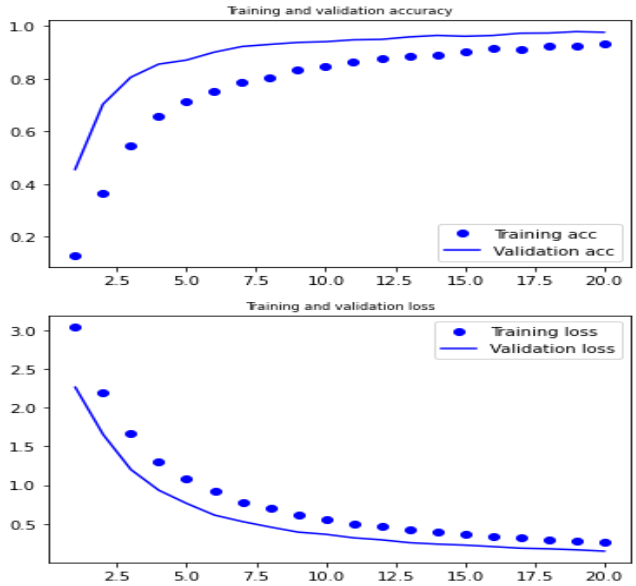

Fig. 8. Plot result of 30 epochs.

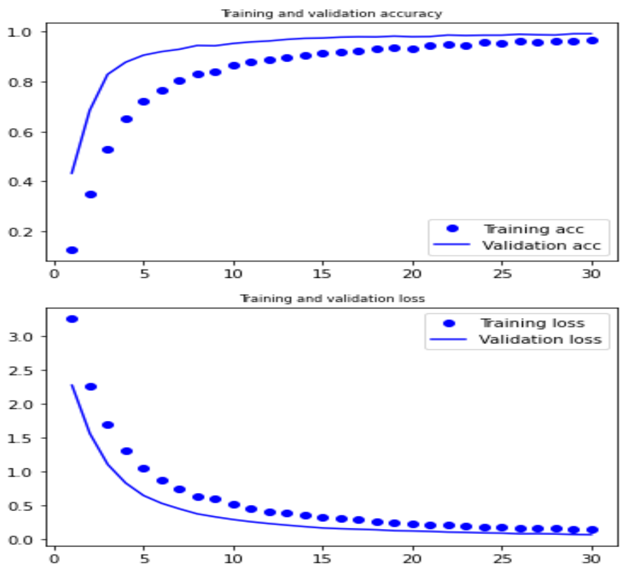

Fig. 9. Plot result of 20 epochs.

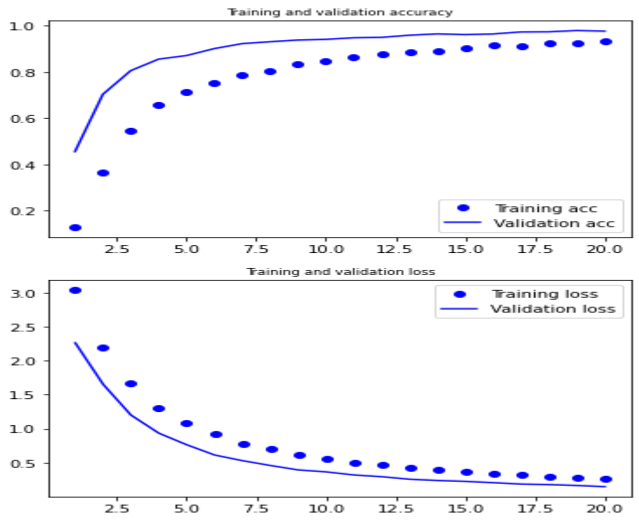

Fig. 10. Plot result with ReLU

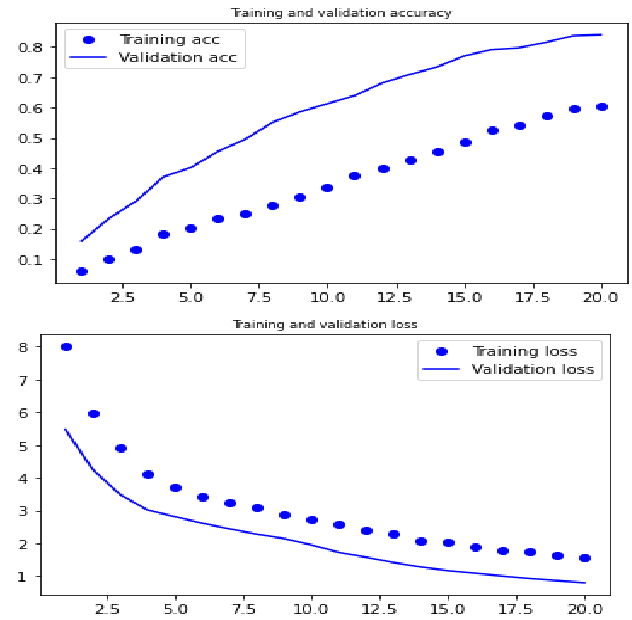

Fig. 11. Plot result with Softmax.

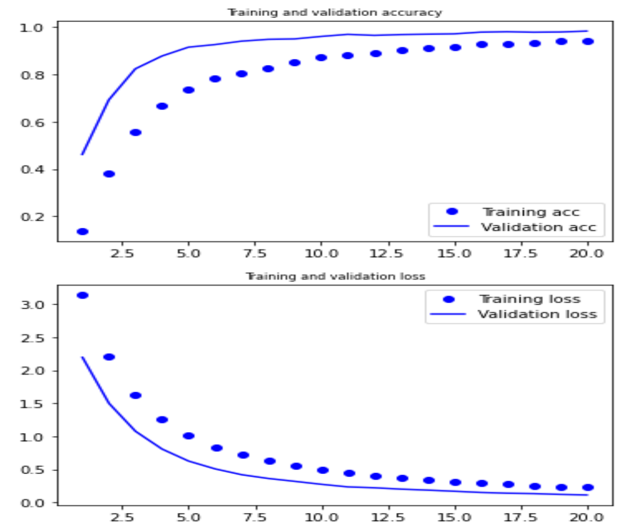

Fig. 12. Plot result with Sigmoid.

B. *Discussion*

The results produced by are Convolutional Neural network affected by the hyperparameters such as learning rate, number of epochs, and activation function.

Although we have done some modification on the learning rate, we can easily identify the optimal option will be the default learning rate at le-5. Lower learning rate at le-3 allows the model to be trained more optimal. Yet, it takes much more time for the training process to be done. In our experiment, the running time is 1.22h higher than the running time of le-5, but with a lower validation accuracy. From TABLE II, we can see that we have done the experiments with parameters other than le-3. Those are learning rate which is lower than le-5, they do take shorter time but with inconsiderable accuracy. Other than that, TABLE III shows that the model with 30 epochs is better. It generated a validation accuracy 0.0147 higher than the default 20 epochs, taking only 0.19h longer. The experiments with different activation functions also returned us different results. We can see that the ReLU activation function did not produce an ideal accuracy, while the accuracy produced by the Softmax, and Sigmoid activation function are close to each other. However, the Sigmoid activation function took 1.1h longer to produce an accuracy that is only 0.0055 higher. Thus, we can conclude that the optimal parameter settings are learning rate at le-5, 30 epochs or more, and with Softmax activation function.

ACKNOWLEDGMENT

We would like to thank for the generous support from Kaggle for providing the dataset which made this research to be finished.

REFERENCES

Abdo, N. A., Yusop, R. bin, & Abdulla, R. (2022). Obstacle Avoidance Robot Using Convolutional Neural Network. Journal of Applied Technology and Innovation, 6(4), 27–33.

Akbari Fard, M., Hadadi, H., & Tavakoli Targhi, A. (2016). Fruits and Vegetables Calorie Counter Using Convolutional Neural Networks. Proceedings Of The 6Th International Conference On Digital Health Conference. doi: 10.1145/2896338.2896355

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021, March 31). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00444-8

Bandyopadhyay, H. (2022, August 3). Image Classification Explained [+V7 Tutorial]. V7. Retrieved September 11, 2022, from https://www.v7labs.com/blog/image-classification-guide#h1

Fang, W., Love, P. E., Luo, H., & Ding, L. (2020, January). Computer vision for behaviour-based safety in construction: A review and future directions. Advanced Engineering Informatics, 43, 100980. https://doi.org/10.1016/j.aei.2019.100980

Feng, J., Zeng, L., & He, L. (2019, February 23). Apple Fruit Recognition Algorithm Based on Multi-Spectral Dynamic Image Analysis. Sensors, 19(4), 949. https://doi.org/10.3390/s19040949

Fruit-Classification/v17_fruit_classify_with_pretrainedmodel.py at master · MeAmarP/Fruit-Classification. GitHub. (2019). Retrieved 2 July 2022, from https://github.com/MeAmarP/Fruit-Classification/blob/master/v17_fruit_classify_with_pretrainedmodel.py.

Geng, L., Huang, Y., Wang, W., Song, Y., Song, N., Zhao, H., & Guo, Q. (2021, April 23). Research on Apple Variety Classification Based on the Combination of Hyperspectral and Deep Learning. 2021 7th International Conference on Computing and Artificial Intelligence. https://doi.org/10.1145/3467707.3467716

Gill, N. (2022). Artificial Neural Networks Applications and Algorithms. Xenonstack.com. Retrieved 9 September 2022, from https://www.xenonstack.com/blog/artificial-neural-network-applications.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377. https://doi.org/10.1016/j.patcog.2017.10.013

Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. Frontiers in Human Neuroscience, 3. https://doi.org/10.3389/neuro.09.031.2009

Li, H. C., Deng, Z. Y., & Chiang, H. H. (2020, October 27). Lightweight and Resource-Constrained Learning Network for Face Recognition with Performance Optimization. Sensors, 20(21), 6114. https://doi.org/10.3390/s20216114

Liu, D., Shen, J., Yang, H., Niu, Q., & Guo, Q. (2019, January 25). Recognition and localization of actinidia arguta based on image recognition. EURASIP Journal on Image and Video Processing, 2019(1). https://doi.org/10.1186/s13640-019-0419-6

Lowe, D. G. (2004, November). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

Marr, B. *What Are Artificial Neural Networks - A Simple Explanation For Absolutely Anyone | Bernard Marr*. Bernard Marr. Retrieved 9 September 2022, from https://bernardmarr.com/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/.

Mishra, M. (2020). Convolutional Neural Networks, Explained. Medium. Retrieved 10 September 2022, from https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

ML Practicum: Image Classification | Machine Learning | . (n.d.). Google Developers. Retrieved September 11, 2022, from https://developers.google.com/machine-learning/practica/image-classification?hl=zh_cn

N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

Nganga, K. (2022). Building A Multiclass Image Classifier Using MobilenetV2 and TensorFlow. Engineering Education (EngEd) Program | Section. Retrieved 11 September 2022, from https://www.section.io/engineering-education/building-a-multiclass-image-classifier-using-mobilenet-v2-and-tensorflow/#:~:text=MobilenetV2%20is%20a%20pre%2Dtrained,thereby%20saving%20time%20for%20development.

Palaz, D., Magimai-Doss, M., & Collobert, R. (2019, April). End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. Speech Communication, 108, 15–32. https://doi.org/10.1016/j.specom.2019.01.004

Purnamasari, P., Taqiyuddin, M., & Ratna, A. (2017). Performance comparison of text-based sentiment analysis using recurrent neural network and convolutional neural network. Proceedings Of The 3Rd International Conference On Communication And Information Processing - ICCIP '17. doi: 10.1145/3162957.3163012

Purnamasari, P., Taqiyuddin, M., & Ratna, A. (2017). Performance comparison of text-based sentiment analysis using recurrent neural network and convolutional neural network. Proceedings Of The 3Rd International Conference On Communication And Information Processing - ICCIP '17. doi: 10.1145/3162957.3163012

Qu, Y., Song, H., Liang, H., Ma, J., & Wang, W. (2019). Protein Image Classification based on Convolutional Neural Network and Recurrent Neural Network. Proceedings Of The 2019 3Rd International Conference On Computer Science And Artificial Intelligence. doi: 10.1145/3374587.3374617

Qu, Y., Song, H., Liang, H., Ma, J., & Wang, W. (2019). Protein Image Classification based on Convolutional Neural Network and Recurrent Neural Network. Proceedings Of The 2019 3Rd International Conference On Computer Science And Artificial Intelligence. doi: 10.1145/3374587.3374617

S. M. H. Khan, A. Hussain and I. F. T. Alshaikhli, "Comparative Study on Content-Based Image Retrieval (CBIR)," 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), 2012, pp. 61-66, doi: 10.1109/ACSAT.2012.40.

Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way. Medium. Retrieved 11 September 2022, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

Sahoo, S. (2021). How to Train MobileNetV2 On a Custom Dataset. Roboflow Blog. Retrieved 11 September 2022, from https://blog.roboflow.com/how-to-train-mobilenetv2-on-a-custom-dataset/.

SARI, A., SETIAWAN, H., ADIPUTRA, T., & WIDYANANDA, J. (2021). FRUIT CLASSIFICATION QUALITY USING CONVOLUTIONAL NEURAL NETWORK AND AUGMENTED REALITY. Jatit.org. Retrieved 11 September 2022, from http://www.jatit.org/volumes/Vol99No22/7Vol99No22.pdf.

Shahi, T. B. (2022, February 25). *Fruit classification using attention-based MobileNetV2 for industrial applications*. PLOS ONE. Retrieved September 13, 2022, from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0264586

Sharma, D. (2021, January 14). *Image Classification Using Convolutional Neural Networks: A step by step guide*. Analytics Vidhya. Retrieved September 13, 2022, from https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/

Sharma, N., Jain, V., & Mishra, A. (2018d). An Analysis Of Convolutional Neural Networks For Image Classification. Procedia Computer Science, 132, 377–384. https://doi.org/10.1016/j.procs.2018.05.198

Xiang, Q., Wang, X., Li, R., Zhang, G., Lai, J., & Hu, Q. (2019). Fruit Image Classification Based on MobileNetV2 with Transfer Learning Technique. Proceedings Of The 3Rd International Conference On Computer Science And Application Engineering - CSAE 2019. doi: 10.1145/3331453.3361658

Y. -m. Cheung and J. Deng, "Ultra local binary pattern for image texture analysis," Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), 2014, pp. 290-293, doi: 10.1109/SPAC.2014.6982701.

Zach, Z. (2020, October 30). *How to Sort a Data Frame by Date in R (With Examples)*. Statology. Retrieved August 22, 2022, from https://www.statology.org/sort-by-date-in-r/