

Smart GreenGrocer: Automatic Vegetable Type Classification Using the CNN Algorithm

Raden Bagus Muhammad AdryanPutra Adhy Wijaya¹, Delfia Nur Anrianti Putri*²,
Dzikri Rahadian Fudholi³

^{1,2}Program Studi S1 Ilmu Komputer FMIPA UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ¹raden.bag2003@mail.ugm.ac.id, *²delfia.nur2004@mail.ugm.ac.id,

³dzikri.r.f@ugm.ac.id

Abstrak

Dalam industri makanan, proses pemisahan sayuran pada umumnya dilakukan oleh tenaga manusia yang terlatih secara visual. Namun, karena terdapat kekurangan dalam penggunaan tenaga manusia, seperti banyaknya waktu yang diperlukan dan sering terjadinya kesalahan, hal ini tidak optimal. Oleh karena itu, proses otomatisasi diperlukan untuk menangani keterbatasan tersebut. Penggunaan computer vision dapat membantu mengurangi kebutuhan sumber daya manusia dengan cara melakukan otomatisasi pada proses klasifikasi. Sayuran memiliki berbagai warna dan bentuk, sehingga proses klasifikasinya menjadi proses klasifikasi multikelas yang rumit dikarenakan oleh keragaman intraspecies dan kesamaan interspecies dari karakteristik-karakteristik pembedanya. Untuk menjawab tantangan ini, kami mengusulkan solusi menggunakan deep learning dengan Convolutional Neural Network (CNN) untuk melakukan klasifikasi multi-label pada beberapa tipe sayuran. Kami melakukan beberapa percobaan pada model CNN saat dilakukan proses pelatihan dengan mengkombinasikan parameter-parameter yang ada, seperti menggunakan beberapa nilai dari batch_size dan beberapa jenis optimizer. Pada proses pelatihan di awal, kami menggunakan nilai learning rate 0.01 dan akan semakin mengecil jika telah mencapai lokal minimum sehingga hasil yang optimal akan didapatkan. Klasifikasi dilakukan pada 15 tipe sayuran dan menghasilkan akurasi sebesar 98.1% pada data testing dengan waktu pelatihan 25 menit 45 detik.

Kata kunci—Pembelajaran Mesin Mendalam, Adamax, Optimasi Parameter CNN, Rmsprop, SGD

Abstract

In the food industry, separating vegetables is done by visually trained professionals. However, because it takes plenty of time to sort a large number of different types of vegetables, human errors might arise at any time, and using human resources is not always effective. Thus, automation is needed to minimize process time and errors. Computer vision helps reduce the need for human resources by automatizing the classification. Vegetables come in various colors and shapes; thus, vegetable classification becomes a challenging multiclass classification due to intraspecies variety and interspecies similarity of these main distinguishing characteristics. Consequently, much research is made to automatically discover effective methods to group each type of vegetable using computers. To answer this challenge, we proposed a solution utilizing deep learning with a Convolutional Neural Network (CNN) to perform multi-label classification on some types of vegetables. We experimented with the modification of batch size and optimizer type. In the training process, the learning rate is 0.01, and it adapts on arrival in the local minimum for result optimization. This classification is performed on 15 types of vegetables and produces 98.1% accuracy on testing data with 25 minutes and 45 seconds of training time.

Keywords— Deep Learning, Adamax, CNN Parameter Optimization, RMSProp, SGD

1. INTRODUCTION

Vegetables have been a necessary global food resource over time. Each type of vegetable carries its benefits for the life of other living beings. Nutrition and vitamins inside are advantageous for the growth and development of humans and animals, for instance, to improve eyesight, maintain digestive system health, and reduce cancer risk [1].

Before vegetables are distributed, they need to be sorted based on their type and category. The variety of shapes, colors, and textures of each type of vegetable could be distinguished by humans in the manual sorting process. However, because it takes plenty of time to sort a large number of different types of vegetables, human errors might arise at any time, and using human resources to sort vegetables is not always effective [2].

Consequently, some research has been done to discover effective methods and approaches to automatically group each type of vegetable with the help of computers. A study has developed vegetable classification using a Support Vector Machine (SVM) that uses the color and shape of their leaf[3]. A paper on this topic offers a solution to classify vegetables using a machine learning algorithm Support Vector Machine (SVM) utilizing color features and Saliency-HOG[4]. Other studies also proposed an automatic categorization with the Naive Bayes approach using Hue Saturation Value or HSV as its feature[5]. However, conventional machine learning methods have some downsides, like the need to do feature engineering manually and the requirement of the high cost of computation.

Several studies, such as the CNN model, use deep learning to solve problems. A paper offers a solution to classify children's facial emotions using the CNN model, which achieves an accuracy score of 99.9%[6]. A study has also developed a CNN model for solving mobile-based image recognition, which can reach 93.6% for the average accuracy score[7]. Other studies have offered a solution for classifying vehicle systems using the CNN model[8]. We can conclude that the CNN model can solve many problems with computer vision.

Thus, this paper proposes a classification model that can classify the types of vegetables with the Convolutional Neural Network. We build an effective model to solve the problem arising around multi-label vegetable classification. In this study, we offer these specific contributions as follows:

1. We introduced a novel method for solving vegetable classification problems with a CNN algorithm to train and develop the model.
2. We built a model to perform multi-label classifications on some vegetable types. This classification model could answer vegetable categorization problems involving intraspecies variety and interspecies similarity of color and shapes. This model could further replace conventional machine learning methods to distinguish different types of vegetables.
3. We tested our proposed model to obtain better accuracy in a short time. We also tuned the parameters needed so that the trained model received the best possible accuracy rate.

2. METHODS

2.1 Main Idea

The primary goal of this paper is to build a robust model for vegetable classification from an image dataset containing images from 15 types of vegetables using the CNN algorithm. CNN is well known for its high-accuracy prediction; thus, we use this supervised learning algorithm to create a multi-label classifier that could map vegetable image data to its pre-existing predicted categories. Following steps are taken to make the model. Firstly, we acquire the dataset and augment the images. Then, we fit the model with the train data and obtain the performance score with accuracy.

2.2 Dataset

In this study, we gather the dataset Vegetable Image Dataset[9] from Kaggle containing 21.000 colored images with 15 types of vegetables, as shown in Figure 1. Some variations in the position and number of vegetables of the same species are contained in the picture. The picture could also contain other than vegetable objects, such as hands, as depicted in the Brinjal labeled image in Figure 1. This variation means that the dataset we use for model training represents real-world data. To generate the model, we separate the dataset into three parts: the training dataset to build the model, the validation dataset to evaluate the model while training, and the testing dataset to evaluate the model's performance. Table 1 depicts the dataset distribution. We use 71,4% of the dataset to train the model; the rest is for validation 14,3% and testing 14.3%.

Table 1. Dataset Distribution.

Dataset	Sample
Data Training (71,4%)	15000
Data Validation (14,3%)	3000
Data Testing (14.3%)	3000
Total	21000

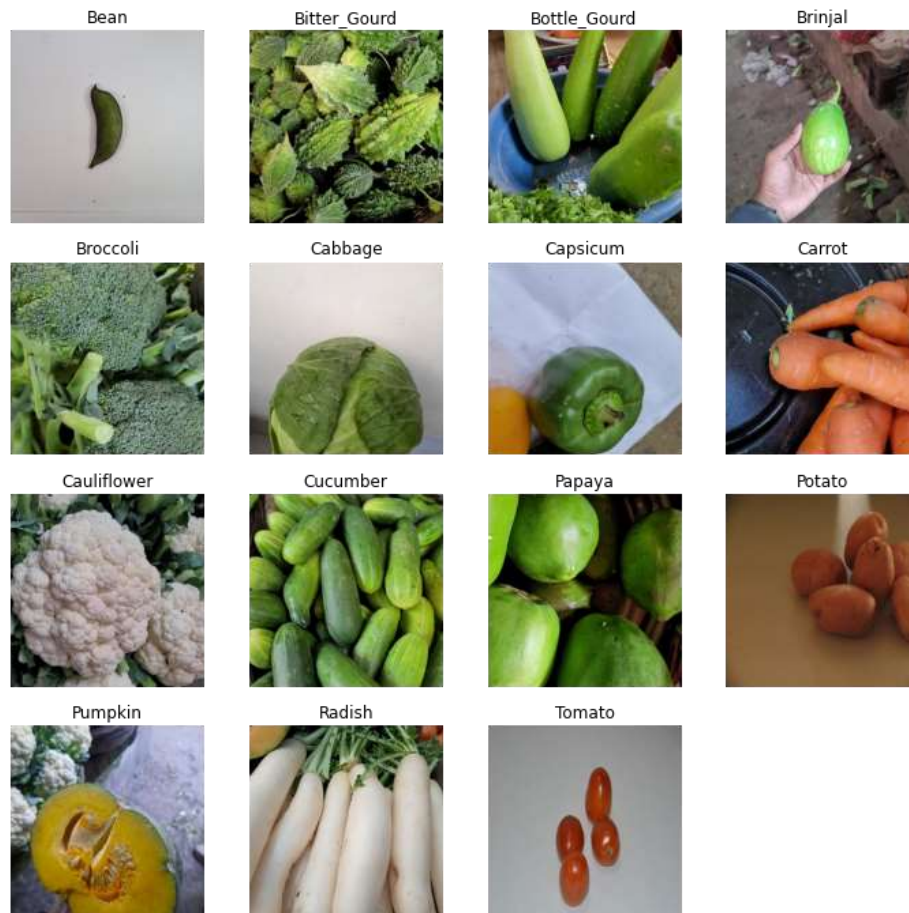


Figure 1. Some samples of the dataset that represent the 15 classes

2.3 Data pre-processing

In this stage, we augment the images in the training and validation set by doing the following processes: rescaling, rotation, width and height shifting, shearing, zooming, and flipping. Throughout the augmentation process, the dataset splits into training and validation

parts. Here, we set the fill mode as the nearest filling, the batch size as 32, and the image target size as 150x150 pixels. Afterward, we encode the classes with indices to indicate the label of each type of vegetable image.

2.2 Classification

2.2.1 CNN

Deep learning is a process of artificial intelligence imitating how brain cells work to solve problems. One deep learning method well-suited for 2-dimensional data is Convolutional Neural Network (CNN). The ability of this algorithm to detect and learn the features of CNN makes it advantageous for image classification tasks. Having similar characteristics as other neural network-based algorithms, CNN also consists of many neurons with weight, bias, and activation functions.

2.2.1.1 Feature learning

This stage is an encoding of an image into features that represent it. This process consists of some layers working together to obtain the characteristics of the image, specifically the convolution layer, activation function layer, and pooling layer. The first type of layer is the convolution layer. In this layer, the filtering process with the sliding window concept occurs by using a filter with a predefined size and value. This filter acts as a multiplier in which dot multiplication is done on each intensity of an area of a matrix where the filter lies with the filter. The resulting matrix is then passed onto the next layer for further computation. The model's training utilizes the filters in this layer and the back-propagation of the model's parameters. Next is the activation function. ReLU is one of the most well-known activation functions. It maps all non-positive numbers as zero, and the rest is linear. The pooling layer is vital in reducing the dimension of resulting convolution matrices. This process enables faster computation due to the requirement of a smaller number of parameters to update, which lowers the risk of overfitting. There are two main types of pooling layers: max pooling and average pooling. Max pooling takes the maximum value of each matrix input, and the other computes the average of all entries in the filter matrix range[10].

2.2.1.2 Classification

Classification of each neuron that has been extracted in feature learning is done by some types of layers, namely the input layer, hidden layer specifically with dropout regularization, and output layer. The input layer is a one-dimensional array, a flattened array from the previously computed feature map matrix. Each neuron of this layer is connected to a neuron in the hidden layers. The hidden layers use multi-layer perceptron (MLP) to transform the dimension of the data to be linearly classified by utilizing backward and forward propagation. Dropout is a handy regularization technique that shuts down neurons in the hidden layers to limit the contribution of some neurons to the output. As a result, other than reducing the risk of overfitting, which becomes a common problem in CNN, a less extensive computation is needed to complete the process. The output layer is the final layer where each input goes into an activation function, usually softmax, to produce a probabilistic output ranging from 0 to 1[11].

2.2.2 Modeling

Our proposed model utilizes CNN as the backbone of the predictive model. The architecture of the model is shown in Figure 2. This architecture consists of twelve layers in which 2-dimensional convolution layers are used in the beginning, each followed by a max pooling layer. Then, the resulting matrices from the last max pooling layer are flattened and go through dropout layers before being inserted into the dense layer. Our model contains some parameters, such as 4 convolutional layers (conv2D), 4 pooling layers (MaxPooling2D), 1 layer flatten, 1 layer dropout, and 2 dense layers, as shown in Figure 3. The first dense layer is used as the hidden

layer, and the second hidden layer is used as the output layer, giving the number between 1 and 15 as the output.

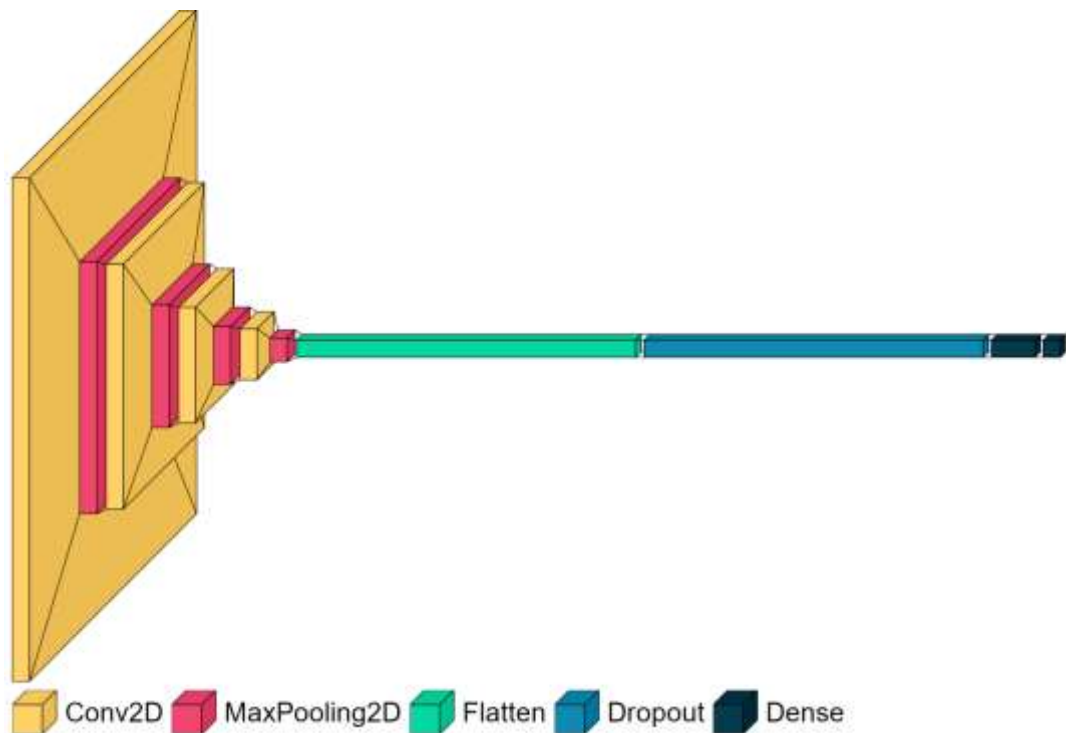


Figure 2. Model Architecture of Smart GreenGrocer.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 74, 74, 64)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	36928
max_pooling2d_1 (MaxPooling2)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_3 (MaxPooling2)	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dropout (Dropout)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776
dense_1 (Dense)	(None, 15)	7695
Total params: 3,479,631		
Trainable params: 3,479,631		
Non-trainable params: 0		

Figure 3. Parameters of CNN Model.

We feed the model with augmented data split into training and validation portions. The training data is used to develop the ability of the model to classify different varieties of vegetables from the image data while it is validated using the validation portion. The resulting model is then tested, and the performance score is obtained using the testing data.

3. RESULT AND DISCUSSION

In this research, the computing device used is GPU GTX 980 Ti. The setting of the epoch is 50 throughout the training and testing phase. To optimize calculations, we use early stopping criteria and reduce the learning rate on a plateau for callbacks with parameters, as stated in Table 2.

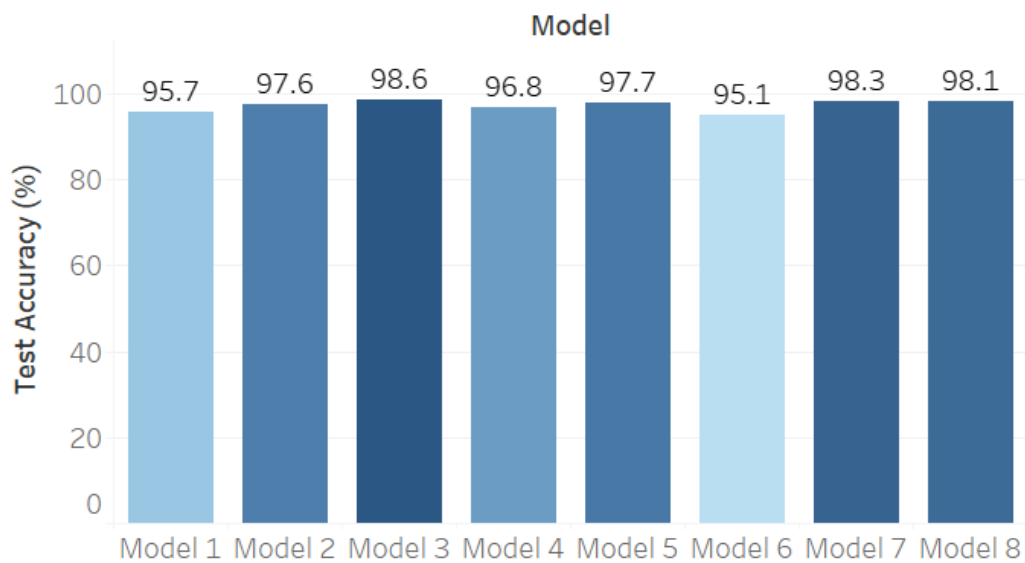
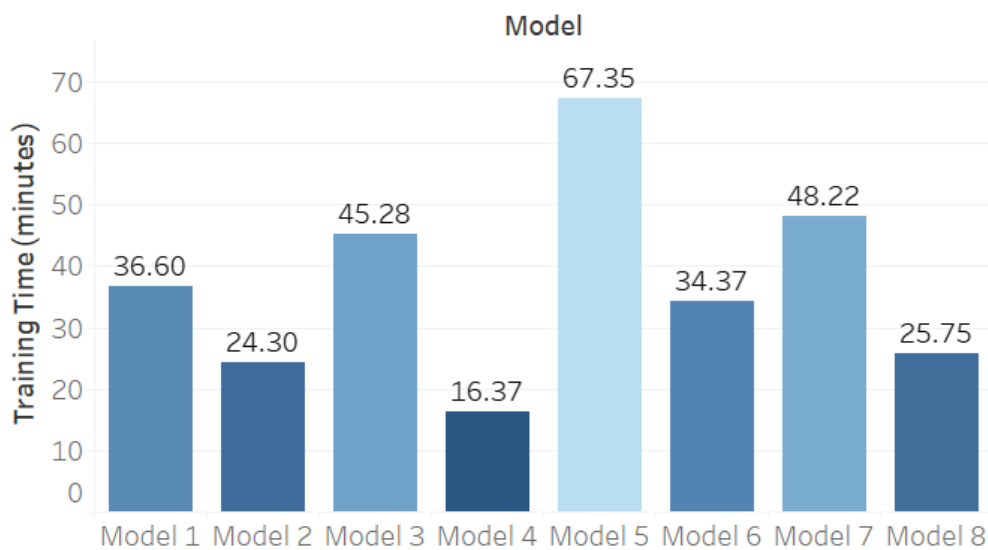
Table 2. Parameters Of Callbacks.

Callbacks	Parameters	Value
Early Stopping	min_delta	0.01
	patience	10
	restore_best_weights	True
Reduce Learning Rate On Plateau	factor	0.1
	patience	5
	min_delta	0.01

We also vary the parameters on optimizer type and batch size to obtain the best result possible based on the computation time and accuracy of the model. Table 3 below provides detail on the result obtained with different parameter settings, and Figure 4 and 5 shows the visualization of the metrics for each configuration.

Table 3. Experiment Results.

Model	Optimizer	Batch Size	Test Loss	Test Accuracy	Training Time
1	Adam	32	0.1658	0.9573	36m 36s
2		64	0.0790	0.9757	24m 18s
3	RMSProp	32	0.0712	0.9863	45m 17s
4		64	0.1028	0.9677	16m 22s
5	SGD	32	0.0725	0.9773	1h 7m 21s
6		64	0.1716	0.9510	34m 22s
7	Adamax	32	0.0484	0.9833	48m 13s
8		64	0.0706	0.9810	25m 45s

**Figure 4.** Test Accuracy of Models.**Figure 5.** Training Time of Models.

Based on the experiment, as shown in table 3 as visualized in Figure 4 dan 5, the most optimal configuration based on training time is CNN with RMSProp optimizer with 64 training batches with an accuracy of 96.77% and training time of 16 minutes and 22 seconds. Meanwhile, the highest accuracy achieved, 98.63%, is obtained using the configuration of RMSProp optimizer with 32 training batches.

However, we can gain a similar result with a shorter training time with the Adamax optimizer in terms of accuracy and loss. The most optimal model we obtained is model 8, with 98.1% testing accuracy in 25 minutes and 45 seconds of training. Furthermore, the second-best accuracy is obtained from a model that uses Adamax as its optimizer. This accuracy score implies the proper ability of the model to classify images of the given vegetables. This configuration also depicts a more stable tradeoff of training and validation accuracy and loss throughout the training process, as shown in Figure 8, compared to model 3, which gives the best accuracy in this experiment.

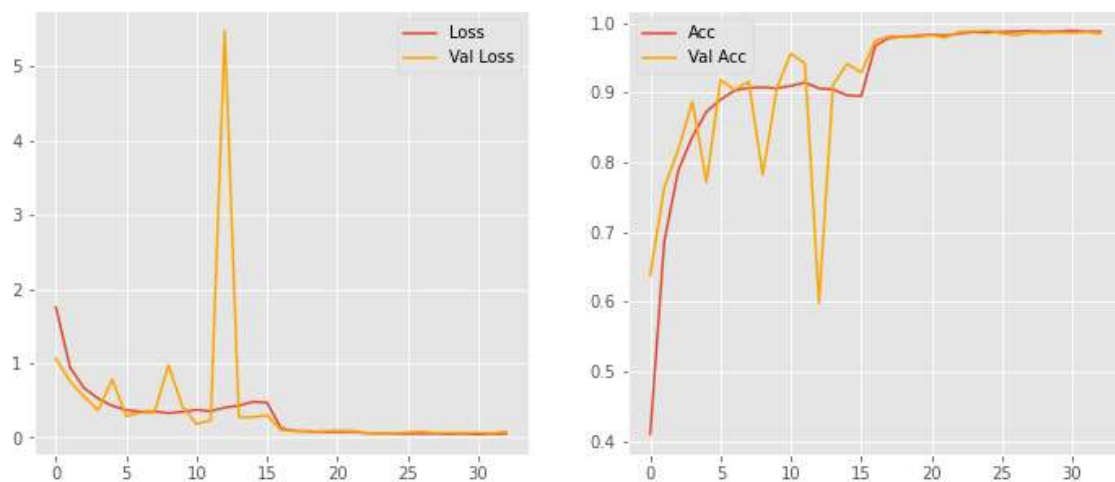


Figure 6. Training and validation accuracy and loss for model 3.

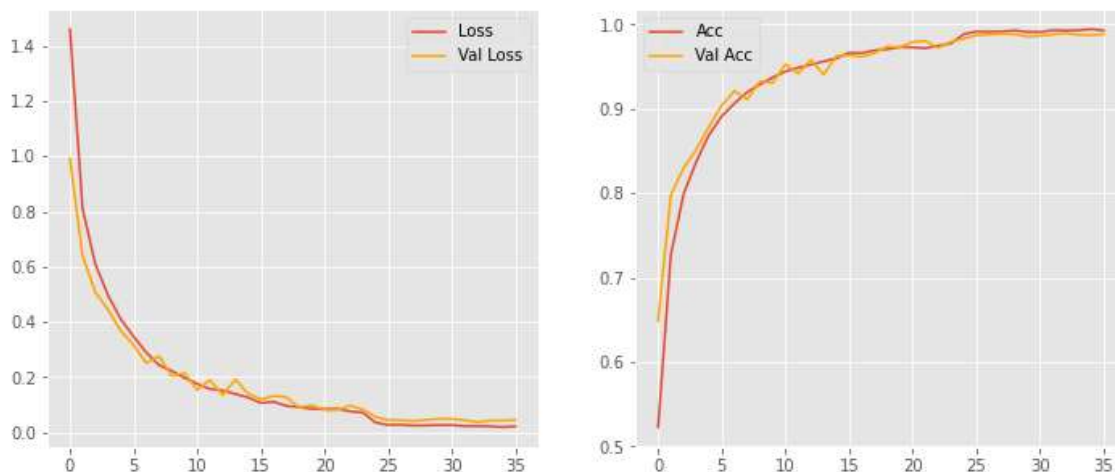


Figure 7. Training and validation accuracy and loss for model 7.

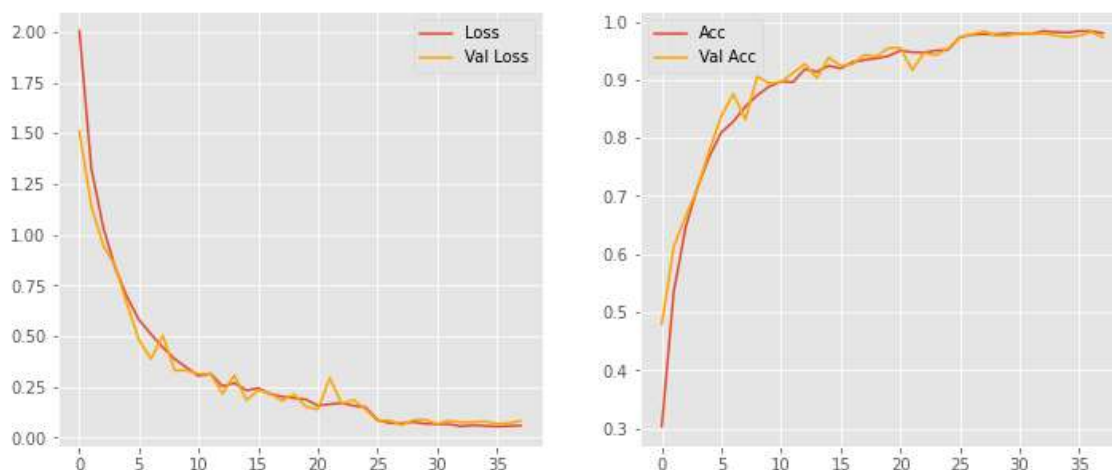


Figure 8. Training and validation accuracy and loss for model 8.

Figures 6, 7, and 8 show the tradeoff of training and validation accuracy and loss for parameter configuration of models 3, 7, and 8, respectively. The training for model 3 shows an extreme oscillation of validation loss and accuracy as the epoch increases, while models 7 and 8 give a more stable and smoother oscillation. This means model 7 and 8, which uses Adamax as their optimizers, are more stable during the training process.

4. CONCLUSION

We conclude that the most optimal configuration based on training time is CNN with RMSProp optimizer with 64 training batches. The highest accuracy is obtained using the configuration of RMSProp optimizer with 32 training batches. However, a more optimal result with a high accuracy rate and shorter training time is obtained with the Adamax optimizer using the batch size of 64, which produces an accuracy of 98.10% in 25 minutes and 45 seconds of training.

In the future study, we can use another algorithm to improve this model using Generative Adversarial Network (GAN) and Graph Convolutional Network (GCN) architecture. To produce higher-quality accuracy, we can use dynamic neural networks with additional features that can be developed.

REFERENCES

- [1] A. Zurbau *et al.*, "Relation of Different Fruit and Vegetable Sources With Incident Cardiovascular Outcomes: A Systematic Review and Meta-Analysis of Prospective Cohort Studies," *J. Am. Heart Assoc. Cardiovasc. Cerebrovasc. Dis.*, vol. 9, no. 19, p. e017728, Oct. 2020, doi: 10.1161/JAHA.120.017728.
- [2] M. Mukhiddinov, A. Muminov, and J. Cho, "Improved Classification Approach for Fruits and Vegetables Freshness Based on Deep Learning," *Sensors*, vol. 22, no. 21, p. 8192, Oct. 2022, doi: 10.3390/s22218192.
- [3] N. A. Nasharuddin, "Multi-feature Vegetable Recognition using Machine Learning Approach on Leaf Images," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 6, pp. 1789–1794, Aug. 2019, doi: 10.30534/ijatcse/2019/110842019.
- [4] Y. Yohannes, M. R. Pribadi, and L. Chandra, "Klasifikasi Jenis Buah dan Sayuran Menggunakan SVM Dengan Fitur Saliency-HOG dan Color Moments," *ELKHA J. Tek. Elektro*, vol. 12, no. 2, Art. no. 2, Oct. 2020, doi: 10.26418/elkha.v12i2.42160.
- [5] H. Nurrani, A. K. Nugroho, S. Heranurweni, E. Supriyanto, and Generousdi -, "VEGETABLE TYPE CLASSIFICATION USING NAIVE BAYES ALGORITHM

- BASED ON IMAGE PROCESSING,” *JAICT*, vol. 7, no. 2, Art. no. 2, Oct. 2022, doi: 10.32497/jaict.v7i2.3762.
- [6] Y. Ihza and D. Lelono, “Face Expression Classification in Children Using CNN,” *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 16, no. 2, Art. no. 2, Apr. 2022, doi: 10.22146/ijccs.72493.
- [7] N. Wiranda and A. E. Putra, “Mobile-based Primate Image Recognition using CNN,” *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 16, no. 2, Art. no. 2, Apr. 2022, doi: 10.22146/ijccs.65640.
- [8] W. Wahyono and J. Hariyono, “Determining Optimal Architecture of CNN using Genetic Algorithm for Vehicle Classification System,” *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 13, no. 1, Art. no. 1, Jan. 2019, doi: 10.22146/ijccs.42299.
- [9] M. I. Ahmed, S. Mamun, and A. Asif, *DCNN-Based Vegetable Image Classification Using Transfer Learning: A Comparative Study*. 2021, p. 243. doi: 10.1109/ICCCSP52374.2021.9465499.
- [10] J. Shah and S. Kamat, “A Method for Waste Segregation using Convolutional Neural Networks.” arXiv, Feb. 23, 2022. Accessed: Jan. 27, 2023. [Online]. Available: <http://arxiv.org/abs/2202.12258>
- [11] A. Sharma and G. Phonsa, “Image Classification Using CNN.” Rochester, NY, Apr. 24, 2021. doi: 10.2139/ssrn.3833453.