# Comparative study on fine-tuning deep learning models for fruit and vegetable classification

**Abd Rasid Mamat[1], Mohamad Afendee Mohamed[1], Mohd Fadzil Abd Kadir[1], Norkhairani Abdul Rawi[2], Azim Zaliha Abd Aziz[2], Wan Suryani Wan Awang[1]**
[1]Department of Computer Science, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
[2]Department of Multimedia Studies, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia

**Article Info**

***Article history:***

Received Jul 21, 2024
Revised Apr 1, 2025
Accepted Apr 23, 2025

***Keywords:***

Accuracy
Classification
Convolution neural network
Hyperparameter
Loss

**ABSTRACT**

Fruit and vegetable recognition and classification can be a challenging task due to their diverse nature and have become a focal point in the agricultural sector. In addition to that, the classification of fruits and vegetables increases the cost of labor and time. In recent years, deep learning applications have surged to the forefront, offering promising solutions. Particularly, the classification of fruits using image features has garnered significant attention from researchers, reflecting the growing importance of this area in the agricultural domain. In this work, the focus was on fine-tuning hyperparameters and the evaluation of a state-of-the-art deep convolutional neural network (CNN) for the classification of fruits and vegetables. Among the hyperparameters studied are the number of batch size, number of epochs, type of optimizer, rectified unit, and dropout. The dataset used is the fruit_vegetable dataset which consists of 36 classes and each class contains 1,000 images. The results show that the proposed model based on the batch size=64 and the number of epochs=25, produces the most optimal model with an accuracy value (training) of 99.02%, while the validation is 95.73% and the loss is 6.06% (minimum).

***Corresponding Author:***

Abd Rasid Mamat
Department of Computer Science, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin
Besut Campus, 22200 Besut, Terengganu Darul Iman, Malaysia
Email: arm@unisza.edu.my

## 1. INTRODUCTION

Image classification has its roots in a wide range of technologies developed over many years [1]–[3]. These technologies include advancements in computer vision, machine learning, and pattern recognition. Researchers and engineers have worked extensively to improve the ability of machines to recognize and categorize images accurately. As a result of these efforts, image classification systems have become incredibly sophisticated and powerful. The applications of these classifiers are both varied and widely used across many different fields. Some applications focus on broad tasks, such as interpreting and understanding the overall context of an image. Others are designed for highly specific and critical purposes, like detecting signs of cancer in patient test results [2], [4], [5]. Overall, the impact of image classification technologies continues to grow, influencing industries from agriculture, transportation, and military to name a few.

Research in fruit processing holds substantial significance when connected with other economic sectors like agriculture, encompassing both wholesale and retail markets, alongside the processing industry. These factors have pushed researchers who developed various methods to automatically process fruits, either to identify or estimate their quality efficiently, classification, and packaging towards more focused research. This is because the processing, classification, and separation of plants require intensive energy and time [6].

The most efficient classifiers that perform well in classifying images such as fruits are those by using deep learning algorithms [7]–[9]. Deep learning, although a theoretical concept, is nothing new. It has enjoyed a surge of interest over the past decade, becoming the hottest trend in machine learning due to many factors. Deep learning approaches have significantly outperformed state-of-the-art approaches in many tasks across different fields such as data intrusion, image classification, computer vision, speech processing, and natural language processing (NLP) [10]–[12]. However, there remains an opportunity to explore hyperparameters in deep learning to achieve the best classification results through fine-tuning. The focus is on the batch size, the number of epochs, and their relationship, while the other hyperparameters are set as outlined in section 2.2.

Several studies related to automatic fruit classification have been done so far by many scientists and researchers. It involves fruit classification, determining ripeness stage, disease detection, assessing citrus level, and classifying fresh fruit palm oil in ripe bunches [13]–[15]. For example, for the process of harvesting dates based on 5 different classifications of dates, a robotic harvesting model was proposed by Altaheri *et al.* [16]. The model uses an internal dataset containing 8,000 images for training and testing and achieves 99% accuracy.

Other researchers in [17], developed a fruit classification model for industrial applications. In this model, the authors used public datasets and one of the datasets contained images of fruits which are complex to identify. The accuracy of the proposed model is 85%. Next, the authors in their research employed convolutional neural networks (CNNs) to classify fruits and vegetables based on RGB image data [18]. Despite the challenge of some items sharing similar colors and shapes, their approach achieved improved classification accuracy compared to other methods.

Deep learning methods for fruit classification are extensively applied in the post-harvest stage and the fruit industry. In a particular study, a CNN-based model was introduced to categorize apples into normal and defective classes. This model was integrated into a fruit sorting system, demonstrating a remarkable accuracy of 92% while processing each apple in less than 72 milliseconds [19]. Despite the increasing interest in artificial intelligence (AI) for reducing food waste, there remains a notable gap in research concerning the application of AI for classifying and detecting fruits and vegetables. This study aims to fill this gap by designing and implementing an intelligent system capable of accurately identifying 36 different classes of fruits and vegetables through the tuning of hyperparameters.

## 2. RESEARCH METHOD

Deep learning is a highly active research area in computer vision and image classification. A typical architecture of deep CNN comprises an input layer, an output or classification layer, and multiple hidden layers (feature extraction is done in the hidden layers). These hidden layers often include convolutional, pooling, and fully connected layers, along with the possibility of a SoftMax layer [20]–[22]. In general, the hyperparameters are filter size (kernel size), number of filters, pooling, activation function, learning rate, batch size, number of epochs, and dropout [22], [23].

Subsequently, the tuning or adjustment of the hyperparameters is batch size and the number of epochs, while other parameters, namely the activation function, dropout value, learning rate, and optimizer type, are kept fixed. Specifically, the rectified linear unit (ReLU) is utilized as the activation function, Adam serves as the optimizer, and learning rate is an automatic method and a dropout rate of 0.2 (20%) is applied. On the other hand, the variables batch size and number of epochs are variable parameters that will fine-tune the model training process for finding and optimizing the accuracy performance for the data set.

### 2.1. Dataset

Sample images consisting of real-world information are referred to as datasets and digital image collection is known as data acquisition [24]. The dataset used to work is the fruit_vegetable dataset which is publicly available on Kaggle which is publicly available. It is free and downloaded from *https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition*. Many researchers used this dataset for their research [25], [26]. Figure 1 shows the example of images in the dataset.

This dataset contains 36 classes and 1,000 images for each class. Figure 1, shows an example of the different classes of fruits we used for analysis. Then the dataset is split into 3 categories namely training dataset (80%), testing dataset (10%), and finally validation dataset (10%) [8], [27].

#### 2.1.1. Training dataset

Each class in the training dataset contains 100 images, providing a balanced representation for each category. Altogether, the dataset is made up of a total of 3,600 images, ensuring a substantial amount of data for the learning process. This training dataset plays a crucial role in developing the network model by providing the examples needed for learning. It is specifically used to train the parameters of the model, including adjusting the weights and biases to optimize performance.

### 2.1.2. Validation dataset

The validation dataset is used to evaluate the performance of the trained model and ensure it generalizes well to new data. For this particular task, each class in the dataset contains 10 images, providing a smaller but balanced set for assessment. In total, the validation dataset consists of 360 images, which represents 10% of the entire dataset. This validation step is essential for monitoring the model's accuracy and preventing issues such as overfitting.

### 2.1.3. Testing dataset

The aim of using a testing dataset in deep learning is to provide an independent and unbiased evaluation of a trained model's performance. This dataset is crucial because it helps determine how well the model can make predictions on previously unseen data. For this evaluation process, a total of 360 images were used, ensuring a fair and representative test. These images were evenly taken from ten different classes, maintaining balance across the categories.

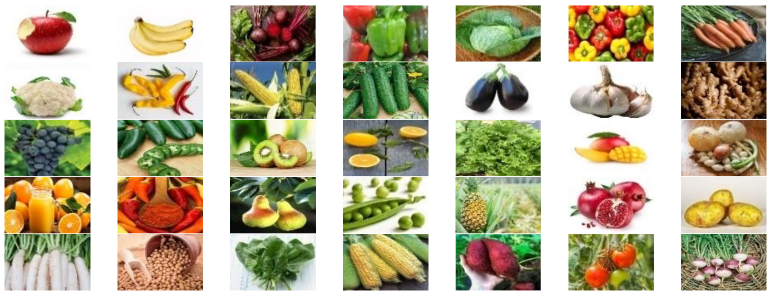


Figure 1. An example of images in a dataset

## 2.2. Tuning hyperparameters of the proposed models

In this work, we have incorporated model-tuning techniques to prevent the model from overfitting. The hyperparameters tuned are the batch size and number of epochs, with a on their relationship and summarized as follows:

i) Batch-size, the batch size refers to the number of training examples utilized in one iteration in training neural networks. Number batch size is used 16, 32, and 64.
ii) Number of epochs, the number of epochs based on the number of times the entire dataset is passed forward and backward through the neural network during the training process. The performance is compared on the epochs number 10, 15, and 25.
iii) Optimizer, Adam is adaptive algorithm and it is used to optimize all the experiments. Typically, when using Adam, the learning rate ranges from 0.0001 to 0.001, with a starting point of 0.001 [27], [28].
iv) ReLU is a popular and common activation function used in deep learning [29]–[31]. The mathematical expression for ReLU is shown in (1).

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

Where x is the input to the activation function and f(x) is the output after applying ReLU activation.
v) The dropout technique helps avoid the issue of overfitting and during the training, neurons are randomly chosen and discarded. In this model, the dropout value is 0.2 [32], [33].

## 2.3. Proposed model

The proposed model is considered one of the most common deep learning architectures [21], [34], as shown in Figure 2. The network architecture of the proposed model is designed as a sequence model to handle sequential data. A summary of the model is as follows:

i) Firstly, the input layer. This layer represents the raw input data, such as an image. Each pixel in the image may be represented as a separate input node. It is followed by a preprocessing layer as a function to normalize the pixel of images into range 0-1 [35]. It is a good practice to normalize the data to avoid different scales of the feature vectors and thus improve data integrity [36], [37].
ii) The next layer is the Conv2D (16,3, ReLU) layer, where this layer contains 16 filters (or kernels) of size 3×3 and the activation are ReLU. Each filter learns different features from the input data.

iii) The MaxPooling2D layer is a pooling operation typically used in CNNs for down-sampling feature maps. It helps reduce the spatial dimensions of the input, thereby reducing computational complexity and controlling overfitting. The next layer is Conv2D (32,3 ReLU). In this case, there are 32 filters and the size is 3×3 ReLU is used for the activation function.

iv) The following layer is Conv2D (64,3 ReLU). This layer shows 64 filters of size 3×3. ReLU is also used ReLU for activation function. Next the flattened layer. This layer refers to the process of converting a multidimensional array (e.g., a 2D) into a one-dimensional array. Flattening bridges this gap by reshaping the 2D feature maps into a 1D array, which can then be fed into the dense layers for further processing and decision-making.

v) The dropout layer is the layer before the dense layer. Dropout is a form of regularization that helps improve the generalization of a neural network by reducing overfitting. Overfitting occurs when a model learns to perform well on the training data but performs poorly on unseen data. The number of 0.2 or 20% refers to during training, which means, 20% of the neurons in the first dense layer will be randomly dropped, helping prevent overfitting and improving the model's ability to generalize to new data.

vi) Lastly, the dense layer is also known as a fully connected layer because each neuron (or unit) in the dense layer is connected to every neuron in the previous layer and every neuron in the subsequent layer. The number 128 in the dense layer refers to this layer has 128 neurons with a ReLU for activation function.

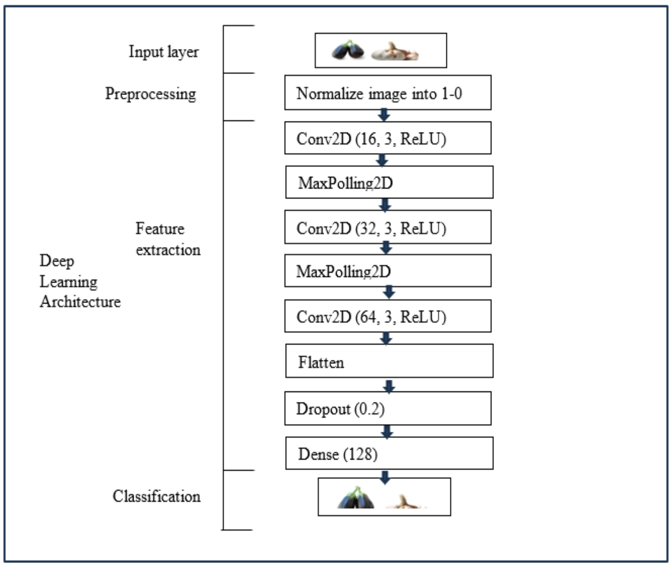


Figure 2. The proposed architecture of deep learning

## 2.4. Performance evaluation

To assess the effectiveness of the proposed classification methodology, we utilize accuracy analysis for training, validation, and testing. Additionally, we incorporate an analysis of the loss function to determine whether the model is overfitting or not. In (2) and (3) are employed to calculate both accuracy and loss.

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} \times 100\% \tag{2}$$

$$Loss\ (Sparse\ Categorical\ Cross - Entropy) = -\frac{1}{N}\sum_{i=1}^{N}(p_i y_i) \tag{3}$$

Where N is the number of samples and $p_i$, $y_i$ is the predicted probability of the true class label $y_i$ for sample i.

## 3. RESULTS AND DISCUSSION

In this study, an assessment of the appropriateness of state-of-the-art deep convolutional neural networks for the task of image classification was done. This focuses on tuning hyperparameters such as the number of batch size and number of epochs. The results of the experiment are discussed in Figures 3-5 and Tables 1-3.

### 3.1. Hardware and software

The baseline system used in our evaluation is a PC. PC configuration for the project, where the processor is an 11th Generation Intel(R), Core (TM) i5-1135G7 @ 2.40 GHz, 2419 Mhz, 4 Cores. OS is Microsoft Windows 10, Phyton, Keras, and Jupyter Lab are used for implementation [38], [39].

### 3.2. Model accuracy and model loss

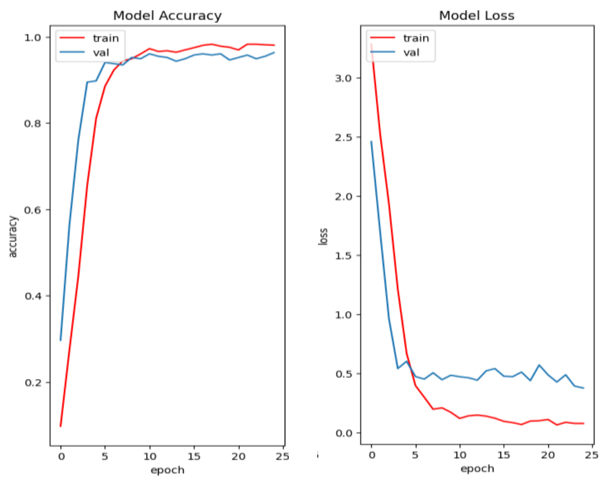
In deep learning model accuracy and model loss are plotted to show the accuracy of the proposed model. In model accuracy, train (training), and val (validate) accuracy verse epoch is plotted. Training accuracy represents the percentage of correctly classified examples in the training dataset and the aim is to measure how well the model fits the training data during the training process. Validate accuracy shows the percentage of correctly classified examples in the validation dataset, which the model has not seen during training. The higher training accuracy and validation accuracy represent an ideal scenario where the model performs well on both the training and validation datasets. It suggests that the model has learned the underlying patterns in the data and is generalized effectively to new examples.

Next, the model loss, training, and validate loss verse epoch are also plotted. Training loss refers to the error or discrepancy between the predicted outputs of a deep neural network and the actual target outputs during the training process. The validation loss serves as a proxy for how well the model is generalizing to new data. A lower validation loss indicates that the model is making more accurate predictions on unseen examples from the validation dataset. The model loss aims to achieve low training and validation losses simultaneously, indicating that the model has learned meaningful patterns from the training data and can generalize effectively to new, unseen data.

Figure 3 shows the model of model accuracy (Figure 3(a)) and model loss (Figure 3(b)) according to train and validation data according to batch size=16, a number of epochs 0 to 25, and the optimization algorithm is Adam. In Figure 3, when the epoch value approaches 15 to 25, the model accuracy (Figure 3(a)) shows a consistent increase in accuracy, reaching a plateau where it remains stable. Similarly, in the model loss (Figure 3(b)), the training loss decreases steadily and then stabilizes within the same epoch range.

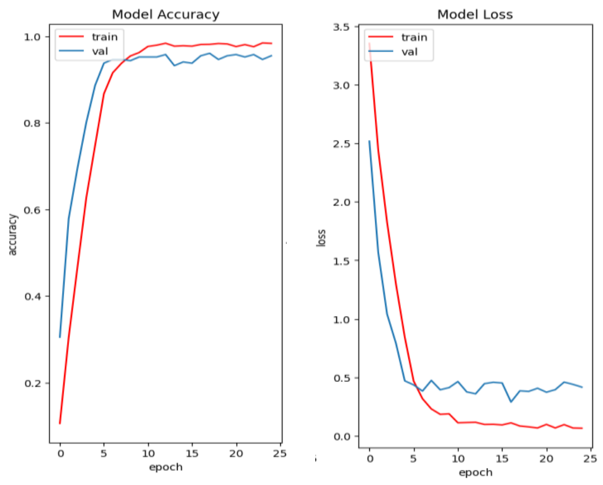
Figure 4 shows the model of model accuracy (Figure 4(a)) and model loss according (Figure 4(b)) to train and validation data according to batch size=32, a number of epochs 0 to 25, and the optimization algorithm are Adam. In Figure 4, the pattern closely mirrors that of Figure 3. When the epoch value reaches 15 to 25, the model accuracy (Figure 4(a)) exhibits a consistent increase and then stabilizes. Similarly, the model loss (Figure 4(b)) demonstrates a decreasing trend and eventually stabilizes within the same epoch range.

Figure 5 shows the model of model accuracy (Figure 5(a)) and model loss according (Figure 5(b)) to train and validation data according to batch size=64, a number of epochs 0 to 25, and the optimization algorithm are Adam. In Figure 5, the pattern closely resembles that of Figures 3 and 4. As the epoch value nears 15 to 25, the accuracy model (Figure 5(a)) displays a steady increase followed by stability. Similarly, in the model loss (training loss) (Figure 5(b)), there's a decreasing trend, eventually stabilizing between the 15 and 25 epochs.



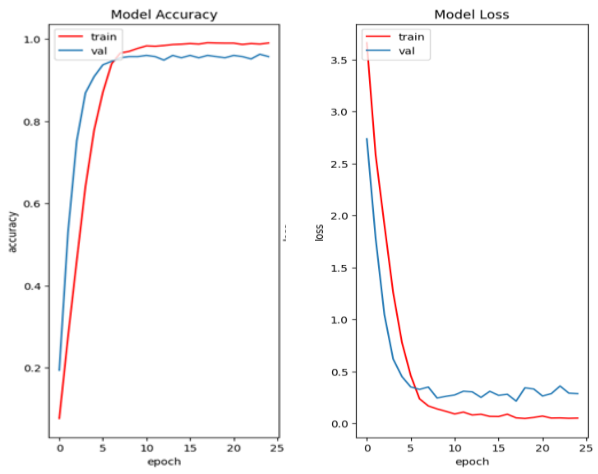
(a) (b)

Figure 3. Model of (a) model accuracy and (b) model loss according to train and validation data according to batch size=16, number of epochs 0 to 25, and the optimization algorithm is Adam



(a) (b)

Figure 4. Model of (a) model accuracy and (b) model loss according to train and validation data according to batch size=32, number of epochs 0 to 25, and the optimization algorithm is Adam



(a) (b)

Figure 5. Model of (a) model accuracy and (b) model loss according to train and validation data according to batch size=64, number of epochs 0 to 25, and the optimization algorithm is Adam

### 3.3. Comparison of accuracy based on different batch sizes at a number of epochs 10

Based on Table 1, increasing the batch size from 16 to 64 resulted in an increase in accuracy (both training and validation) of 2.90 and 0.86%, respectively. This demonstrates that the model's accuracy is directly proportional to the increase in batch size and inversely proportional to the training loss. Consequently, the model becomes more accurate as the batch size increases. However, it's notable that the average time taken per step or batch during training (ms/step) also increases with the batch size.

Based on Table 2, the same pattern as Table 1 is observed. Increasing the batch size from 16 to 64 resulted in an increase in accuracy (both training and validation) of 1.74 and 0.57%, respectively. This increase in accuracy indicates a more accurate model with a larger batch size. Additionally, it's noteworthy that the average time taken per step or batch during training decreased from batch size 16 to 64.

In Table 3, it also shows the same pattern as Tables 1 and 2. The accuracy value increases (both training and validation) by 0.93 and 0.86%, when the batch size increases from 16 to 64. Meanwhile, the average time taken per step or batch during training reduced from batch size 16 to 64.

Based on the results, the accuracy improves with an increase in batch size (resulting in more data) and the number of epochs (allowing the model more opportunities to learn from the data, leading to better convergence and reduced underfitting). The results illustrate the highest accuracy obtained, when tuning parameters for batch size is 64 and the number of epochs is 25 (Table 3). This is because a larger batch size and more epochs provide a more stable and accurate estimate of the gradient of the loss function, reducing noise and fluctuations in the learning process, which ultimately leads to a more accurate proposed model. Therefore, it can produce better generalization to unseen data, as the model can avoid overfitting by training on more data points in each iteration. This approach can enhance the accuracy of validation results. Additionally, larger batch sizes and more epochs benefit modern graphics processing units (GPUs) or tensor processing units (TPUs), allowing for faster training while maintaining or even improving model accuracy.

Table 1. Comparison of accuracy and loss using different batch sizes for a number of epochs=10

| Batch Size | Num of epochs | Training accuracy (%) | Training loss (%) | Validation accuracy (%) | Validation loss (%) | Average time taken per step or batch during training (ms/step) |
|---|---|---|---|---|---|---|
| 16 | 10 | 0.9466 | 0.2087 | 0.9487 | 0.4833 | 216 |
| 32 | 10 | 0.9593 | 0.1768 | 0.9516 | 0.4127 | 399 |
| 64 | 10 | 0.9740 | 0.1192 | 0.9573 | 0.2607 | 731 |

Table 2. Comparison of accuracy based on different batch sizes for a number of epochs=15

| Batch Size | Num of Epochs | Training accuracy (%) | Training loss (%) | Validation accuracy (%) | Validation loss (%) | Average time taken per step or batch during training (ms/step) |
|---|---|---|---|---|---|---|
| 16 | 15 | 0.9693 | 0.1262 | 0.9487 | 0.5391 | 212 |
| 32 | 15 | 0.9837 | 0.0840 | 0.9402 | 0.4587 | 396 |
| 64 | 15 | 0.9861 | 0.0679 | 0.9544 | 0.3091 | 743 |

Table 3. Comparison of accuracy based on different batch sizes for a number of epochs=25

| Batch Size | Num of Epochs | Training accuracy (%) | Training loss (%) | Validation accuracy (%) | Validation loss (%) | Average time taken per step or batch during training (ms/step) |
|---|---|---|---|---|---|---|
| 16 | 25 | 0.9811 | 0.0675 | 0.9544 | 0.4752 | 216 |
| 32 | 25 | 0.9845 | 0.0655 | 0.9548 | 0.4173 | 401 |
| 64 | 25 | 0.9902 | 0.0606 | 0.9630 | 0.2869 | 636 |

### 3.4. Testing data for the proposed model

Based on the results from subsection 3.3, the proposed model was trained using 25 epochs and batch sizes of 16, 32, and 64. The accuracy of the training dataset for the proposed model is presented, with examples from the apple and cabbage classes and the result displayed in Table 4. According to the example of this class, it was found that the average accuracy of the training dataset increases when the batch size and number of epochs increase. It shows increasing the batch size and number of epochs leads to higher accuracy on the testing dataset, it indicates that the chosen batch size and number of epochs are conducive to better generalization and overall model performance.

Table 4. Average accuracy of the testing data for class apple and watermelon

| Average accuracy of training (%) of 25 numbers of epochs | | |
|---|---|---|
| | Class | |
| Batch size | Apple | Watermelon |
| 16 | 93.8300 | 89.8883 |
| 32 | 96.2550 | 97.0500 |
| 64 | 99.5700 | 99.7600 |

## 4. CONCLUSION

This work has developed to model fruit and vegetables using fine-tuning hyperparameters of CNN. From the experiments, a proposed model with a combination of the batch size=64 and the number of epochs=25, produces the most optimal model with an accuracy value (training) of 99.02%, while the validation is 95.73% and the loss is 6.06% (minimum). It can be concluded that an increase in batch size and number of epochs is directly proportional to accuracy for both training and validation and inversely proportional to training and validation loss and this relationship leads to more accurate model predictions. Considering this result, which involves the relationship between batch size and the number of epochs, there is an opportunity for researchers to explore further studies on the fine-tuning relationships between other hyperparameters to produce a more accurate prediction model.

**FUNDING INFORMATION**

This project is funded by the Center for Research Excellence, Incubation Management Center, University Sultan Zainal Abidin, Malaysia. Under Project Code is UniSZA/2023/DPU 1.0/05.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Abd Rasid Mamat | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  | ✓ | ✓ | ✓ |  | ✓ | ✓ | ✓ |
| Mohamad Afendee Mohamed | ✓ |  | ✓ | ✓ |  |  | ✓ |  |  | ✓ | ✓ |  |  |  |
| Mohd Fadzil Abd Kadir |  | ✓ | ✓ | ✓ |  |  | ✓ |  |  | ✓ | ✓ |  |  |  |
| Norkhairani Abdul Rawi |  |  |  |  | ✓ |  | ✓ |  |  | ✓ |  |  | ✓ |  |
| Azim Zaliha Abd Aziz | ✓ |  |  |  | ✓ | ✓ | ✓ |  |  | ✓ |  |  |  |  |
| Wan Suryani Wan Awang |  | ✓ |  |  |  | ✓ | ✓ | ✓ | ✓ |  | ✓ |  | ✓ |  |

| | | |
|---|---|---|
| C : **C**onceptualization | I : **I**nvestigation | Vi : **Vi**sualization |
| M : **M**ethodology | R : **R**esources | Su : **Su**pervision |
| So : **So**ftware | D : **D**ata Curation | P : **P**roject administration |
| Va : **Va**lidation | O : Writing - **O**riginal Draft | Fu : **Fu**nding acquisition |
| Fo : **Fo**rmal analysis | E : Writing - Review & **E**diting | |

## CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## DATA AVAILABILITY

The dataset used to work be fruit_vegetable's dataset and which is publicly available on Kaggle which is publicly available. It free and download from https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition.

## REFERENCES

[1] M. Wu, J. Zhou, Y. Peng, S. Wang, and Y. Zhang, "Deep learning for image classification: a review," in *Proceedings of 2023 International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2023)*, Springer, Singapore, pp. 352–362, 2024, doi: 10.1007/978-981-97-1335-6_31.

[2] P. T. Q. Anh, D. Q. Thuyet, and Y. Kobayashi, "Image classification of root-trimmed garlic using multi-label and multi-class classification with deep convolutional neural network," *Postharvest Biology and Technology*, vol. 190, Aug. 2022, doi: 10.1016/j.postharvbio.2022.111956.

[3] A. Z. A. Aziz, M. S. M. Sabilan, and N. F. Mansor, "The effectiveness of using depth images for gait-based gender classification," *IAENG International Journal of Computer Science*, vol. 50, no. 3, 2023, [Online]. Available: https://www.iaeng.org/IJCS/issues_v50/issue_3/IJCS_50_3_03.pdf

[4] A. Carbone, "Cancer classification at the crossroads," *Cancers*, vol. 12, no. 4, Apr. 2020, doi: 10.3390/cancers12040980.

[5] M. Zubair, S. Wang, and N. Ali, "Advanced approaches to breast cancer classification and diagnosis," *Frontiers in Pharmacology*, vol. 11, Feb. 2021, doi: 10.3389/fphar.2020.632079.

[6] C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, "Recent advancements in fruit detection and classification using deep learning techniques," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–29, Jan. 2022, doi: 10.1155/2022/9210947.

[7] R. Butuner, I. Cinar, Y. S. Taspinar, R. Kursun, M. H. Calp, and M. Koklu, "Classification of deep image features of lentil varieties with machine learning techniques," *European Food Research and Technology*, vol. 249, no. 5, pp. 1303–1316, May 2023, doi: 10.1007/s00217-023-04214-z.

[8] N.-E.-A. Mimma, S. Ahmed, T. Rahman, and R. Khan, "Fruits classification and detection application using deep learning," *Scientific Programming*, vol. 2022, pp. 1–16, Nov. 2022, doi: 10.1155/2022/4194874.

[9] R. Mahum *et al.*, "A novel framework for potato leaf disease detection using an efficient deep learning model," *Human and Ecological Risk Assessment: An International Journal*, vol. 29, no. 2, pp. 303–326, Feb. 2023, doi: 10.1080/10807039.2022.2064814.

[10] S. I. Juma, Z. Muda, M. A. Mohamed, and W. Mohamed, "Machine learning techniques for intrusion detection system: a review," *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 3, pp. 422–429, 2015.

[11] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[12] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, "Exploring the frontiers of deep learning and natural language processing: a comprehensive overview of key challenges and emerging trends," *Natural Language Processing Journal*, vol. 4, Sep. 2023, doi: 10.1016/j.nlp.2023.100026.

[13] H. Herman, A. Susanto, T. W. Cenggoro, S. Suharjito, and B. Pardamean, "Oil palm fruit image ripeness classification with computer vision using deep learning and visual attention," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 12, no. 2, pp. 21–27, 2020, [Online]. Available: https://jtec.utem.edu.my/jtec/article/view/5543

[14] Suharjito, G. N. Elwirehardja, and J. S. Prayoga, "Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches," *Computers and Electronics in Agriculture*, vol. 188, 2021, doi: 10.1016/j.compag.2021.106359.

[15] S. Saifullah, D. B. Prasetyo, Indahyani, R. Drezewski, and F. A. Dwiyanto, “Palm oil maturity classification using k-nearest neighbors based on RGB and l*a*b color extraction,” *Procedia Computer Science*, vol. 225, pp. 3011–3020, 2023, doi: 10.1016/j.procs.2023.10.294.

[16] H. Altaheri, M. Alsulaiman, and G. Muhammad, “Date fruit classification for robotic harvesting in a natural environment using deep learning,” *IEEE Access*, vol. 7, pp. 117115–117133, 2019, doi: 10.1109/ACCESS.2019.2936536.

[17] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, “Automatic fruit classification using deep learning for industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, Feb. 2019, doi: 10.1109/TII.2018.2875149.

[18] J. Steinbrener, K. Posch, and R. Leitner, “Hyperspectral fruit and vegetable classification using convolutional neural networks,” *Computers and Electronics in Agriculture*, vol. 162, pp. 364–372, Jul. 2019, doi: 10.1016/j.compag.2019.04.019.

[19] S. Fan *et al.*, “On line detection of defective apples using computer vision system combined with deep learning methods,” *Journal of Food Engineering*, vol. 286, Dec. 2020, doi: 10.1016/j.jfoodeng.2020.110102.

[20] M. Sewak, S. K. Sahay, and H. Rathore, “An overview of deep learning architecture of deep neural networks and autoencoders,” *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 1, pp. 182–188, Jan. 2020, doi: 10.1166/jctn.2020.8648.

[21] S. S., J. I. Zong Chen, and S. Shakya, “Survey on neural network architectures with deep learning,” *Journal of Soft Computing Paradigm*, vol. 2, no. 3, pp. 186–194, Jul. 2020, doi: 10.36548/jscp.2020.3.007.

[22] N. Rai and X. Sun, “WeedVision: a single-stage deep learning architecture to perform weed detection and segmentation using drone-acquired images,” *Computers and Electronics in Agriculture*, vol. 219, Apr. 2024, doi: 10.1016/j.compag.2024.108792.

[23] B. Bischl *et al.*, “Hyperparameter optimization: foundations, algorithms, best practices, and open challenges,” *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, Mar. 2023, doi: 10.1002/widm.1484.

[24] A. R. Mamat, F. S. Mohamad, N. abdul Rawi, M. K. Awang, M. I. Awang, and M. F. A. Kadir, “Region based image retrieval based on texture features,” *Journal of Theoretical and Applied Information Technology*, vol. 92, no. 1, 2016.

[25] K. Wangchuk, T. Dorji, P. Ram Dhungyel, and P. Galey, “Fruits and vegetables recognition system in Dzongkha using visual geometry group network,” *Zorig Melong- A Technical Journal of Science, Engineering and Technology*, vol. 8, no. 1, Aug. 2023, doi: 10.17102/v8004.

[26] M. Mukhiddinov, A. Muminov, and J. Cho, “Improved classification approach for fruits and vegetables freshness based on deep learning,” *Sensors*, vol. 22, no. 21, Oct. 2022, doi: 10.3390/s22218192.

[27] C.-H. Chang, L. Rampasek, and A. Goldenberg, “Dropout feature ranking for deep learning models,” *arXiv*, Dec. 2017, [Online]. Available: http://arxiv.org/abs/1712.08645

[28] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” *arXiv*, Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[29] A. F. Agarap, “Deep learning using rectified linear units (ReLU),” *arXiv*, Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.08375

[30] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, “Understanding deep neural networks with rectified linear units,” *arXiv*, Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.01491

[31] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, “Activation functions in deep learning: a comprehensive survey and benchmark,” *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022, doi: 10.1016/j.neucom.2022.06.111.

[32] C. Garbin, X. Zhu, and O. Marques, “Dropout vs. batch normalization: an empirical study of their impact to deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 19–20, pp. 12777–12815, May 2020, doi: 10.1007/s11042-019-08453-9.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014, [Online]. Available: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

[34] S. K. Devalla *et al.*, “A deep learning approach to digitally stain optical coherence tomography images of the optic nerve head,” *Investigative Opthalmology & Visual Science*, vol. 59, no. 1, Jan. 2018, doi: 10.1167/iovs.17-22617.

[35] M. G. Aliyu, M. F. A. Kadir, A. R. Mamat, and M. Mohamad, “Noise removal using statistical operators for efficient leaf identification,” *International Journal of Computer Aided Engineering and Technology*, vol. 10, no. 4, 2018, doi: 10.1504/IJCAET.2018.092834.

[36] Ò. Lorente, I. Riera, and A. Rana, “Image classification with classic and deep learning techniques,” *arXiv*, May 2021, [Online]. Available: http://arxiv.org/abs/2105.04895

[37] M. Rizon, N. A. Najihah Yusri, M. F. Abdul Kadir, A. R. bin Mamat, A. Z. Abd Aziz, and K. Nanaa, “Determination of mango fruit from binary image using randomized hough transform,” in *Eighth International Conference on Machine Vision (ICMV 2015)*, A. Verikas, P. Radeva, and D. Nikolaev, Eds., Dec. 2015, doi: 10.1117/12.2228511.

[38] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A comparative study of fine-tuning deep learning models for plant disease identification,” *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[39] S. Chetlur *et al.*, “CuDNN: efficient primitives for deep learning,” *arXiv*, Oct. 2014, [Online]. Available: http://arxiv.org/abs/1410.0759

## BIOGRAPHIES OF AUTHORS

**Abd Rasid Mamat** received Ph.D in Computer Science at Universiti Sultan Zainal Abidin in 2022, where he currently serves as a Senior Lecturer. His main research areas include image processing, content-based image retrieval, decision support systems, and deep learning. He can be contacted at email: arm@unisza.edu.my.

**Mohamad Afendee Mohamed** received his Ph.D in Mathematical Cryptography at Universiti Putra Malaysia in 2011. Upon completion, he served the university for three years as a senior lecturer. In 2014, he moved to Universiti Sultan Zainal Abidin and later assumed an associate professor position. His current research interests include both theoretical and application issues in the domain of data security and mobile and wireless networking. Dr Mohamed has authored more than 100 articles that have appeared in various journals, book chapters, and conference proceedings. He can be contacted at email: mafendee@unisza.edu.my.

**Mohd Fadzil Abdul Kadir** received Ph.D. in Engineering (System Engineering) at the Mie University, Mie, Japan, in 2012. Since 2006, he has been with the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, where he is currently a Senior Lecturer. His main areas of research interest are digital image processing, pattern recognition, information security, and cryptography. He is also a member of the Malaysia Board of Technologists. He can be contacted at email: fadzil@unisza.edu.my.

**Norkhairani Abdul Rawi** is a senior lecturer in the Department of Multimedia Studies, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin. She received her master’s degree in information science in 2003. Her research interests are multimedia and image processing. She can be contacted at email: khairani@unisza.edu.my.

**Azim Zaliha Abd Aziz** is a senior lecturer at the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Terengganu, Malaysia. She received her PhD (Computer Science) from the University of Reading, United Kingdom in 2017. She did her first degree in Computer Science in 2000 at the Faculty of Computer Science, Universiti Teknologi Malaysia. She can be contacted at email: azimzaliha@unisza.edu.my.

**Wan Suryani Wan Awang** is a senior lecturer in Computer Science at Universiti Sultan Zainal Abidin in Kuala Terengganu, Malaysia, where she has worked since 2007. She earned her undergraduate degree in computer studies from Sheffield Hallam University in the UK, a master’s degree from what is now Universiti Malaysia Terengganu, and a Ph.D. in Computer Science from Cardiff University. Her expertise in distributed systems and databases allows her to offer valuable insights in both academic and professional settings. She can be contacted at email: suryani@unisza.ed.my.