Journal of Applied Data Sciences
Vol. 6, No. 1, January 2025, pp. 101-114

Convolutional Neural Network Based Deep Learning Model for Accurate Classification of Durian Types

Diana Diana[1,*], Tri Basuki Kurniawan[2], Deshinta Arrova Dewi[3], Mashal Kasem Alqudah[4], Musab Kasim Alqudah[5]
Mohd Zaki Zakari[6], Eyna Fahera Binti Eddie Fuad[7]

[1]*Faculty of Science Technology, Universitas Bina Darma, Palembang, Indonesia*

[2]*Magister Program, Universitas Bina Darma, Palembang, Indonesia*

[3]*Faculty of Data Science and Information Technology, Inti International University, Malaysia*

[4]*Institute of Public Administration, Digital Transformation and Information Programs Department, Riyadh, Saudi Arabia*

[5]*College of Science and Computer Engineering, Yanbu, Taibah University, Yanbu, Al-Madinah Al-Munawwarah, Saudi Arabia*

[6,7]*Faculty of Computer and Mathematics Sciences, University Technology Mara, Malaysia*

(Received: June 10, 2024; Revised: September 20, 2024; Accepted: October 17, 2024; Available online: December 27, 2024)

Abstract

Durian recognition is significant among fans of the durian community since many people tend to get confused, especially if they are not familiar with durian species, which can lead them to be involved in durian fraud. The development of this prototype can detect and classify durian fruits into three categories, including Musang King, Black Thorn, and D24, which can significantly benefit consumers. The prototype in this research involves training using a dataset of durian images, specifically in Musang King, Black Thorn, and D24 varieties. Preprocessing techniques such as resizing and scaling data are applied to enhance the quality and consistency of the dataset. The models chosen to develop this prototype include VGG-16 and Xception, and each model is compared according to its accuracy percentage. The accuracy outcomes of VGG-16 and Xception models are 56.64% and 92%, respectively. The models used a total of 1,372 images of durian with three classifications. Based on the findings, further enhancement of the CNN models for durian classification can be done by implementing different architectures, techniques, and methods. Moreover, future models can consider real-time image capture and processing capabilities to enhance the practicality of the system for durian consumers. The prototype developed in this study demonstrates the feasibility of using deep learning techniques for accurate and efficient durian classification, paving the way for future advancements in automated fruit grading and quality control systems in the durian industry.

Keywords: Durian Recognition, Deep Learning, Image Classification, Product Innovation, Process Innovation

1. Introduction

Fruits are widely grown in multiple weather conditions, depending on their needs. Some fruits, called seasonal fruits, can only be grown under certain conditions and weather. Durians are one of the examples of seasonal fruits [1] that are greatly loved locally and globally. Durian is known as the ‘King of Fruits’ due to its strong smell [2] and unique flavor. Fruits are due to their pungent smell [2] and exceptional flavor. Durian is a tropical fruit that originated and is widely grown in Southeast Asia, including Thailand, Indonesia, the Philippines, and our very own country, Malaysia [3]. The leading international exporters of durian are Malaysia, Thailand, and Indonesia.

The value of production for durians in 2021 was RM8.46 million, with a hectare of 48.2% of fruit hectarage in Malaysia (Department of Agriculture Malaysia [DOA]) [3]. In recent years, durian’s popularity has increased globally [4] and resulted in an increment in commercial production and exports. Durian has numerous benefits, including high levels of vitamins B and C, potassium, magnesium, iron, and antioxidants [5]. Furthermore, it is rich in natural sugars [6], which can become a source of energy. It also contains high fiber [6], which can prevent constipation and help with

***Corresponding author: Diana Diana (diana@binadarma.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i1.480

digestion. However, durian is also high in fat and calories [6]. Durian business has become a massive thing over the past few years, and some types of durians are pricier, such as D24 and D197, also known as 'Musang King.' People are willing to spend their money every season to get their hands on their favorite durian [7]. Due to its popularity, there has been an increased number of individuals starting with durian businesses either to gain side income or to be their primary source of income. However, issues may come with the company, especially towards the consumers, such as market exploitations [2] for high demand in durians.

In some cases, irresponsible individuals try to provide consumers with incorrect information about durian [8] since the consumers may not be familiar with and lack knowledge of durian types. This fraud issue could happen not only to locals but also to tourists, which can lead to a wrong impression of the country. D197, widely known as Musang King, is one of the most famous durian types and one of the most expensive ones [9], and there will always be a topic of durian fraud among consumers. Consumers would come to durian stalls and pay a sum to get their hands on Musang King but would end up with cheaper durian types without them even knowing.

This project's scope is to have durian recognition and only focus on three types of durian species: Musang King, Black Thorn, and D24. The prototype used a deep learning approach for durian recognition called convolutional neural network (CNN) models, specifically VGG-16 and Xception. It focuses on the visual appearance of the durian if the durian fulfills the specification of the durian's variation. Hence, it only depends on the outside features and not the inside part of the durian. This project aims to recognize the three species of durians successfully and accurately.

The prototype only focuses on three types of durian species, Musang King, Black Thorn, and D24, because too many types or variations of durian exist to be covered. Furthermore, the dataset used in this project needed to be bigger, approximately 19 Gigabytes, containing ten types of durian species. It would be time-consuming if all classes from the dataset used for the project went through all steps in the model development process.

By developing the prototype, consumers may avoid getting scammed or receiving incorrect types of durians from durian sellers. The prototype can also help ensure that the quality control of durians sold to consumers is based on their desired type rather than the other way around. Furthermore, it helps to increase the transparency of the durian's origin to allow consumers to be informed about their purchases.

2. Method

2.1. Image Recognition

Image detection and recognition are vast and vital areas of practicality in real-world applications. Image recognition in fruits is one of the topics discussed in this study. Numerous recent studies have addressed various challenges related to fruit detection and recognition, including fruit recognition, feature detection, and fruit segmentation. According to [10], object classification involves creating a classification model using available data, enabling the mapping of data in a database to specific categories for predictive purposes. A powerful classification method is the support vector machine (SVM), derived from statistical learning and utilizes a supervised learning model during training. The primary learning strategy of SVM aims to maximize the margin, a concept that can be mathematically formulated as a convex quadratic programming problem.

2.2. Fruit Recognition

Agriculture is a sector that contributes many benefits to certain countries, but it has its specific constraints [11]. Research finds that there has been an incredible increase in the variation of fruits [12], and upgraded types are being bred intensively to get more variations and options [13]. This has resulted in peaks in businesses with specific seasonal conditions [13]. However, these conditions are unpredictable and can only be predicted on a short-term basis of seasonal outcome since it depends on the weather conditions and the labor market. Fruit recognition's precision can be limited and constantly influenced by conditions like the amount of light, size of coverage area, visual angles, scale variation, and environmental changes [14]. However, fruit recognition can reduce errors caused by manual observation and fraud in selling and buying durian.

2.3. Convolutional Neural Network

The convolutional neural network (CNN) is one of the classes of neural network architectures that resembles a conventional feed-forward neural network. CNN imitates the visual cortex of the human brain [15]. A neural network of convolutional layers consists of filters that will convolve across the input according to the specific filter size and stride distance [15]. Developing a reliable and accurately built system in fruit detection uses deep neural convolutional networks, including fruit detection and fruit classification. Faster Regional CNN (Faster R-CNN) has been a development of a cutting-edge item detector for the past few years [16].

2.4. VGG-16

Simonyan and Zisserman introduced the Visual Geometry Group (VGG) network architecture, known for its straightforward design. It consists of multiple 3 × 3 convolutional layers stacked on each other, gradually increasing in depth. Max pooling is employed to handle volume size reduction. Following these layers, two fully connected layers with 4,096 nodes each are used, and a SoftMax classifier is applied. However, deploying VGG can be challenging due to its substantial size. Specifically, VGG16 requires over 533 MB of storage, while VGG19 demands 574 MB. Despite the deployment complexities, VGG remains widely utilized in various deep learning image classification tasks, primarily because of its simplicity [17].

2.5. Xception

Xception, short for "Extreme Inception," is an architecture proposed by Google that contains the same number of parameters as Inception V3. The key to its improved performance is its efficient parameter utilization and increased capacity. Unlike the Inception architecture, Xception separates the cross-channel and spatial correlation mappings in its output maps. This separation was introduced in the Xception architecture to enhance its capabilities. The network employs 36 convolutional layers for feature extraction, organized into 14 modules, each surrounded by linear residual connections. However, the first and last modules need to include these specific representations. Additionally, in the final fully connected layer, the number of classes has been reduced to 6 [18].

2.6. ResNet

The ResNet architecture has significantly contributed by showcasing the feasibility of training intense networks using standard stochastic gradient descent (SGD) and an appropriate initialization technique. The key innovation lies in the incorporation of residual modules. In a subsequent publication, the authors introduced identity mappings to enhance accuracy. Despite its increased depth compared to VGG, ResNet achieves a more petite model size by utilizing global average pooling instead of fully connected layers. This efficient approach reduces the model size to 102 MB for ResNet50 [19]. The research framework for developing this project's prototype had eight main phases. The first phase was Preliminary Research, followed by Knowledge Acquisition, Data Acquisition, Data Preprocessing, Model Development, Model Evaluation, Prototype Development, and Prototype Testing, as shown in figure 1.

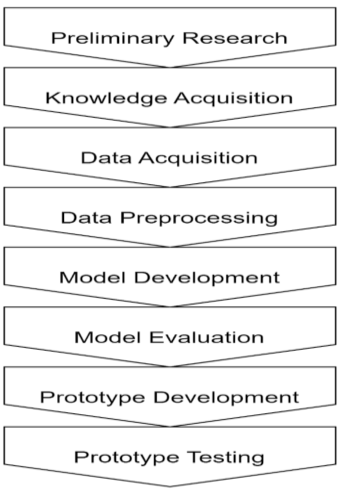

Figure 1. Research framework

The first objective of this project is to determine the features of fruits for image recognition in durian. The approach needed to achieve this objective was doing preliminary research and studying in detail to understand the problem with the fruit's characteristics or visual appearance. The initial research started with reading journals and articles on online platforms, mainly Google Scholar and IEEE. It was also essential to have all articles and journals that are being referred to be from reliable sources. The deliverables from this preliminary research were the problem statement, research questions, research objectives, scope, constraint, and significance.

The second approach for the first objective was to collect knowledge from previous research through knowledge acquisition. This process identifies durian features needed for image recognition using artificial intelligence methods. Knowledge acquisition is mainly done to gain intense knowledge about the domain and scope of this project. The outcome of this project is a literature review analysis.

The second objective of this project is to design a model that can be used for fruit recognition based on its features. To complete this objective, it is necessary to begin the approaches with data collection, where this phase is to identify an open-source dataset for fruit images. It was obtained through the open-source platform Kaggle since it is the most accessible platform for getting a secondary dataset of fruit images. The reason for data collection is to train the pictures so that the system can detect and recognize the fruit.

We used the dataset from Eden Barua on Kaggle [20] at https://www.kaggle.com/datasets/edenbarua/picture/data. It has 3611 durian images for training and testing, with ten different durian species: 101, Black Thorn, D13, D2, D24, D88, Golden Phoenix, KH, Kucing Tidur, and Musang King [21]. The only species of durian used in this project are Black Thorn, D24, and Musang King, as shown in figure 2.

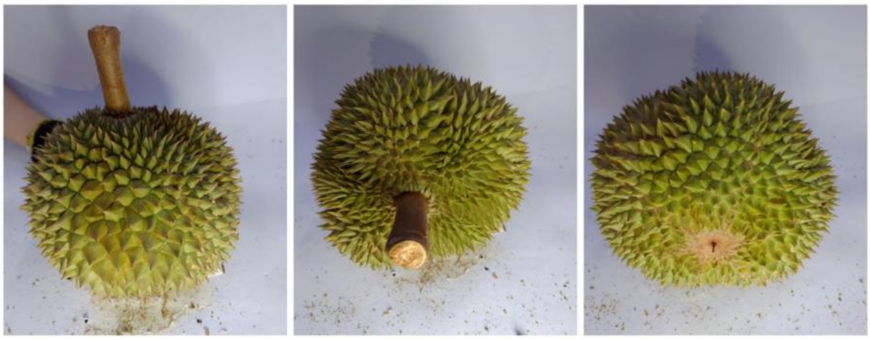

Figure 2. Sample images of Black Thorn

Data preprocessing is an essential step in machine learning and deep learning tasks. It involves transforming raw data into a format suitable and efficient for training a model. Two models were developed for this project: the VGG-16 model and the Xception model. After choosing the appropriate dataset for this project, the images must undergo a preprocessing process before the dataset is ready to train.

Data preprocessing includes splitting the dataset into training, validation, and testing sets. The training set is used to train the model, the validation set is used to tune hyperparameters, and the testing set is used to evaluate the model's performance on unseen data. In this project, datasets were separated into a 70:20:10 ratio, which means 70% of the dataset was used for training, 20% of the dataset was used for validation, and 10% of the dataset was used for testing. For deep learning tasks, data augmentation is a form of preprocessing where the dataset is artificially expanded by applying various transformations (e.g., rotations, flips) to the existing data. This helps to increase the diversity and generalization of the model.

The first activity in model evaluation is parameter tuning. Parameters are settings that govern the model's behavior during training and inference. It includes learning rate, batch size, number of epochs, and dropout rate. Properly tuning these parameters can significantly impact the model's accuracy and generalization ability. Throughout the parameter tuning process, we repeatedly train the model with different combinations of hyperparameters and evaluate its performance on the validation set. Metrics such as accuracy, precision, recall, and F1 score assess how well the model performs for different parameter choices. We might also monitor the training progress to identify overfitting or underfitting issues.

The main deliverable of this model evaluation process is obtaining the best model configuration. After trying various parameter combinations and evaluating their results, the set of hyperparameters that led to the highest overall performance on the validation set was selected. This "best model" is the one that generalizes well to unseen data and is expected to perform well in real-world scenarios.

This phase's deliverable is an operational prototype with which users can access and interact. The prototype will recognize and classify durian images into three types: Musang King, D24, or Black Thorn. Users will upload their durian images through the user-friendly Gradio interface and receive instant predictions. This prototype is a significant step towards a fully functioning and deployable durian recognition project.

3. Results and Discussion

Only three classes were used in this project: Black Thorn, D24, and Musang King. The next step in model development is using the prepared dataset to train the model for durian classification. In this phase, multiple model tests are done to see which model works best for the project. Different models have different parameters that need tuning to get the highest accuracy. In this project, two other models, VGG-16 and Xception, are used for multi-classification, as shown in table 1.

Table 1. The dataset used in this project

Durian Classes	
Durian Type	**Number of Images**
Black Thorn	71
D24	529
Musang King	772

For the first experiment, VGG-16 was used as the architecture. The input dataset was separated into a 70:20:10 ratio, which means 70% of the input is separated to the train path, 20% will go to the validation path, and the rest will go to the testing path. The preprocessing method uses data augmentation to get more input before going for model training. The data augmentation for the train path includes rescaling, rotation, horizontal and vertical shift, shear transformation, random zoom, and horizontal flip. On the other hand, the valid path only focuses on the rescale. After the data augmentation process, the total images found in train, validation, and test are 1252 images, 494 images, and 261 images, respectively, as shown in figure 3.

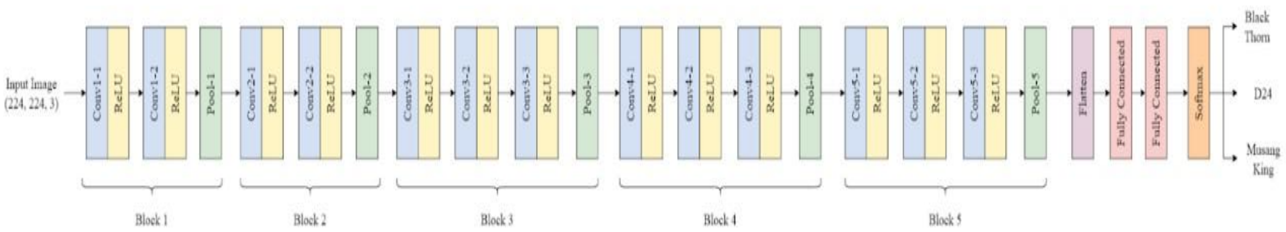

Figure 3. Model Architecture for Experiment 1

The model architecture shows that it is a CNN model using TensorFlow Keras functional API. The VGG architecture is well-known for its simplicity and effectiveness in image classification tasks. This model consists of five convolutional blocks, each comprising multiple convolutional and pooling layers, followed by two fully connected (dense) layers and an output layer. Once the architecture is defined, the model is compiled with the 'Adam' optimizer, a popular optimization algorithm known for its adaptivity to different datasets. The categorical cross-entropy loss function is chosen since this is a multi-class classification problem. Additionally, accuracy is used as the evaluation metric to monitor the model's performance during training.

3.1. Experiment 1

This VGG-16 model is designed to classify images into one of three durian classes and is ready for training and evaluation on a suitable dataset. Its effectiveness in capturing image features is attributed to combining convolutional layers with max pooling, which enables the network to learn hierarchical representations and achieve higher accuracy in image classification tasks. After developing the model architecture, the model is ready to undergo training.

Table 2. Model training results

Model Train Result				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	1.0604	0.5172	0.8481	0.5562
2	0.8641	0.5615	0.8565	0.5562
3	0.8566	0.5623	0.8521	0.5583
⋮	⋮	⋮	⋮	⋮
18	0.8510	0.5639	0.8586	0.5521
19	0.8500	0.5623	0.8472	0.5542
20	0.8474	0.5590	0.8426	0.5604

Table 2 shows several patterns. The training loss decreases, and the training accuracy increases across epochs. This suggests that the model effectively learns from the training data and minimizes the discrepancies between predictions and proper labels.

Meanwhile, the validation loss and accuracy show fluctuations over epochs. This behavior is expected and indicates that the model's performance on the validation data is sensitive to the variations in the dataset and model updates during training. However, the small gap between training and validation performance indicates that the model is moderately balanced. The overall performance of this model needs to be better, considering the accuracy over 20 epochs was only 55.9%, as shown in figure 4, figure 5, and figure 6.

Figure 4. Model accuracy for Experiment 1

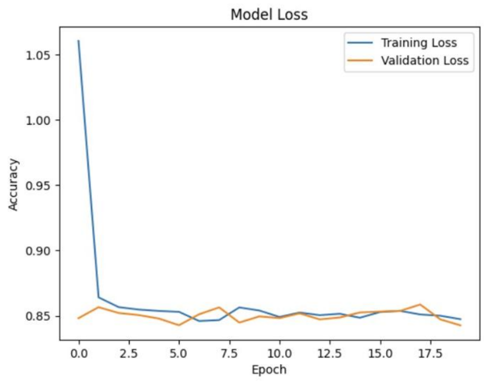

Figure 5. Model loss for Experiment 1

```
8/8 [==============================] - 43s 5s/step - loss: 0.8602 - accuracy: 0.5664
Test Loss: 0.8602184653282166
Test Accuracy: 0.56640625
```

Figure 6. Test loss and test accuracy results

The plot indicates how well the model performed on a separate test data set, which was not seen during training. The test loss is around 0.8602, representing the model's error rate on this unseen data. The test accuracy is approximately 56.64%, showing the percentage of the test data the model could classify correctly. This relatively low accuracy suggests that while the model can learn to fit the training data, it needs help to apply this learning effectively to new, unseen scenarios.

3.2. Experiment 2

The proposed model is an Xception model, which is short for "Extreme Inception." This model shares the exact parameter count as Inception V3, yet it achieves enhanced performance thanks to its utilization of parameters and expanded capacity. Unlike the inception architecture, Xception separates the output maps' cross-channel and spatial correlation mappings. In this experiment, three parameters were tuned: learning rate, dense layers, and dropout rate. This second experiment starts by testing three learning rates to determine which has the best outcome. The learning rates for this experiment are 0.001, 0.01, and 0.1. Table 3, table 4, and table 5 below shows the results for the following learning rates.

Table 3. The learning rate of 0.001

Learning rate for 0.001				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.7348	0.6946	0.8291	0.6073
2	0.3058	0.8796	0.7576	0.6400
3	0.2222	0.9161	0.6531	0.6909
⋮	⋮	⋮	⋮	⋮
18	0.0450	0.9915	0.2393	0.9164
19	0.0538	0.9842	0.2323	0.9309
20	0.0434	0.9878	0.2285	0.9382

Table 4. The learning rate of 0.01

Learning rate for 0.01				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.9881	0.7908	1.2924	0.6982
2	0.2932	0.9282	0.7782	0.8327
3	0.2380	0.9355	0.8427	0.8400
⋮	⋮	⋮	⋮	⋮
18	0.0869	0.9793	0.9835	0.8909
19	0.0869	0.9891	0.6778	0.9091
20	0.0125	0.9951	0.7832	0.9164

Table 5. The learning rate of 0.1

Learning rate for 0.1				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	14.2277	0.7603	9.3835	0.7745
2	4.9759	0.8905	11.4745	0.6327
3	5.2647	0.8966	11.8164	0.7564
⋮	⋮	⋮	⋮	⋮
18	2.7341	0.9611	21.7105	0.8182
19	2.0921	0.9611	13.8053	0.9055
20	4.6942	0.9453	18.4726	0.8873

Figure 7 illustrates the learning curves for different learning rates during Experiment 2, showing the progression of training and validation accuracy over epochs. For a learning rate of 0.001, the training accuracy increases gradually and stabilizes close to 95%, while the validation accuracy reaches around 90%, indicating stable learning and good generalization. With a learning rate of 0.01, the model converges faster, achieving training accuracy above 95% and validation accuracy around 92%, though slight overfitting is observed. In contrast, a learning rate of 0.1 causes unstable training, with high fluctuations in accuracy and lower validation performance, peaking at approximately 85%, likely due to overshooting during optimization. Overall, the learning rate of 0.01 strikes the best balance between convergence speed and performance, while 0.001 ensures stable but slower learning, and 0.1 proves too aggressive for reliable training.

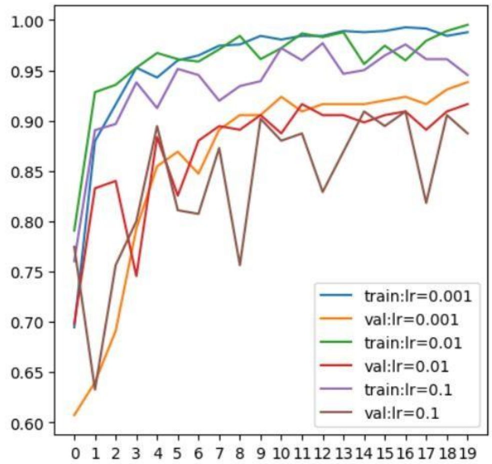

Figure 7. The learning curve for learning rates in Experiment 2

It can be concluded that the most suitable learning rate for this experiment is 0.001, which had an accuracy and validation accuracy of 98.78% and 93.82%, respectively. The model added inner dense layers after deciding on the learning rate value. There were three different values for dense layers: 10, 100, and 1000, as shown in table 6, table 7 and table 8.

Table 6. Inner Dense Layers in Size 10

Inner Dense Layers in Size 10				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.5491	0.7847	0.6898	0.6727
2	0.2460	0.9027	0.5714	0.7564
3	0.1845	0.9307	0.4179	0.8582
⋮	⋮	⋮	⋮	⋮
18	0.0115	0.9988	0.2370	0.9309
19	0.0130	0.9988	0.2478	0.9345
20	0.0081	1.0000	0.2309	0.9309

Table 7. Inner Dense Layers in Size 100

Inner Dense Layers in Size 100				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.5413	0.8005	0.5118	0.8182
2	0.1289	0.9550	0.5612	0.7709
3	0.1006	0.9611	0.4141	0.8545
⋮	⋮	⋮	⋮	⋮
18	0.0258	0.9891	0.3796	0.9164
19	0.0165	0.9939	0.3149	0.9309
20	0.0128	0.9988	0.3369	0.9273

Table 8. Inner Dense Layers in Size 1000

Inner Dense Layers in Size 100				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.6659	0.8455	1.4510	0.6436
2	0.1627	0.9501	1.1226	0.7345
3	0.0926	0.9720	1.0117	0.8036
⋮	⋮	⋮	⋮	⋮
18	0.0140	0.9939	0.7551	0.9091
19	0.0207	0.9927	0.8671	0.9236
20	0.0166	0.9939	1.1300	0.9018

Figure 8 presents the learning curves for different sizes of the inner dense layers during training, using a learning rate of 0.001. The figure compares three configurations with layer sizes of 10, 100, and 1000, showing their respective training accuracy across epochs. All three configurations exhibit a similar pattern of rapid convergence within the first few epochs, where the accuracy quickly increases to above 90%. As training progresses, the curves stabilize near 95%, with only minor variations between the different layer sizes. The performance of the model remains consistent regardless of the size of the inner dense layers, suggesting that increasing the layer size does not significantly impact training accuracy for this learning rate. This indicates that smaller layer sizes may be sufficient to achieve optimal performance, providing a more efficient model without sacrificing accuracy.

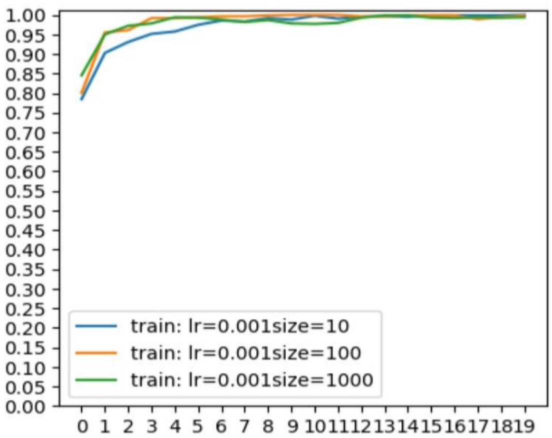

Figure 8. The learning curve for inner dense layers' size

The model proceeded with 1000 inner dense layers with an accuracy of 99.39%. Furthermore, the model needs drop-out regularization to prevent overfitting in deep neural networks. The drop rate in this experiment has three different values: 0.0, 0.2, and 0.5 (see table 9, table 10, and table 11 for the result)

Table 9. Drop rate 0.0.

Drop rate 0.0				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.7579	0.8224	1.1004	0.7273
2	0.1863	0.9538	1.4221	0.6873
3	0.0953	0.9745	0.7601	0.8145
⋮	⋮	⋮	⋮	⋮
38	0.0430	0.9927	0.9322	0.9127
39	0.0259	0.9915	1.1763	0.9345

Drop rate 0.0				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
40	0.0409	0.9939	1.0107	0.9018

Table 10. Drop rate 0.2.

Drop rate 0.2				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.8240	0.8054	0.5630	0.8255
2	0.1750	0.9392	0.8245	0.7782
3	0.0704	0.9708	1.0091	0.7527
⋮	⋮	⋮	⋮	⋮
38	0.2171	0.9696	1.3597	0.9164
39	0.1022	0.9818	1.4732	0.9018
40	0.0359	0.9951	1.4783	0.9236

Table 11. Drop rate 0.5.

Drop rate 0.5				
Epoch	**Loss**	**Accuracy**	**Val_loss**	**Val_accuracy**
1	0.8012	0.7981	1.1264	0.6836
2	0.2359	0.9270	0.7468	0.7709
3	0.2133	0.9355	0.7342	0.7927
⋮	⋮	⋮	⋮	⋮
38	0.1565	0.9830	1.4586	0.8982
39	0.1059	0.9830	1.3930	0.9055
40	0.1469	0.9805	1.2101	0.9055

The chosen drop rate for this experiment is 0.5, with an accuracy of 98.05% and validation accuracy at 90.55% training with 40 epochs. After experimenting with different learning rates, inner dense layers, and drop rates, the model is now ready to train all pre-processed images and the parameters set at the beginning of the chapter. The training model takes the learning rate value of 0.001, inner dense layers in size 1000, and the drop rate at 0.5 with 40 epochs. The training results show a loss value of 1.83%, representing the average difference between the predicted values and the actual ground truth labels during the training process as shown in figure 9.

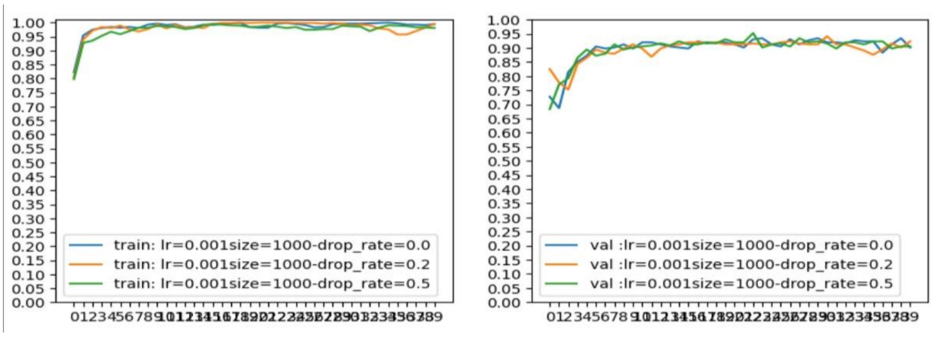

Figure 9. The learning curve for drop rates is 0.0, 0.2, and 0.5.

The accuracy value has an excellent score of 99.64%, indicating that the model performs well and makes accurate predictions on the training data. The validation loss value is 74.17%, and validation accuracy is 93.09%. The validation loss is relatively low, but the validation accuracy is high, indicating that the model performs well on unseen data without significant overfitting. These results suggest that the model is well-trained and can accurately predict new, unseen data, as shown in figure 10.

```
test_loss, test_accuracy = model.evaluate(test_dataset)
print(f'Test accuracy = {test_accuracy:0.4f}')
```

```
9/9 [==============================] - 50s 5s/step - loss: 0.2338 - accuracy: 0.92
00
Test accuracy = 0.9200
```

Figure 10. Model testing results

This model's test accuracy is 0.9200, meaning it can make correct predictions for approximately 92% of the samples in the test dataset. The results indicate that the model performs well and can generalize its learned patterns to new data or images. The test results for all three types of durians, Black Thorn, D24, and Musang King, are successfully classified with scores of 83.19%, 89.19%, and 99.98%, respectively. Black Thorn scored slightly lower compared to the other types, and this may be caused by the imbalance dataset, which initially had only 71 total images compared to Musang King, which had the highest data of 772 photos, and D24, which had 529 images before preprocessing. Figure 11, figure 12, and figure 13 illustrates the testing process carried out in this study.

```
1/1 [==============================] - 0s 146ms/step
{'Black Thorn': 0.8319031, 'D24': 0.01665693, 'Musang King': 0.15144002}
```

Figure 11. Test results for Black Thorn

```
1/1 [==============================] - 0s 151ms/step
{'Black Thorn': 0.0031514799, 'D24': 0.89190954, 'Musang King': 0.10493899}
```

Figure 12. Test results for D24

```
1/1 [==============================] - 0s 145ms/step
{'Black Thorn': 7.443643e-05, 'D24': 0.00013671488, 'Musang King': 0.9997888}
```

Figure 13. Test results for Musang King

3.3. Prototype Development

The user interface for this project was developed using Gradio (see figure 14), a Python library that helps create a simplified web interface that allows users to interact with the model. Users can upload an image, and the prototype will perform durian classifications accordingly. The prototype enables the user to upload one photo at a time to make a prediction, and the user can re-enter new input multiple times.

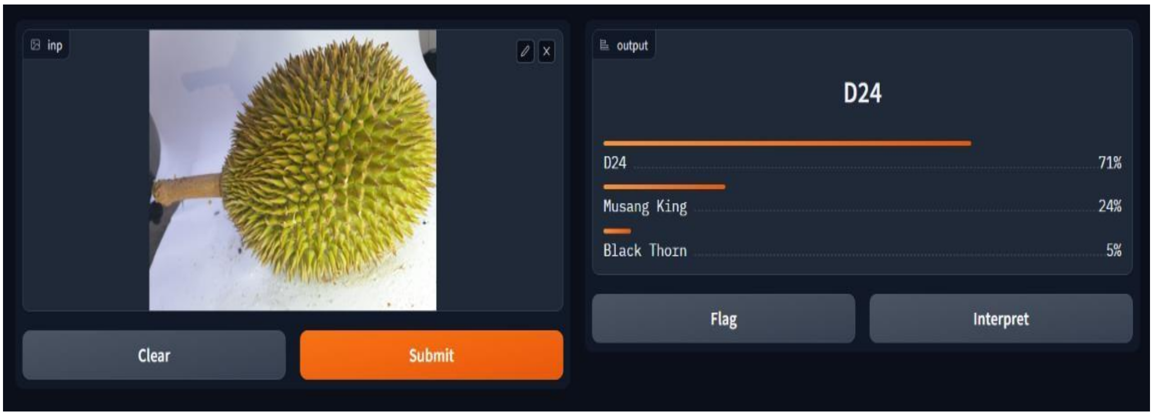

Figure 14. Prototype Testing

4. Conclusion

Through extensive literature review and analysis, we have identified the critical visual features that distinguish different types of durians, such as Musang King, D24, and Black Thorn. These features include color, texture, shape, and other crucial characteristics in the recognition process. Understanding these distinctive features has laid the groundwork for developing an accurate and robust durian recognition prototype.

After evaluating various state-of-the-art models, we have successfully applied a pre-trained deep learning model, such as Xception or VGG-16, for durian recognition. By fine-tuning the model on our durian dataset, we have achieved high accuracy in classifying durians into their respective types. The chosen model's ability to capture intricate patterns and variations in durian images has significantly improved the recognition performance. Using the pre-trained model as the backbone, we have created an interactive and user-friendly prototype for durian recognition. The prototype, built with Gradio, allows users to upload their durian images and receive real-time predictions for their durian types. This prototype serves as a proof of concept and showcases the potential of the developed model in a practical and accessible manner.

However, the project also faces certain limitations that need to be addressed. The availability of a diverse and extensive durian dataset is crucial for the model's performance. While efforts have been made to collect and curate a comprehensive dataset, the limited availability of labeled durian images may introduce biases and affect the model's generalization capabilities. Additionally, durians exhibit natural variations in color, shape, and texture, even within the same type. These variations challenge the model in accurately distinguishing durians with subtle differences. Improving the model's robustness to account for these variations remains an ongoing research goal. Several recommendations and future work directions can be suggested to address the limitations and enhance the project's effectiveness. Expanding the dataset with new durian images collected from various sources and environments will improve the model's generalization and adaptability. Hyperparameter tuning is another crucial aspect that can significantly impact model performance. Systematic exploration of different hyperparameter settings and optimization techniques will optimize the model's training process and boost accuracy.

Assembling multiple models or predictions can further enhance recognition accuracy and provide more reliable results. Finding which method works best and selecting appropriate combinations will be valuable for the project's future success. Engaging with durian experts, growers, and consumers throughout the development process will ensure that the recognition model and prototype meet their needs and requirements since they are highly knowledgeable about durians. Community engagement can also facilitate gathering valuable feedback and refining the project's direction.

5. Declarations

5.1. Author Contributions

Conceptualization: D.D., T.B.K., D.A.D., M.Z.Z., E.F.E.F.; Methodology: E.F.E.F.; Software: D.D.; Validation: D.D., E.F.E.F., and M.Z.Z.; Formal Analysis: D.D., E.F.E.F., and M.Z.Z.; Investigation: D.D.; Resources: E.F.E.F.; Data

 ISSN 2723-6471

Curation: E.F.E.F.; Writing Original Draft Preparation: D.D., E.F.E.F., and M.Z.Z.; Writing Review and Editing: E.F.E.F., D.D., and M.Z.Z.; Visualization: D.D. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] N. A. Husin, S. Rahman, R. Karunakaran, and S. J. Bhore, "A review on the nutritional, medicinal, molecular and genome attributes of Durian (Durio zibethinus L.), the King of fruits in Malaysia," *Bioinformation*, vol. 14, no. 6, pp. 265–270, Jun. 2018. doi: 10.6026/97320630014265.

[2] M. Mohd Ali, N. Hashim, S. A. Aziz, and O. Lasekan, "Exploring the chemical composition, emerging applications, potential uses, and health benefits of durian: A review," *Food Control*, vol. 113, no. 1, p. 107189, 2020. doi: 10.1016/j.foodcont.2020.107189.

[3] A. M. Wahid, T. Hariguna and G. Karyono, "Optimizing Feature Extraction for Website Visuals: A Comparative Study of AlexNet and Inception V3," *2024 12th International Conference on Cyber and IT Service Management (CITSM), Batam, Indonesia*, vol. 2024, no. 1, pp. 1-6, doi: 10.1109/CITSM64103.2024.10775681.

[4] S. Ketsa, A. Wisutiamonkul, Y. Palapol, and R. E. Paull, "The Durian," *in Horticultural Reviews*, vol. 2020, no. 1, pp. 125–211, 2020. doi: 10.1002/9781119625407.ch4.

[5] N. Mohd Nordin, H. Anuar, Y. F. Buys, F. Ali, S. Thomas, and N. A. Mohd Nasir, "Effect of freeze-dried durian skin nanofiber on the physical properties of poly(lactic acid) biocomposites," *Polymer Composites*, vol. 42, no. 2, pp. 842–848, Feb. 2021. doi: 10.1002/pc.25869.

[6] N. A. A. Aziz and A. M. Mhd Jalil, "Bioactive Compounds, Nutritional Value, and Potential Health Benefits of Indigenous Durian (Durio Zibethinus Murr.): A Review," *Foods*, vol. 8, no. 3, pp. 1–15, 2019. doi: 10.3390/foods8030096.

[7] S. Thongkaew, C. Jatuporn, P. Sukprasert, P. Rueangrit, and S. Tongchure, "Factors affecting the durian production of farmers in the eastern region of Thailand," *International Journal of Agricultural Extension*, vol. 9, no. 2, pp. 285–293, 2021. doi: 10.33687/ijae.009.02.3617.

[8] C. Airriess, "Constructing durian terroir and geographical indications in Penang, Malaysia," *Singapore Journal of Tropical Geography*, vol. 41, no. 1, pp. 6–22, Jan. 2020. doi: 10.1111/sjtg.12298.

[9] C. S. Leng, N. M. I. M. Nor, L. Zakaria, and S. N. M. Sidique, "Durian stem canker status and relative soil properties in Peninsular Malaysia," *Journal of Sustainability Science and Management*, vol. 19, no. 6, pp. 85–97, Jun. 2024. doi: 10.46754/jssm.2024.06.007.

[10] L. T. Duong, P. T. Nguyen, C. Di Sipio, and D. Di Ruscio, "Automated fruit recognition using EfficientNet and MixNet," *Computers and Electronics in Agriculture*, vol. 171, no. 1, p. 105326, 2020. doi: 10.1016/j.compag.2020.105326.

[11] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018. doi: 10.1109/JIOT.2018.2844296.

[12] A. S. Prabowo and L. J. Mawarani, "Edible coating development of durian seeds starch and glucomannan with the addition of essential oil as an antimicrobial to increase shelf life of tomato and cauliflower," *IOP Conference Series: Materials Science and Engineering*, vol. 833, no. 1, p. 012034, Jun. 2020. doi: 10.1088/1757-899X/833/1/012034.

[13] X. Lin, S. Yang, J. Zhao, W. Zhang, Y. Li, and F. Zhang, "Assessment of genetic diversity and discovery of molecular markers in durian (Durio zibethinus L.) in China," *Diversity*, vol. 14, no. 9, pp. 1–12, 2022. doi: 10.3390/d14090769.

[14] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Computer Networks*, vol. 168, no. 1, p. 107036, 2020. doi: 10.1016/j.comnet.2019.107036.

[15] M. G. Lim and J. H. Chuah, "Durian types recognition using deep learning techniques," *in 2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, vol. 2018, no. 1, pp. 183–187, 2018. doi: 10.1109/ICSGRC.2018.8657535.

[16] D. N. V. S. L. S. Indira, J. Goddu, B. Indraja, V. M. L. Challa, and B. Manasa, "A review on fruit recognition and feature evaluation using CNN," *Materials Today: Proceedings*, vol. 80, no. 1, pp. 3438–3443, 2023. doi: 10.1016/j.matpr.2021.07.267.

[17] Y. A. S. A. F. A. Harmandeep Singh Gill, Osamah Ibrahim Khalaf, "Multi-Model CNN-RNN-LSTM based fruit recognition and classification," *Intelligent Automation and Soft Computing*, vol. 33, no. 1, pp. 637–650, 2022. doi: 10.32604/iasc.2022.022589.

[18] S. S. S. Palakodati, V. R. Chirra, Y. Dasari, and S. Bulla, "Fresh and rotten fruits classification using CNN and transfer learning," *Revue d'Intelligence Artificielle*, vol. 34, no. 5, pp. 617–622, 2020. doi: 10.18280/ria.340512.

[19] X. Yang, Z. Zeng, S. G. Teo, L. Wang, V. Chandrasekhar, and S. Hoi, "Deep learning for practical image recognition: Case study on Kaggle competitions," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '18)*, vol. 2018, no. 1, pp. 923–931, 2018. doi: 10.1145/3219819.3219907.

[20] E. Barua, "Durian Dataset," *Kaggle*, 2021. [Online]. Available: https://www.kaggle.com/datasets/edenbarua/picture/data.

[21] G. B. Sujang, S. D. Ramaiya, N. Saupi, and S. Y. Lee, "Profiling of volatile organic compounds (VOCs) of wild edible durians from Sarawak, Borneo associated with its aroma properties," *Horticulturae*, vol. 9, no. 2, pp. 257-269, 2023.