Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1701~1708
e-ISSN: 2548-6861

Turtle Dove Classification Using CNN Algorithm With MobileNetV2 Transfer Learning

Muhammad Minanul Lathif [1]*, Sendi Novianto [2]
Study Program in Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
lathifminanul@gmail.com[1], sendi.novianto@dsn.dinus.ac.id[2]

Article Info

Article history:

Received 2025-02-19
Revised 2025-07-16
Accepted 2025-07-30

Keyword:

Classification,
Convolutional Neural Network,
MobileNetV2,
Transfer Learning,
Turtle Dove.

ABSTRACT

This study aims to optimize the performance of a Convolutional Neural Network (CNN) model based on the MobileNetV2 architecture in classifying Java sparrow images by testing four main parameters: optimizer, learning rate, number of epochs, and batch size. The dataset consists of 800 images divided evenly into four classes. The results show that using the Adam optimizer yields the best accuracy with a training accuracy of 97.50%, validation accuracy of 98.75%, and testing accuracy of 98.13%. A learning rate of 0.001 produces the same results, indicating consistent performance with this configuration. Epoch testing shows that 35 epochs yield the highest performance with a training accuracy of 98.39%, validation accuracy of 100%, and testing accuracy of 98.75%. Meanwhile, batch size testing shows that a batch size of 32 yields the highest testing accuracy of 98.85%, a batch size of 64 yields the highest training accuracy of 98.63%, and a batch size of 128 yields the highest validation accuracy of 99.58%. These findings suggest that smaller batch sizes tend to yield better performance in terms of model generalization, while larger batch sizes provide higher stability in the training process but do not always reflect actual performance on the test data. The results of this study can serve as a reference for selecting parameter configurations to improve the accuracy and generalization of image classification models using MobileNetV2. These results emphasize the importance of proper parameter settings in improving the accuracy and stability of image classification models. They can be a reference in model development in object recognition.

I. Introduction

Indonesia has 1539 bird species or 17% of all bird species in the world. Three hundred and eighty-six of these species are endemic to Indonesia and are naturally found in Indonesia[1]. Birds are almost found in every place and have an important position as one of Indonesia's animal wealth. The species are very diverse and each type has its own beauty value.

Raising animals, especially birds, has become a popular hobby for many people in Indonesia [2][3]. Apart from providing entertainment, keeping birds can also provide emotional satisfaction for the owner. Various types of birds, ranging from those with melodious voices to those with exotic looks, are often kept as pets at home[4]. This activity not only strengthens the relationship between humans and animals, but also fosters a sense of responsibility in caring for living things. In addition, keeping birds can be an educational tool for children to learn about nature and animal life[5]. However, it is important for keepers to understand the needs and welfare of their pets in order to create a healthy and comfortable environment for the birds[6].

The turtle dove (Geopelia striata) is a member of the Columbidae family [7]. In general, this bird can be recognized by characteristics such as relatively small body size and dark brownish body feather color. The turtle dove (Geopelia striata) is one of the most popular bird species in Indonesia, especially among bird lovers. The beauty of its melodious voice and elegant appearance makes this bird attractive to many people[8].

In the market there are various breeds and color variants of commonly found tertle dove birds, such as turtle dove local, cemani ordinary, cemani majapahit, and golden. For breeds and mocca, silver, cotton white, striated white, beige, and black are the color variants [8].

A problem faced by the general public is the difficulty in accurately identifying the species. Most people only know turtle dove in general without paying attention to the detailed differences that may exist between one type and another. This is due to the variety of feather colors and patterns that are quite similar between one type and another and the large number of types of turtle dove birds. This can be an obstacle when you want to get a turtle dove bird with a certain type as desired.

In the field of digital image processing, there are various algorithms that can be used, including KNN, Support Vector Machine, and Convutional Neural Network. Like the research that has been done by, R A Saputra,[9] analyzing images of rice leaf disease using the KNN Based On GLCM Feature Extraction algorithm with an accuracy of 65.83%. In research by Muhathir,[10] The use of SVM algorithm with kernel on puppet image classification with 83.4% accuracy. Meanwhile, research by Katarzyna Baran,[11] Smartphone thermal imaging for stressed people classification using CNN MobileNetV2 produces 91% accuracy. From the use of algorithms that have been used for research, it proves that the CNN algorithm with MobilenetV2 architecture is a good algorithm for image classification.

The highest accuracy results are from the algorithm above in the CNN MobilenetV2 algorithm. Therefore, the researcher proposes a solution to overcome this problem using the Convolutional Neural Network (CNN) algorithm through a transfer learning approach using MobileNetV2 to classify turtle dove images. CNN is a deep learning algorithm that is excellent at extracting task-specific features, such as color, texture, and shape patterns. This feature learning is closely related to the classification task to achieve better performance[12]. In this study, the MobileNet model, which is known to be efficient in image processing, was used [13], is used as the base model for *transfer learning* on the turtle dove bird image dataset. By applying transfer learning, the model can utilize the pre-trained weights in (ImageNet), thus speeding up the training process and improving the accuracy of multi-class classification according to the needs of the dataset.

The main objective of this research is to provide practical solutions for the community of turtle dove lovers in recognizing and distinguishing turtle dove species more easily and accurately. This CNN and MobileNet-based classification system is expected to identify the unique characteristics of each type of bird, such as color, texture, and shape patterns. With this technology, it is expected that people can better understand the diversity of turtle dove species and can choose birds according to their preferences more efficiently.

The development of turtle dove identification technology using CNN algorithm with MobileNet transfer learning also has the potential to make a significant contribution in the field of ornithology and conservation in Indonesia. By improving accuracy and efficiency in species recognition, this system can support the preservation of Indonesia's rich fauna and maintain the balance of existing natural ecosystems.

II. METHOD

Several research methods can be used to classify turtle dove images, such as utilizing image processing techniques with the MobileNet model which is one of the architectures of the Convolutional Neural Network (CNN) algorithm. The process of identifying turtle dove birds used in the classification of turtle dove birds is shown in Figure 1.

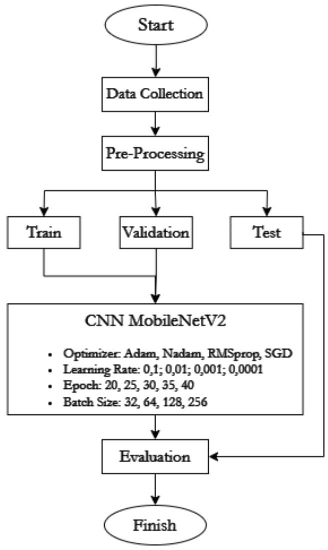


Figure 1: Research Process

A. *Data Collection*

Research in the classification of this type of turtle dove takes data in the form of images of turtle dove taken manually by researchers. There are 800 image data used, which are categorized into 200 cemani image data, 200 mocca image data, 200 white image data, 200 silver image data. When taking pictures, researchers used a sony A6000 mirrorles camera with a distance of 40-50cm. Since the data collection method of the turtle dove images was done manually, where each image was taken with a variety of positions, it can be ensured that the collected images are clean from duplication, noise, or other anomalies that can be the reason why the data cleaning process is needed. Considering this, the stage that will be carried out in this study after collecting all the necessary data is pre-processing without cleaning the data.

B. *Pre-processing*

Once all the turtle dove images required for the study are collected, the next step is data pre-processing. During this process, a number of processes are performed such as Format standardization or converting the format of the entire image into the same format. The purpose of format standardization is to reduce errors and improve the efficiency of the process as different formats may give different characteristics to the image.[14]. The format used in this research is JPG format. After all the image data has the same format, the next step is to resize the image or change the size of the turtle dove image to 224 x 224 pixels so that the image dimensions remain consistent. The next step is image normalization which is one of the steps to accelerate convergence, improve accuracy, and ensure model consistency by changing the ratio and distribution of pixel values in the image.[15]. Researchers decided to use the Min-Max normalization method in this study. This method works by changing the original data linearly to produce a balanced comparison result value between the data before and after the process. This method can be used to normalize images by changing the pixel value from 0 to 1.[16]. The Min-Max normalization technique works as follows:

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$$

Where x_{new} is the normalized pixel value, x_{old} is the original pixel value before normalization, x_{max} is the maximum pixel value in the original image, and x_{min} is the minimum pixel value in the original image[17].

After the normalization stage using the Min-Max method to ensure a consistent scale of data values, the next step is the application of the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique. CLAHE is an image processing technique used to enhance image contrast in a better and controlled way. In this process, the image is divided into small blocks and subjected to histogram equalization.[18]. This technique serves to even out the distribution of pixel intensity in each block resulting in better contrast, especially in areas with uneven lighting

The next step is to divide all data into training data, validation data and test data with a ratio of 7:1:2, namely 70% as training data, 10 as validation data, and 20% as test data. The use of this ratio aims for a sufficient training and evaluation phase of the model, resulting in good accuracy. In dividing the data, the random method is used, after the data has been successfully divided randomly according to the predetermined ratio, the data is stored in a new folder so that it is easy to use for the next step.

The last step before heading to the modeling stage is the image augmentation process. Image augmentation is one of the techniques that can change images with various orientations, such as enlarging the shape and size of the image. This augmentation activity is carried out to increase the size of the dataset by changing the data to prevent the model from memorizing certain patterns.[19]. In this research, the augmentation process uses a number of techniques including 20 degree shear, 20 degree rotation, 20% zoom and horizontal data reversal. These techniques will be performed using ImageDataGenerator from Keras, the data that has been incorporated into the model will be forwarded.

C. *CNN MobileNetV2*

In this research, the Convolutional Neural Network (CNN) model used is MobileNetV2, which is a CNN architecture specifically designed for mobile applications and devices with limited resources. MobileNetV2 is a development of MobileNetV1 which integrates the concept of depthwise separable convolution with a linear bottleneck that allows better computational efficiency without sacrificing accuracy [20]. The CNN structure used in this study consists of several convolutional layers that function to extract features from the input image. To provide a clearer picture of the structure of CNN and MobileNetV2, the following image shows the CNN MobileNetV2**.**

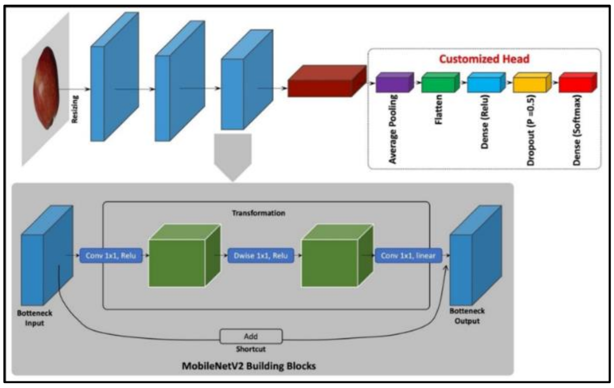


Figure 2. CNN MobileNetV2

MobileNetV2 relies on a lighter structure by reducing the number of parameters and computational operations, enabling faster and more efficient image processing, especially on devices with low computing power. The next stage is the creation of the most optimal model, to get the best model performance by testing in determining the optimizer, learning rate, epoch, and batch size. plays an important role in improving the accuracy and speed of training.

1) *Proper optimizer* selection is critical in improving training accuracy and speed[21]. Several types of optimizers that have been determined by researchers, such as Adam, Nadam, RMSprop, and SGD will be tested to determine the most suitable for the dataset used.

2) *Furthermore*, the optimal learning rate parameter is also tested, as too large a learning rate value can cause the model to fail to converge, while too small a value can slow down the training process.[22]. Therefore, experiments were conducted with various learning rates including 0.1; 0.01; 0.001; 0.0001; with various learning rate values to find the value that gives the best results in terms of convergence and accuracy.

3) *Furthermore*, Epoch testing is performed which is a parameter defining one trajectory through the complete training set when training a deep learning model.[23]. Tests were conducted with several different epoch values, namely 20, 25, 30, 35, 40. This trial aims to find the number of epochs that produce effective training without losing the model's ability to generalize.

4) *Finally*, the batch size was tested with two different sizes, namely 32, 64, 128, 256 to determine the size of the data group processed in one training step. Smaller tested batch sizes allow for more frequent model updates, and the resulting gradient updates will mostly be noise[24]. On the other hand, larger batch sizes improve training efficiency but require more memory. Therefore, experiments were conducted with various batch sizes to find the optimal size according to the dataset.

D. Evaluation

The model evaluation process is carried out by utilizing test data to measure the performance of the model that has been developed. This stage aims to identify the extent to which the model is able to provide accurate results, using the Precision, Accuracy, Recall, and F1 score evaluation metrics. calculated with the following formula.

1) *Precision*: is the total predicted positive case rate that has been predicted from the positive case rate. A high precision is associated with a fairly low positive rate[25].

$$Precission = \frac{tp}{tp + fp}$$

2) *Accuracy*: is a measure of the ratio between the number of correct predictions and the total number of samples in the evaluation dataset[26].

$$Accuracy = \frac{tp + tn}{tn + tp + fp + fn}$$

3) *Recall*: is the proportion of documents that are classified as positive by the model compared to all data that are actually positively labeled[27].

$$Recall = \frac{tp}{tp + fn}$$

4) *F1-score*: is a matrix value for assessing model performance that combines precision and recall measure in the concept of harmonic mean[28].

$$F1score = \frac{2 \times Precission \times Recall}{Precission + Recall}$$

III. RESULTS AND DISCUSSION

a b c d

Figure 3. Trutle Dove image samples (a) cemani image, (b) mocca image, (c) White image, (d) silver image.

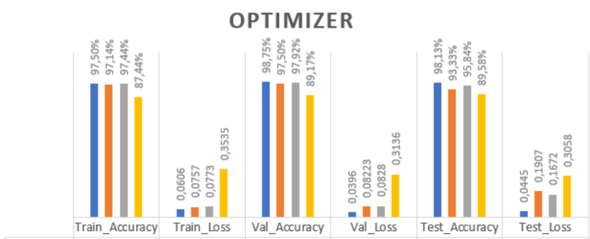
Based on Figure 5, the dataset used in this research is a collection of image data obtained through field studies by taking pictures directly using a Sony A6000 Mirrorless camera. The image capture process is done manually maintaining a distance of 40-50 cm between the camera and the object. The dataset consists of 800 images divided into 4 categories, namely, cemani, mocca, white, and silver. The image data will be used for testing using CNN MobilenetV2 by testing 4 parameters, namely optimizer, learning rate, epoch, and batch size which aims to get the best performance results in the classification of turtle dove images by looking at the parameter values of accuracy, recall, precision, and f1-score.

A. System Testing

System testing will be carried out with four scenarios, namely comparing various optimizers, comparing learning rate values, comparing the number of epochs, and comparing batch size.

1. Optimizer Testing

In the training process, the selection of the optimizer is very important, because it can train the model significantly. The first test is optimizer testing to determine the most efficient type of optimizer in classifying turtle dove images. Optimizers used in classifying turtle dove images include Adam, Nadam, RMSprop, and SGD. At this stage the researcher determines the parameters with a learning rate of 0.001, number of epochs 25 and batch size 32. The test results in comparing optimizers are shown in the figure.



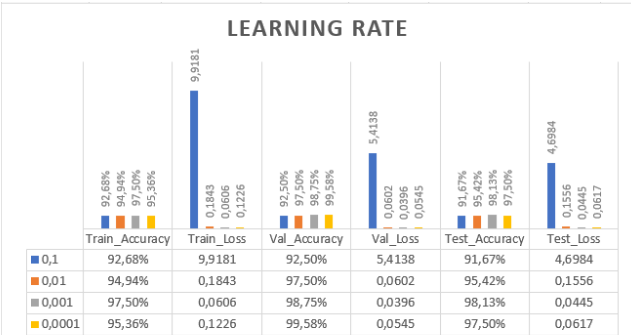
	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss
Adam	97,50%	0,0606	98,75%	0,0396	98,13%	0,0445
Nadam	97,14%	0,0757	97,50%	0,08223	93,33%	0,1907
RMSprop	97,44%	0,0773	97,92%	0,0828	95,84%	0,1672
SGD	87,44%	0,3535	89,17%	0,3136	89,58%	0,3058

Figure 4: Bar chart optimizer testing

Based on the figure above, we get efficient results on the Adam optimizer with an training accuracy 97.50%, validation accuracy 98.75%, and testing accuracy 98.13%. Adam's optimizer (Adaptive Moment Estimation) is a combination of two optimizations, namely Momentum and RMSProp, so it has advantages in fast convergence and stability when facing complex loss functions. Furthermore, the Adam optimizer will be used as a comparison test of learning rate, epoch and batch size.

2. Learning Rate Testing

The second stage is to test the learning rate value by using the best analysis results from the first stage, namely determining the optimizer. This second stage uses a learning rate value of 0.1; 0.01; 0.001; 0.0001; with the optimizer Adam, number of epochs 25, and batch size 32. The following are the results of learning rate testing.



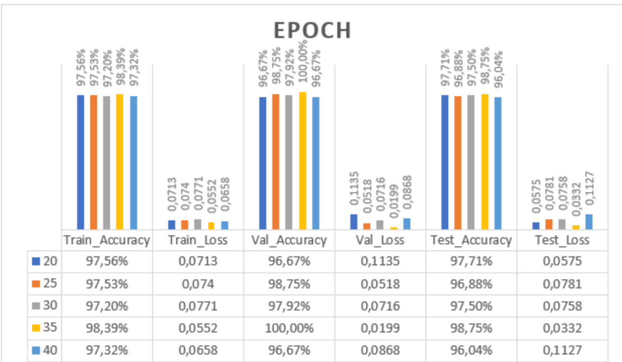
	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss
0,1	92,68%	9,9181	92,50%	5,4138	91,67%	4,6984
0,01	94,94%	0,1843	97,50%	0,0602	95,42%	0,1556
0,001	97,50%	0,0606	98,75%	0,0396	98,13%	0,0445
0,0001	95,36%	0,1226	99,58%	0,0545	97,50%	0,0617

Figure 5: Bar chart learning rate testing

Based on the picture above, the highest accuracy result is obtained at a learning rate of 0.001 with an training accuracy value of 97.50%, %, validation accuracy 98.75%, and testing accuracy 98.13%. Furthermore, the learning rate of 0.001 will be used as a comparison test of epoch and batch size.

3. Epoch Testing

Epoch is a training process that is carried out repeatedly in order to achieve optimal accuracy results. Epoch testing which is a test of the number of epochs to find out the best results from previous tests. The number of epochs used in this study are 20, 25, 30, 35, and 40 with the provisions of the Adam optimizer, the learning rate value of 0.001 and batchsize 32 which have been determined from the results of previous tests. The following are the results of testing the epoch:



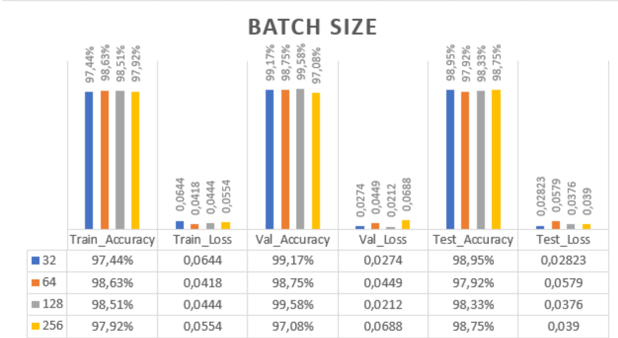
	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss
20	97,56%	0,0713	96,67%	0,1135	97,71%	0,0575
25	97,53%	0,074	98,75%	0,0518	96,88%	0,0781
30	97,20%	0,0771	97,92%	0,0716	97,50%	0,0758
35	98,39%	0,0552	100,00%	0,0199	98,75%	0,0332
40	97,32%	0,0658	96,67%	0,0868	96,04%	0,1127

Figure 6: Bar chart epoch testing

Based on the figure above, we get high accuracy results in epoch testing with the number of epochs 35. In this test, the number of epochs 35 with has an training accuracy 98.39%, validation accuracy 100%, and testing accuracy 98.75%. From this value it can be seen that the number of epochs 35 is better than the number of epochs 20, 25, 30 and 40.

4. Batch Size Testing

Batch Size is the size of the dataset division in each batch at each epoch to speed up the training process. The last scenario involves testing the batch size value using the best parameters obtained from the previous scenario.



	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss
32	97,44%	0,0644	99,17%	0,0274	98,95%	0,02823
64	98,63%	0,0418	98,75%	0,0449	97,92%	0,0579
128	98,51%	0,0444	99,58%	0,0212	98,33%	0,0376
256	97,92%	0,0554	97,08%	0,0688	98,75%	0,039

Figure 7: Bar chart batch size testing

a) *Testing batch size 32*

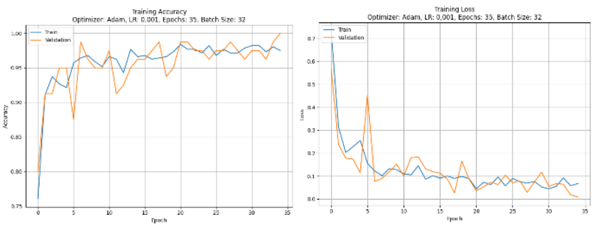


Figure 8: Batch Size 32 Performance

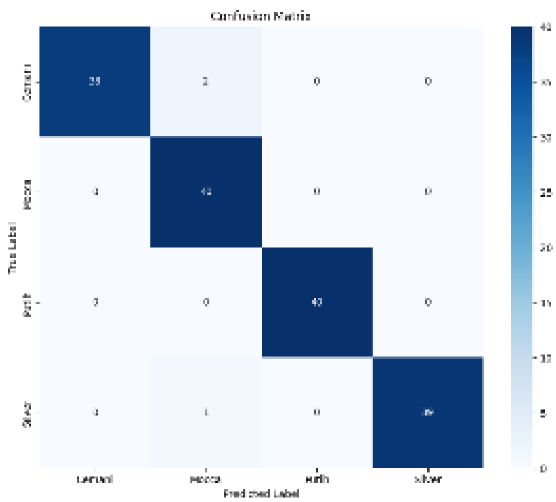


Figure 9: Batch Size 32 Confusion Matrix

TABLES 1
BATCH SIZE 32 TESTING RESULT

Scenario	Precision	Recall	F1-Score	Accuracy
Cemani	0.99	0.97	0.98	0.97
Mocca	0.97	1.00	0.98	0.97
Putih	1.00	1.00	1.00	0.97
Silver	1.00	0.98	0.99	0.97

Figure 8 and Figure 9 above present the results of testing batch size 32, with Figure 8 showing the model graph and Figure 9 showing the confusion matrix. In addition, the results of the evaluation metrics are presented separately in Table 1. Based on the test results, batch size 32 produces of training accuracy 97.44%, validation accuracy 99,17%, and testing accuracy 98,95%. Based on the graph and evaluation, the model is quite good because it produces a fitting model.

b) *Testing batch size 64*

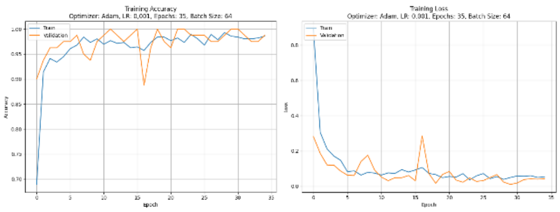


Figure 10: Batch Size 64 Performance

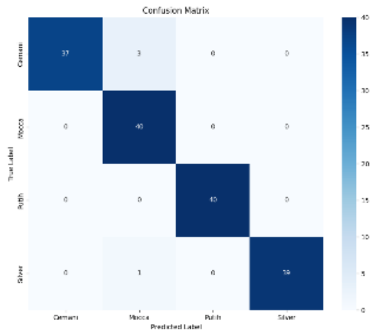


Figure 11: Batch Size 64 Confusion Matrix

TABLES 2
BATCH SIZE 64 TESTING RESULT

Scenario	Precision	Recall	F1-Score	Accuracy
Cemani	0.99	0.97	0.98	0.98
Mocca	0.94	0.99	0.96	0.98
Putih	0.99	1.00	1.00	0.98
Silver	1.00	0.96	0.98	0.98

Figure 10 and Figure 11 above present the results of testing batch size 64, with Figure 10 showing the model graph and Figure 11 showing the confusion matrix. In addition, the results of the evaluation metrics are presented separately in Table 2. Based on the test results, batch size 64 produces of training accuracy 98.63%, validation accuracy 98.75%, and testing accuracy 98.95%. Based on the graph and evaluation, the model is quite good because it produces a fitting model.

c) *Testing batch size 128*

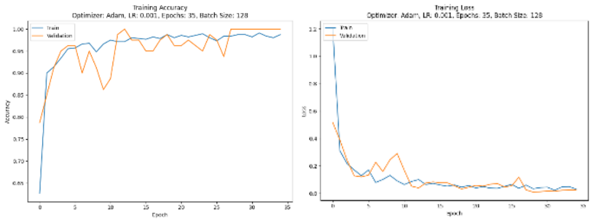


Figure12: Batch Size 128 Performance

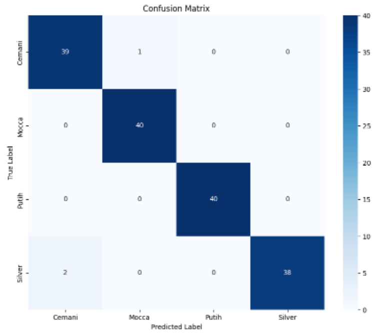


Figure 13: Batch Size 128 Confusion Matrix

TABLES 3
BATCH SIZE 128 TESTING RESULT

Scenario	Precision	Recall	F1-Score	Accuracy
Cemani	0.98	0.98	0.97	0.98
Mocca	0.96	1.00	0.98	0.98
Putih	1.00	1.00	1.00	0.98
Silver	1.00	0.96	0.98	0.98

Figure 12 and Figure 13 above present the results of testing batch size 128, with Figure 12 showing the model graph and Figure 13 showing the confusion matrix. In addition, the results of the evaluation metrics are presented separately in Table 3. Based on the test results, batch size 128 produces of training accuracy 98.51%, validation accuracy 99.58%, and testing accuracy 98.33%.

d) *Testing batch size 256*

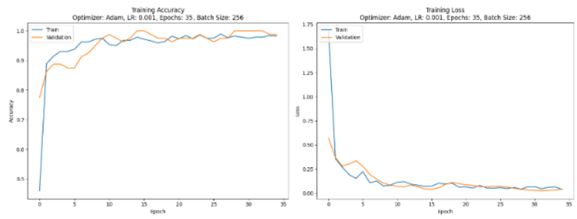


Figure 14: Batch Size 256 Performance

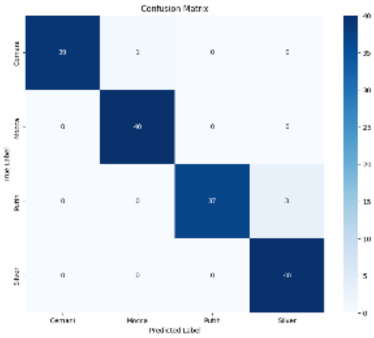


Figure 15: Batch Size 256 Confusion Matrix

TABLES 4
BATCH SIZE 256 TESTING RESULT

Scenario	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Cemani	0.98	0.98	0.99	0.98
Mocca	0.97	0.99	0.99	0.98
Putih	1.00	1.00	1.00	0.98
Silver	1.00	0.97	0.98	0.98

Figure 14 and Figure 15 above present the results of testing batch size 256, with Figure 14 showing the model graph and Figure 15 showing the confusion matrix. In addition, the results of the evaluation metrics are presented separately in Table 4. Based on the test results, batch size 256 produces quite high accuracy, although lower than the others, 97.92%, validation accuracy 97.08%, and testing accuracy 98.75%.

IV. CONCLUSION

In this study, testing was conducted to optimize the performance of the MobileNetV2-based CNN model in classifying tekukur bird images by considering four parameters: optimizer, learning rate, epoch, and batch size. The testing results indicate that each parameter contributes to the model's accuracy. Testing on the optimizer showed that the Adam optimizer yielded the best results, with a training accuracy of 97.50%, a validation accuracy of 98.75%, and a testing accuracy of 98.13%. In the learning rate testing, a value of 0.001 produced the highest accuracy, matching the results from the Adam optimizer testing: 97.50% training accuracy, 98.75% validation accuracy, and 98.13% testing accuracy. In the epoch testing, 35 epochs produced the highest accuracy with a training accuracy of 98.39%, validation accuracy of 100%, and test accuracy of 98.75%.

For batch size, although the commonly used batch sizes are 32 and 64, in this study, the researchers tested up to a batch size of 256 and found that a batch size of 32 produced the highest test accuracy of 98.85%, while the highest train accuracy was achieved at a batch size of 64 with an accuracy of 98.63%, and the highest validation accuracy was at a batch size of 128 with a result of 99.58.

This study shows that smaller batch sizes provide higher test accuracy and train accuracy, indicating excellent generalization ability, although larger batch sizes provide the highest validation accuracy, which may not necessarily reflect the model's performance.

BIBLIOGRAPHY

[1] E. Prasetyo and R. Wulandari, "Richness, Diversity, and Conservation Status of Bird Species in Maron Beach, Semarang, Indonesia," *Quagga J. Pendidik. dan Biol.*, vol. 13, no. 1, p. 95, 2020, doi: 10.25134/quagga.v13i1.3664.

[2] S. J. Marshall, Harry. Collar, Nigel J. Lees, Alexander C. Moss, Andrew. Yuda, Pramana. Marsden, "People and Nature - 2020 - Marshall - Characterizing bird-keeping user-groups on Java reveals distinct behaviours profiles." 2020.

[3] J. Iskandar, B. S. Iskandar, D. Mulyanto, R. L. Alfian, and R. Partasasmita, "Traditional ecological knowledge of the bird traders on bird species bird naming, and bird market chain: A case study in bird market pasty Yogyakarta, Indonesia," *Biodiversitas*, vol. 21, no. 6, pp. 2586–2602, 2020, doi: 10.13057/biodiv/d210631.

[4] "Conservat Sci and Prac - 2021 - Marshall - Understanding motivations and attitudes among songbird-keepers to identify best.pdf."

[5] R. L. White, K. Eberstein, and D. M. Scott, "Birds in the playground: Evaluating the effectiveness of an urban environmental education project in enhancing school children's awareness, knowledge and attitudes towards local wildlife," *PLoS One*, vol. 13, no. 3, pp. 1–23, 2018, doi: 10.1371/journal.pone.0193993.

[6] P. Rose and M. O'brien, "Welfare assessment for captive anseriformes: A guide for practitioners and animal keepers," *Animals*, vol. 10, no. 7, pp. 1–19, 2020, doi: 10.3390/ani10071132.

[7] K. Salaman and K. Magelang, "Inventory of Bird Species in Kedung Kopong and Banyak," vol. 4, no. February, pp. 125–128, 2021.

[8] F. Delfiah, H. R. Harun, S. A. Zahara, and S. Ayu, "Diversitas dan Etno-ornitologi Burung Bernilai Ekonomis sebagai Bentuk Kearifan Lokal Masyarakat di Pasar Hobi , Toddoppuli , Makassar (Diversity and Ethno-ornithology of Economically Valued Birds as a Form of Local Community Wisdom at Pasar Hobi , Toddo," vol. 3, no. Lc, pp. 17–32, 2024.

[9] R. A. Saputra, Suharyanto, S. Wasiyanti, D. F. Saefudin, A. Supriyatna, and A. Wibowo, "Rice Leaf Disease Image Classifications Using KNN Based on GLCM Feature Extraction," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012080.

[10] Muhathir and Al-Khowarizmi, "Measuring the Accuracy of SVM with Varying Kernel Function for Classification of Indonesian Wayang on Images," *2020 Int. Conf. Decis. Aid Sci. Appl. DASA 2020*, pp. 1190–1196, 2020, doi: 10.1109/DASA51403.2020.9317197.

[11] K. Baran, "Smartphone thermal imaging for stressed people classification using CNN+MobileNetV2," *Procedia Comput. Sci.*, vol. 225, pp. 2507–2515, 2023, doi: 10.1016/j.procs.2023.10.242.

[12] S. A. Agnes, J. Anitha, S. I. A. Pandian, and J. D. Peter, "Classification of Mammogram Images Using Multiscale all Convolutional Neural Network (MA-CNN)," *J. Med. Syst.*, vol. 44, no. 1, 2020, doi: 10.1007/s10916-019-1494-z.

[13] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," *2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019*, pp. 0280–0285, 2019, doi: 10.1109/UEMCON47517.2019.8993089.

[14] M. Cui and D. Y. Zhang, "Artificial intelligence and computational pathology," *Lab. Investig.*, vol. 101, no. 4, pp. 412–422, 2021, doi: 10.1038/s41374-020-00514-0.

[15] X. Pei *et al.*, "Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences," *Mater. Des.*, vol. 232, p. 112086, 2023, doi: 10.1016/j.matdes.2023.112086.

[16] H. J. Kim, J. W. Baek, and K. Chung, "Associative Knowledge Graph Using Fuzzy Clustering and Min-Max Normalization in

Video Contents," *IEEE Access*, vol. 9, pp. 74802–74816, 2021, doi: 10.1109/ACCESS.2021.3080180.

[17] A. Masitha and M. Kunta Biddinika, "KLIK: Kajian Ilmiah Informatika dan Komputer Preparing Dual Data Normalization for KNN Classfication in Prediction of Heart Failure," *Media Online*, vol. 4, no. 3, pp. 1227–1234, 2023, doi: 10.30865/klik.v4i3.1382.

[18] P. Musa, F. Al Rafi, and M. Lamsani, "A review: Contrast-limited adaptive histogram equalization (CLAHE) methods to help the application of face recognition," *Proc. 3rd Int. Conf. Informatics Comput. ICIC 2018*, no. October, pp. 1–6, 2018, doi: 10.1109/IAC.2018.8780492.

[19] C. Shorten, T. M. Khoshgoftaar, and B. Furht, *Text Data Augmentation for Deep Learning*, vol. 8, no. 1. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00492-0.

[20] Q. Zhu, H. Zhuang, M. Zhao, S. Xu, and R. Meng, "A study on expression recognition based on improved mobilenetV2 network," *Sci. Rep.*, vol. 14, no. 1, pp. 1–11, 2024, doi: 10.1038/s41598-024-58736-x.

[21] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," pp. 1–56, 2020, [Online]. Available: http://arxiv.org/abs/2003.05689

[22] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "The Effect of Adaptive Learning Rate on the Accuracy of Neural Networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 736–751, 2021, doi: 10.14569/IJACSA.2021.0120885.

[23] G. Geetharamani and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, vol. 76, pp. 323–338, 2019, doi: 10.1016/j.compeleceng.2019.04.011.

[24] S. McCandlish, J. Kaplan, D. Amodei, and O. D. Team, "An Empirical Model of Large-Batch Training," 2018, [Online]. Available: http://arxiv.org/abs/1812.06162

[25] G. Jain, D. Mittal, D. Thakur, and M. K. Mittal, "A deep learning approach to detect Covid-19 coronavirus with X-Ray images," *Biocybern. Biomed. Eng.*, vol. 40, no. 4, pp. 1391–1405, 2020, doi: 10.1016/j.bbe.2020.08.008.

[26] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci. Rep.*, vol. 12, no. 1, pp. 1–9, 2022, doi: 10.1038/s41598-022-09954-8.

[27] B. Jang, M. Kim, G. Harerimana, S. U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Appl. Sci.*, vol. 10, no. 17, 2020, doi: 10.3390/app10175841.

[28] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756