International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Implementation of CNN in Cat Breed Classification Using MobileNet V3

Gani Khalid Kawaskitan[1], Yulisdin Mukhlis[2], Emy Haryatmi[3*], Tri Agus Riyadi[4]
[1,2,3]Master of Electrical Engineering, Gunadarma University, Depok, West Java, Indonesia
[4]Informatics, Gunadarma University, Depok, West Java, Indonesia

Abstract**—***MobileNet has become popular in the machine learning and computer vision communities due to its compatibility with mobile devices and IoT (Internet of Things) devices. The choice of this method in classifying various types of cats such as Turkish Angora, Persian, Bengal, Spynx , and many more is very efficient because the MobileNet model can be implemented well on mobile devices or devices with limited resources. Apart from that, up to now there are 33 purebred cats in the world, which may increase again. By conducting this research, the author tested the MobileNetV3 architectural model to classify purebred cats into 3 cat breeds, namely Turkish Angora, Persian and Domestic Short Hair. This technique consists of 4 main stages, first importing data that has been trained from Kagle. Second, model formation and system training, the last is system testing. In this study, 900 images of the 3 cat breeds were used as testing data. Of the 900 cat images used in the training dataset, Kaggle data was taken. This model produces accuracy, precision, recall, and f1-score with values of 64%, 67%, 64%, and 63%.*

Keywords**—** *CNN, MobileNet V3, Dataset, Cat, Image Processing*.

I. INTRODUCTION

Research conducted by Arum TiaraSari and Emy Haryatmi on detecting images of dry corn kernels using CNN [1]. This research aims to detect images of dry corn kernels by applying the deep Convolutional Neural Network (CNN) method. learning , with results By using 7 convolutional layers, the resulting accuracy value ranges from 80% -100%. The average accuracy value resulting from testing data is 0.90296. The use of convolutional layers is able to detect the strength of the shape of an image.

Research conducted by Nur Azizah Ayunda, Emy Haryatmi , and Tri Agus Riyadi in classifying tomato ripeness using CNN [2]. This research aims to sort using the Convolutional Neural Network (CNN) method to classify tomato ripeness, with the results showing that by testing 10 tomato images, the results obtained were raw tomatoes approaching 90%, ripe tomatoes approaching 90%, and half-ripe tomatoes approaching 80%.

Research conducted by Yuni Naomi Yenusi, Suryasatriya Trihandaru , and Adi Setiawan used CNN to classify the faces of Papuans and other ethnicities [3]. This study aims to compare between CNN architecture models in classifying Papuan ethnic faces and other ethnic faces to see the model with the best accuracy. The selected CNN models are VGG16, VGG-19, ResNet-50 and MobileNet v1 and Mobilenet v2, with the result that MobileNet V1 is the best model. This model produces accuracy, precision , recall , and f1-score with values of 95%, 99%, 91%, and 94%.

Research conducted by al, Gulshan et classify complications of diabetes in the eye using CNN [4]. This study aims to classify DR using transfer learning to present the best CNN model. This review can also help future researchers in selecting a CNN model. Showing the differences in terms of performance with the models used are InceptionV3, AlexNet , VGG16, VGG19, VGG-S, Xception , DenseNet , ResNet , and GoogleNet , with the results of each model having the ability to read DR results.

This research was conducted by Dyah Ajeng Pramudhita, Fatima Azzahra, Ikrar Khaera Arfat, Rita Magdalena and Sofia Saidah to classify diseases in strawberries. [5]This study aims to implement CNN algorithms using MobileNetV3-Large and EfficientNet-B0 models to train a pre-processed four-class classification dataset, such as healthy leaves, leaf mite pest spiders, leaf pest caterpillars, and powdery mildew leaves. Using this architecture helps keep the parameters and model size small. The results of this study show that the MobileNetV3-Large model outperforms EfficientNet-B0. The results obtained with the best accuracy reached 92.14% using the MobileNetV3-Large architecture with hyperparameters optimizer RMSProp , epoch 70, and learning rate 0.0001.

This research was conducted by Andri Heru Saputra and Dhomas Hatta to detect images of vegetable plants based on mobile. android in real time [6]This research aims to build a deep learning based mobile phone model. learning to detect vegetable plant objects in real- time such as bok choy , spinach, kale, and curly kale to determine whether the vegetables are ready to be harvested. The cellular-based architecture was chosen for reasons of latency, privacy, connectivity, and power consumption. using MobileNetV3 as its base architecture to find the best model, has obtained a maximum MAP score of 0.705510 using 36,000 image datasets . Detection can be done with an ideal distance of 5 cm and 10 cm.

II. METHODOLOGY

In this study, a cat breed classification system was designed to determine its accuracy using a digital image processing method that has been tested on a dataset. Kaggle consists of four stages. Figure 1 shows a block diagram of the system designed in this study. In general, the block diagram of a structured system consists of the stages of collecting cat breed datasets , creating a CNN model, training, and testing.

2.1. Dataset

Gani Khalid Kawaskitan, Yulisdin Mukhlis, Emy Haryatmi, and Tri Agus Riyadi, “Implementation of CNN In Cat Breed Classification Using MobileNet V3,” *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 4, pp. 23-28, 2025.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

In this study, the dataset used is a data set (Dataset) downloaded from the Kaggle site containing a collection of cat breed data (. csv) and unbalanced images representing 67 cat breeds totaling 9000 images. The interpretation of this Cat Breed has been selected by the community and not by professionals, there is a possibility of errors in some of the training data results, then the researcher selected 900 images based on 3 cat breeds, namely Angora, Persian , and Domestic.

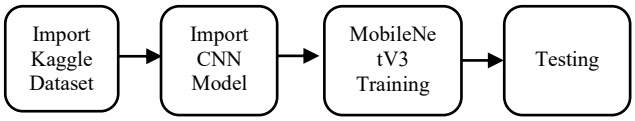

Fig. 1. Method

2.2. CNN Model

MobileNet architecture is based on the concept of depthwise separable convolution , which aims to reduce the number of parameters and computation while maintaining accuracy and speeding up inference on mobile devices or devices with limited resources, and the following is a comparison table for the MobileNetV3 CNN model.

TABLE 1. MobileNet V3 Model-Large

MobileNet V3 Large						
Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	16	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	64	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	120	40	v	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	v	RE	1
$28^2 \times 40$	bneck, 5x5	240	40	v	RE	1
$28^2 \times 40$	bneck, 3x3	200	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	672	112	v	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	v	HS	1
$14^2 \times 112$	bneck, 5x5	960	160	v	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	v	HS	1
$7^2 \times 160$	bneck, 5x5	-	160	v	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

In general, the MobileNetV3 Architecture model is divided into 2 types, namely MobileNetV3-Large and MobileNetV3-Small, where the difference lies in the number and type of layers used (A. Howardet al ., 2019). Table 1 shows a total of 20 steps for the MobileNetV3 – Large model , and Table 2 shows a total of 16 steps for the MobileNetV3 – Small model . SE stands for Squeeze-And-Excite, a block in a neural network that calculates the relationship between channels to improve performance.

TABLE 2. MobileNet V3 Model – Small

MobileNet V3 Small						
Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	v	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	v	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	v	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	v	HS	1
$14^2 \times 40$	bneck, 5x5	144	48	v	HS	1
$14^2 \times 48$	bneck, 5x5	288	48	v	HS	1
$14^2 \times 48$	bneck, 5x5	576	96	v	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	v	HS	1
$7^2 \times 96$	bneck, 5x5	-	96	v	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	-	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Exp size is an abbreviation of expansion size for the expansion size in the block, #out is the number of filters used in the convolutional layer. NL stands for nonlinearity as an activation function used in neural network layers. Then, HS is an abbreviation for activation hard-swish is an activation function which is a variant of Swish and is used to improve computational efficiency and RE is an abbreviation for activation Rectified Linear Unit (ReLU) is one of the most commonly used activation functions in neural networks. NBN is an abbreviation for no batch normalization shows that the batch layer normalization is not used, and s stands for stride which is a parameter in the convolution layer that determines how much the filter is shifted during the convolution operation.

Figure 2 shows an explanation of the MobileNetV3 architecture. The first stage is the image input process with 2D convolution, 3x3 kernel , stride 2, and activation function hard-swish on a 224x224 image with 3 color channels , namely red , green , and blue . Then the next convolution process is carried out with a 3x3 bottleneck where the convolution process begins with a 1x1 convolution on the input to reduce the depth then the 3x3 convolution can be lighter to process the convolution feature because the depth is lower and the 1x1 convolution for the output so that the depth can return to the input , then using ReLU as an activation function . The main purpose of the bottleneck process is to

Gani Khalid Kawaskitan, Yulisdin Mukhlis, Emy Haryatmi, and Tri Agus Riyadi, “Implementation of CNN In Cat Breed Classification Using MobileNet V3,” *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 4, pp. 23-28, 2025.

increase model efficiency. Then the next layer is the same bottleneck function 11 times for MobileNetV3-Small and 15 times for MobileNetV3-Large, in each layer there are only differences in the stride parameter, activation function and kernel size. Then there is a 7x7 pooling process which functions to reduce the dimension of the feature maps , which reduces the number of learnable parameters and reduces the amount of computation performed on the mobile. After that, there is a 1x1 kernel convolution and uses no. batch normalization and activation using hard-swish. In the final layer, there is the same layer but its usage is different because this layer is used for image output . The following is the total parameters of the MobileNetV3 architecture model.

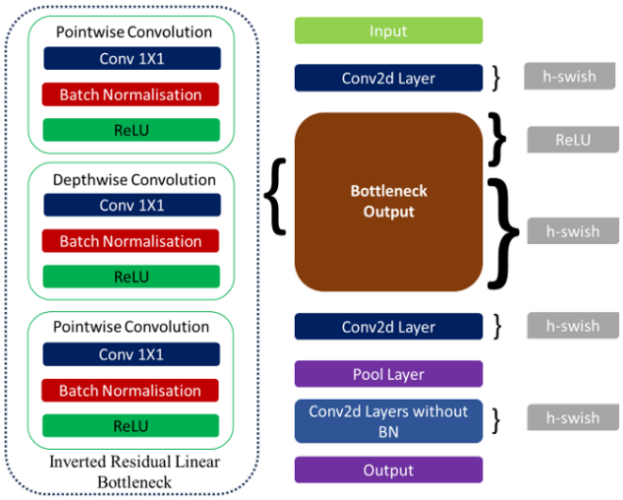

Fig. 2. MobileNetV3 Architecture Details
Source: journal https://doi.org/10.56294/dm2023153

TABLE 3. Model Parameters of MobileNet V3

Total Params:	939120 (3.58 MB)
Trainable params:	0 (0.00 Byte)
Non-Trainable params:	939120 (3.58 MB)

Table 3 shows the number of parameters trained on this model is 939,120. With a Total Trainable value of 0, the trainable value indicates that the model has been previously trained, with the image size in the final convolution process before entering the fully connected layer being 224x224 pixels.

2.3. Training

After preparing the MobileNetV3 CNN Architecture model, the next step was to add a sequential model for testing the MobileNetV3 architecture with a new model, namely sequential, which the researcher prepared based on references from previous research.

TABLE 4. Addition of new sequential models

Layer (type)	Output Shape	Param #
MobilenetV3small (Function al)	(None, 7, 7, 576)	939120
dense (Dense)	(None, 7, 7, 128)	73856
dropout (Dropout)	(None, 7, 7, 128)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387

In table 4, you can see that the sequential model adds several layers, including: Dropout functions to avoid overlapping, dense is used by the model to perform global average classification. polling to get the best pooling value .

TABLE 5. Total parameters for the new sequential model

Total Params:	1013363 (3.87 MB)
Trainable params:	74243 (290.01 KB)
Non-Trainable params:	939120 (3.58 MB)

In Table 5, it can be concluded that for the previous architecture model , the number of parameters trained on this model was 939,120 parameters. After the new sequential model was added , it changed to 1,013,363 parameters. The trainable param value which was previously 0 has now changed to 74,243 parameters, which will then be tested using Keras with epoch 100.

2.3. Testing

In the training process , there is a total of 9000 data, the dataset division process is training data and testing data. The ratio of training data and testing data is 50:50. Consisting of 600 training data and 600 testing data. The testing data is 600 data consisting of 3 groups of images of 200 Angora cat breeds, 200 Persian cat breeds, and 200 domestic cat breeds. The training data is 600 data consisting of 200 images of Angora cat breeds, 200 images of Persian cat breeds , and 200 images of domestic cat breeds.

In the separation process, it will call the directory A dataset containing Angora cat breeds, Persian cat breeds , and domestic cat breeds . If the training data is able to learn these data well, then there is a possibility that the training data will also be able to learn the new data, namely the testing data. However, if the training data is unable to learn the data well, then the training data will not be able to learn the testing data, which will result in an InvalidArgumentError . In machine learning, learning includes the terms epoch and batch size when training . In this training, epoch = 100, batch size = 128 were used.

TABLE 6. Steps of training epoch 100

epoch	loss:	val_loss:	acc:	val_acc:
1/100	1	1,1311	0,3357	0,3333
2/100	1	1,1	0,3286	0,3278
3/100	1,101	1,1018	0,3905	0,3333
4/100	1,094	1,0986	0,3595	0,3333
5/100	1,1	1,0919	0,3476	0,3389

Comparison of loss and accuracy values from training data can be seen in table 6. In this table is a snippet of part of the CNN training process , in the results of epoch 1/100 for the loss value is 1 even more than 1 up to epoch 45 if seen in the training table which means 100% error. Accuracy is 0.3357 which means 33%, val_loss value is 1 which means 100%, val_accuracy value is 0.3333 which means 33%, this shows the results for 100% prediction will be wrong if training is stopped at this stage up to stage 45 which can be seen in table 7.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

TABLE 7. Continued Steps of epoch 100 training

epoch	loss:	val_loss:	acc:	val_acc:
41/100	1,004	1,008	0,5357	0,55
42/100	0,99	1,0011	0,5929	0,5833
43/100	0,977	1,0018	0,6595	0,5333
44/100	1,001	1,0065	0,5452	0,5944
45/100	0,978	0,9978	0,5857	0,5111

Meanwhile, in the continuation table for epochs 96 to 100, which can be seen in table 8.

TABLE 8. Epoch 100 training steps

epoch	loss:	val_loss:	acc:	val_acc:
96/100	0,865	0,8942	0,6548	0,6222
97/100	0,862	0,886	0,6524	0,5778
98/100	0,869	0,8839	0,6405	0,6444
99/100	0,876	0,8983	0,631	0,5833
100/100	0,853	0,8824	0,6548	0,6444

Loss and accuracy values of the training data have improved with the results from epoch 96/100 loss being 0.865, which means above 86.5%, this error value will continue to decrease until the lowest value is at epoch 100/100, which is 0.853 or 85.3%. The highest accuracy value is 0.6548, which means 65% at epoch 96. The highest val_loss value is 1, which means 100%. The val_accuracy value is 0.6444, which means 64%. At this step, the value has reach good average value from loss and accuracy values are marked with mark from the validation that approaches same , if continue training Eat mark will tend The same for Overview of results can be seen in figure 3 is chart from model training with epoch 100.

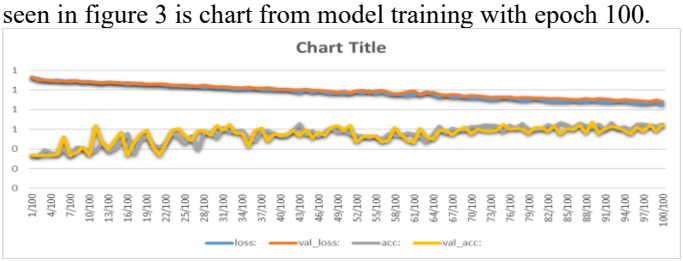

Fig. 3. Model Training Chart Graph

III. Result and Discussion

Accuracy and loss with model plots can be seen in Figure 4 and Figure 5. showing the accuracy and loss graphs of the model from the train data and validation data.

Training and validation data were tested to predict the lowest target error until there was no error or close to 0. The loss value generated from the model on the training data was 0.8532. The loss value generated from the model on the validation data was 0.8532. The loss value is 0.8824. These two values are the lowest and indicate that the resulting loss function is good.

Figure 5 shows the training and validation accuracy. The model predicted correctly for all data. The training and validation data were tested to classify into the correct class until there were no errors or close to 1. The highest accuracy values were 0.6722 for the training data and 0.6714 for the testing data. This indicates that the model can correctly classify with an accuracy of 67%.

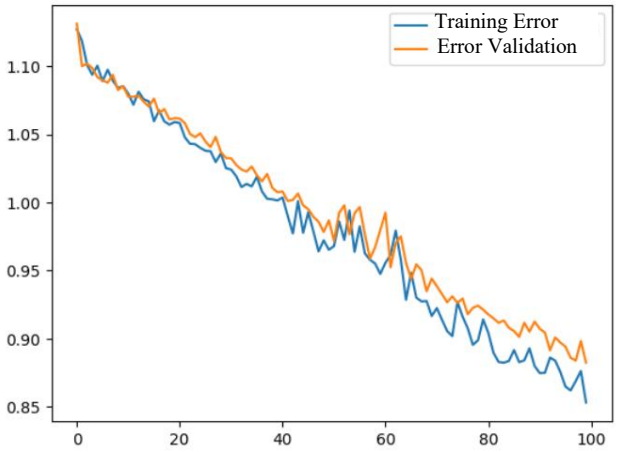

Fig. 4. Training Graph Loss

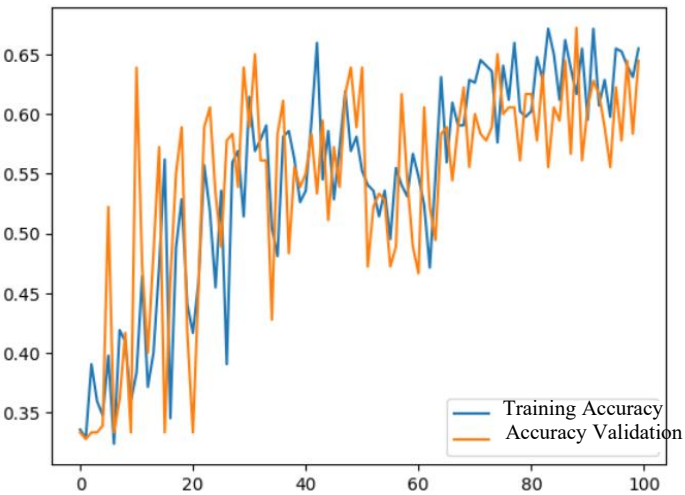

Fig. 5. Chart Training Accuracy

Based on both parameters, loss and accuracy, the model can correctly classify tomato types. As the epoch value increases , the loss value approaches zero or less than one, and the accuracy value increases. This value indicates good loss and accuracy results. Low values are found in the training graph. loss indicates the minimum error. Accuracy value on the training graph high loss can read and differentiate the model classification in each class.

3.1. Normal Value Matrix

Confusion table the 3x3 matrix in Figure 6 is used in the performance evaluation of the classification model that has three classes as seen in Figure 8. Each cell in the table represents the number of instances that fall into a particular combination of the actual classes . class) and predicted class (predicted class).

Gani Khalid Kawaskitan, Yulisdin Mukhlis, Emy Haryatmi, and Tri Agus Riyadi, "Implementation of CNN In Cat Breed Classification Using MobileNet V3," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 4, pp. 23-28, 2025.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Fig. 6. 3x3 Matrix

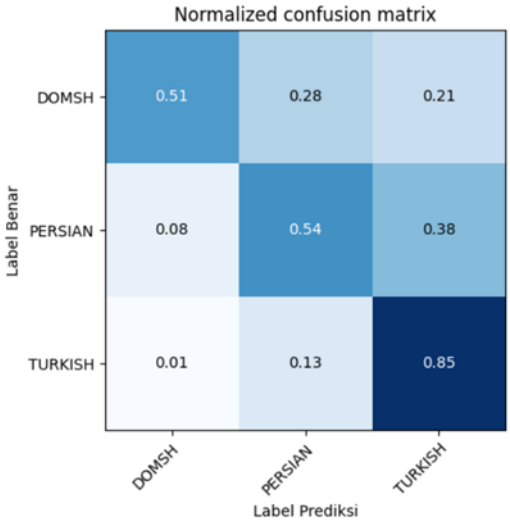

Fig. 7. Normalization of Training Matrix

Figure 7 shows how well our model performs in classifying and identifying specific classes. After obtaining the normalized confusion matrix, The next step in the matrix is to find the True value. Positive , True Negative , False Positive and False Negative, the following is the definition and formula for finding this value:

1) Accuracy

Definition: Measuring to what extent the classification model gives correct prediction in a way overall, considering amount correct and incorrect predictions.

Accuracy = Total True Prediction/Total Data

With confusion matrix (3x3):

$$\text{Accuracy} = \frac{a_{11}+a_{22}+a_{33}}{a_{11}+a_{12}+a_{13}+a_{21}+a_{22}+a_{23}+a_{31}+a_{32}+a_{33}}$$

2) Precision

Definition : Measuring how far the prediction is positive from the real model positive . This gives description of how much both models select the correct instance as positive .

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}$$

With confusion matrix (3x3):

$$\text{Precision}_1 = \frac{a_{11}}{a_{11}+\text{False Positives}_1}$$

$$\text{Precision}_1 = \frac{a_{11}}{a_{11}+a_{21}+a_{31}}$$

$$\text{Precision}_2 = \frac{a_{22}}{a_{22}+\text{False Positives}_2}$$

$$\text{Precision}_2 = \frac{a_{22}}{a_{12}+a_{32}}$$

$$\text{Precision}_3 = \frac{a_{33}}{a_{33}+\text{False Positives}_3}$$

$$\text{Precision}_3 = \frac{a_{33}}{a_{13}+a_{23}}$$

3) Recall

Definition: Measuring to what extent the model can find all true positive instances. This provides description of how good models can recognize all cases positively.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$$

With confusion matrix (3x3):

$$\text{Recall}_1 = \frac{a_{11}}{a_{11}+\text{False Negatives}_1} \quad \text{Recall}_2 = \frac{a_{22}}{a_{22}+\text{False Negatives}_2}$$

$$\text{Recall}_1 = \frac{a_{11}}{a_{11}+a_{12}+a_{13}} \quad \text{Recall}_2 = \frac{a_{22}}{a_{21}+a_{22}+a_{23}}$$

$$\text{Recall}_3 = \frac{a_{33}}{a_{33}+\text{False Negatives}_3}$$

$$\text{Recall}_3 = \frac{a_{33}}{a_{31}+a_{32}+a_{33}}$$

4) F1 – Score

Definition : Is the harmonic mean of Precision and Recall. F1-Score provides balance between Precision and Recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$$

With confusion matrix (3x3):

$$\text{F1-Score}_1 = 2 \times \frac{\frac{a_{11}}{a_{11}+a_{21}+a_{31}} \times \frac{a_{11}}{a_{11}+a_{12}+a_{13}}}{\frac{a_{11}}{a_{11}+a_{21}+a_{31}} + \frac{a_{11}}{a_{11}+a_{12}+a_{13}}}$$

$$\text{F1-Score}_2 = 2 \times \frac{\frac{a_{22}}{a_{12}+a_{32}} \times \frac{a_{22}}{a_{21}+a_{22}+a_{23}}}{\frac{a_{22}}{a_{12}+a_{32}} + \frac{a_{22}}{a_{21}+a_{22}+a_{23}}}$$

$$\text{F1-Score}_3 = 2 \times \frac{\frac{a_{33}}{a_{13}+a_{23}} \times \frac{a_{33}}{a_{31}+a_{32}+a_{33}}}{\frac{a_{33}}{a_{13}+a_{23}} + \frac{a_{33}}{a_{31}+a_{32}+a_{33}}}$$

5) Results

Accuracy, Precision, Recall, and F1-Score are metric general evaluation used in context classification for measure performance from a model or system, as an illustration due to there are 3 models then made table calculation for 3 models where race 1 represents race Domestic Short Hair cat, breed 2 represents race Persian cats, and the 3 breeds represent race Angora cat. in the calculation above can be concluded in Table 7,

TABLE 9. Results of Accuracy, Precision, Recall, and F1-Score

	Ras 1	Ras 2	Ras 3	Rata"
Akurasi	64%	64%	64%	64%
Precision	85%	57%	59%	67%
Recall	51%	54%	86%	64%
F1-Score	64%	55%	70%	63%

In Table 9 it can be concluded For mark Accuracy from each race is 64% which means the average of the new CNN Model can predict 64% correctness , Precision for each model is the highest is introduction Correct For race 1 and average for Precision value is 67%, recall or model capabilities in recognize mark true throughout matrix results for mark highest there is in race 3 with an average of 64%, and for F1-Score value which is Middle value of highest Precision and Recall there is in race 3 with value of 70% while for the average of all models 63%.

3.2. Observation of Learning Outcomes

Observation data obtained from program testing results learning, program results learning assessed with picture random No selected order from picture with size 244x 244 and following sample example for one image test random.

Gani Khalid Kawaskitan, Yulisdin Mukhlis, Emy Haryatmi, and Tri Agus Riyadi, “Implementation of CNN In Cat Breed Classification Using MobileNet V3,” *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 4, pp. 23-28, 2025.

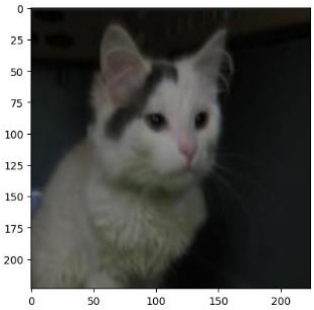

Fig. 8. Random Image Sample

From the explanation in figure 8 will tested as many as 15 random samples where the samples represent from each race, 5 random samples for domestic short hair race, 5 random samples for race persian, and 5 random samples for angora breed, and the following the explanation is in table 10.

TABLE 10. 3-race test table random

no	Nilai random		Ras Gambar	kecepatan (ms/step)	nilai akurasi yang di prediksi			Prediksi CNN	Hasil
	n	Gambar			DomSH	Persian	Angora		
1	23		DOMSH	33	**0,4**	0,34	0,26	DOMSH	BENAR
2	54		DOMSH	40	0,25	0,36	**0,4**	ANGORA	SALAH
3	76		DOMSH	22	0,37	**0,38**	0,25	PERSIAN	SALAH
4	34		DOMSH	41	**0,39**	0,33	0,28	DOMSH	BENAR
5	89		DOMSH	38	0,21	**0,62**	0,18	PERSIAN	SALAH
6	257		PERSIAN	40	0,17	0,37	**0,46**	ANGORA	SALAH
7	234		PERSIAN	23	0,33	**0,35**	0,32	PERSIAN	BENAR
8	289		PERSIAN	46	0,24	**0,44**	0,32	PERSIAN	BENAR
9	356		PERSIAN	60	0,28	**0,39**	0,33	PERSIAN	BENAR
10	387		PERSIAN	41	0,22	**0,46**	0,32	PERSIAN	BENAR
11	465		ANGORA	55	0,23	**0,45**	0,32	PERSIAN	SALAH
12	476		ANGORA	52	0,14	0,37	**0,49**	ANGORA	BENAR
13	568		ANGORA	59	0,19	0,31	**0,5**	ANGORA	BENAR
14	523		ANGORA	105	0,06	0,19	**0,74**	ANGORA	BENAR
15	589		ANGORA	40	0,14	0,33	**0,53**	ANGORA	BENAR

How to read table 10, in the form of every One image random input to in the train data, it will be scanned with a new CNN model. including to in race which image is thin random the from each scan has results accuracy as example in table 9 for test to 1 image entered is image cat DOMSH race with speed reading the model is 33 ms /step and the result accuracy For The DOMSH model estimate is 0.4, the Persian model 0.34, the Angora model 0.26, from model estimates will be taken estimation the biggest namely the next DOMSH model will display results Correct Because predictions in the results by the new model in accordance with the random model being tested. In table 10 it can be seen that tests 2, 3, 5, 6, and 11 show results wrong prediction due to mark highest prediction there is in the wrong race thing This Can happen because at the time CNN model training only get the average accuracy value is 64% which means there is a 36% error rate from the new model This is for summary from prediction For each model can be seen in table 11.

From table 11 it can be concluded for cnn model new Accuracy level best in race3 or TURKISH ANGGORA with average value 56.50 %, if seen in the normal matrix value highest also in race3 with mark of 0.85.

TABLE 11. Random test samples

Uji 5 sample setiap ras			
Penilaian	DomSH	Persian	Turkish
TRUE	2	4	4
FALSE	3	1	1
Keberhasilan	40%	80%	80%
Kecepatan rata”	34,8	42	62,2
%Benar Prediksi	39,50%	41%	56,50%
%Salah Prediksi	27,66%	37%	32%

IV. CONCLUSION

Based on research that has been done so it can conclude that Design and Testing, obtained:

A. Introduction of object digital image with use Convolutional Neural Network (CNN) algorithm with MobileNetV3 architecture model for classify image cat has successfully done, besides That in study This Still Lots of Error Good from creation of new models and selection of datasets that are still A little.

B. With using the MobileNetsV3 model. The proposed method own accuracy by 64%, which was obtained from score training is 67%, validation by 64%, and F1-Score 63%.

REFERENCES

[1] A. TiaraSari and E. Haryatmi, “Application of Convolutional Neural Network Deep Learning in Seed Image Detection Dry Corn,” *RESTI Journal (Engineering) Information Systems and Technology)* , vol. 5, no. 2, pp. 265–271, Apr. 2021, doi : 10.29207/ resti.v 5i2.3040.

[2] NA Ayunda, E. Haryatmi, and TA Riyadi , "Classification of Tomato Ripeness Based on Convolutional Neural Network Methods," *Journal of Information Systems and Informatics* , vol. 5, no. 4, pp. 1658–1675, Dec. 2023, doi : 10.51519/ journalisi.v 5i4.613.

[3] YN Yenusi , Suryasatriya Trihandaru , and A. Setiawan, "Comparison of Convolutional Neural Network (CNN) Models in Face Classification of Papuan and Other Ethnicities," *JST (Journal of Science and Technology)* , vol. 12, no. 1, March. 2023, doi : 10.23887/ jstundiksha.v 12i1.46861.

[4] I. Kandel and M. Castelli, "Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review," Mar. 01, 2020, *MDPI AG* . doi : 10.3390/app10062021.

[5] DA Pramudhita , F. Azzahra, K. Arfat, R. Magdalena, and S. Saidah, "Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture ARTICLE INFO ABSTRACT," *Journal Journal of Electrical Engineering, Computer and Informatics (JITEKI)* , vol. 9, no. 3, pp. 522–534, 2023, doi : 10.26555/ jiteki.v 9i3.26341.

[6] Andri Heru Saputra and Dhomas Hatta Fudholi, “Realtime Object Detection Ready Time Harvest Android Mobile- Based Vegetable Plants with Deep Learning,” *Jurnal RESTI (Engineering) Information Systems and Technology)* , vol. 5, no. 4, pp. 647–655, Aug. 2021, doi : 10.29207/ resti.v 5i4.3190.