Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 4 (2022) 639 - 647 ISSN Media Electronic: 2580-0760

Mask Detection Using Convolutional Neural Network Algorithm

Rizky Amalia[1], Febriyanti Panjaitan[2]
[1,2]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Darma
[1]19142037P@student.binadarma.ac.id, [2]febriyanti_panjaitan@binadarma.ac.id*

Abstract

The World Health Organizations and the Ministry of Health of the Republic of Indonesia have required the use of masks to suppress the spread of COVID-19. WHO provides guidance on how to use a good mask to cover the mouth and nose. This study aims to detect the correct use of masks using the Convolutional Neural Network. CNN is a popular Deep Learning algorithm for image data classification problems. The Mask Usage Detector is built with the help of a pre-trained MobileNetV2 model with an architecture that supports media that has minimum computations. This study will also compare the performance of three optimization methods from CNN, namely Adam, SGD, and RMSprop in detecting the use of masks. Performance will be seen from the test results by analyzing the values of accuracy, precision, and recall. The dataset used is in the form of image data of 2,029 images for 2 categories, namely "masked" and "unmasked". A total of 1,623 images were used as training data and 406 images for test data. Based on the testing process, the accuracy of each optimization is 93.84% with Adam optimization, 84.48% with SGD optimization, and 93.10% with RMSprop optimization. With the proposed model, this study obtains the performance results of the three CNN optimizations, and it is concluded that adam's optimization gives better performance results than the other two optimizations.

Keywords: *Mask detection, Convolutional Neural Network, MobileNetV2, CNN optimizations*

1. Introduction

In 2019, the world began to be hit by the Coronavirus pandemic [1]. Since then, various ways have been done to minimize the spread of COVID-19. One way that is required by the government is to use a mask [2]. The World Health Organization (WHO) also provides advice on the use of masks as part of prevention and control efforts to minimize the spread of the COVID-19 virus. In its guidelines, WHO also recommends wearing a good mask by covering the face, especially the mouth and nose [3].

Many public locations require visitors to wear masks. However, there are still many who ignore these regulations. This condition also sometimes often occurs in workers who work indoors or in offices, even though the spread of the COVID-19 virus can occur not only through splashes when sneezing or coughing, but can also occur when we talk, both indoors and outdoors.

After the surge of the COVID-19 virus, many types of masks were issued both from the UMKM industry and other industries, these masks can be used to cover parts of the face, especially the nose and mouth, sometimes masks have their own advantages and disadvantages. There are about three types of masks that are now commonly used by some people, such as cloth or scuba masks, surgical or medical masks, and N-95 masks. With the various types of masks, sometimes they are not used according to the standards set by WHO, the standard is to ensure that the mask covers the nose and mouth, which is adjusted to the bridge of the nose to minimize the gap between the face and the mask[4].

The development of technology such as Artificial Intelligence (AI) is one of the technologies that is increasingly being applied by several researchers in recent years [4]–[8] to detect or recognize objects in an image [9]. One of the methods of artificial intelligence technology is the Deep Learning (DL) method, which is part of the science of Machine Learning (ML) based on artificial neural networks (ANN). DL can classify or detect directly both image and sound. Convolutional Neural Network (CNN) is one part of DL which is designed to manage data in two-dimensional form. CNN can learn directly from data in the form of images so that it can reduce the burden of programming. CNN's ability is stated as the best method from several researchers in detecting and recognizing objects which usually consist of neurons that have weight, bias, and activation function [10].

Accepted: 09-07-2022 | Received in revised: 09-08-2022 | Published: 22-04-2022

CNN uses optimal value weights to give the minimum of error and maximum value for accuracy. CNN can also optimize the shape of the model into the most accurate form by utilizing its weight [11]. There are several optimizations used in the CNN method, namely Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSprop), and Adam. Each of these optimizations has advantages so that it can provide the best results.

Research conducted by Nyoman Purnama, et al [4] performed mask detection using CNN with two optimizations, namely Adam and Gradient Descent. Based on the test results using the MobileNetV2 pre-training model, it produces an accuracy value of 90% using CNN optimization, namely Adam, and 80% from Gradient Descent optimization.

Another study by Budiman et al[5], tested a mask detection model made using CNN with a web-based MobileNetV2 architecture. Based on the research, the results obtained an average accuracy of 88.53% for the mask detection model.

Other studies using CNN have also been conducted by Hairong Wang, et al. This study aims to apply image recognition to investigative audits. This study uses three CNN architectures namely MobileNetV2, ResNet18, and ShuffleNetV2 and compares three optimizations, namely SGD, Adam and RAdam. Based on the research results, the highest accuracy results for the MobileNetV2 architecture are 54.14% for SGD optimization, 65.4% for Adam optimization, and 75.23% for RAdam optimization [6].

Another study by Abraham L, et al [7]. In this study, a model for the classification of agricultural land habitat classification in Ireland with the architecture of VGG16, ResNet34, MobileNetV2, and Stacked Ensemble was tested using four optimizations, namely Adam, RMSprop, SGD, Adam. Based on this research, the best accuracy results for the MobileNetV2 architecture are models with RMSprop optimization, which is 75.12%.

Research by Ashwaq Alsayed, et al [8], they tested the leaf disease detection model on apple trees using four pre-training models namely VGG16, ResNetV2, InceptionV3, and MobileNetV2 with four different optimizers, namely SGD, Adam, AdaDelta, and RMSprop. From the results of the study, the model was made using ResNetV2 with Adam's optimization getting the highest accuracy results compared to the combination of pre-training models and other optimizations, which was 94.7%.

Based on the background, contributions of this study is: (1) Using datasets from two different sources, namely through open access Kaggle and real-time data from employees of Regional Office VII of the Palembang State Civil Service Agency (BKN); (2) developing a detection model using the CNN method of the MobileNetV2 architecture with three optimizations provided, namely Adam, SGD, and RMSprop; (3) Provide the performance of three optimizations of CNN using two different datasets.

2. Research Methods

The research method used in this study is a combination of qualitative methods to obtain a system design and quantitative methods through experiments to test how effective the system is.

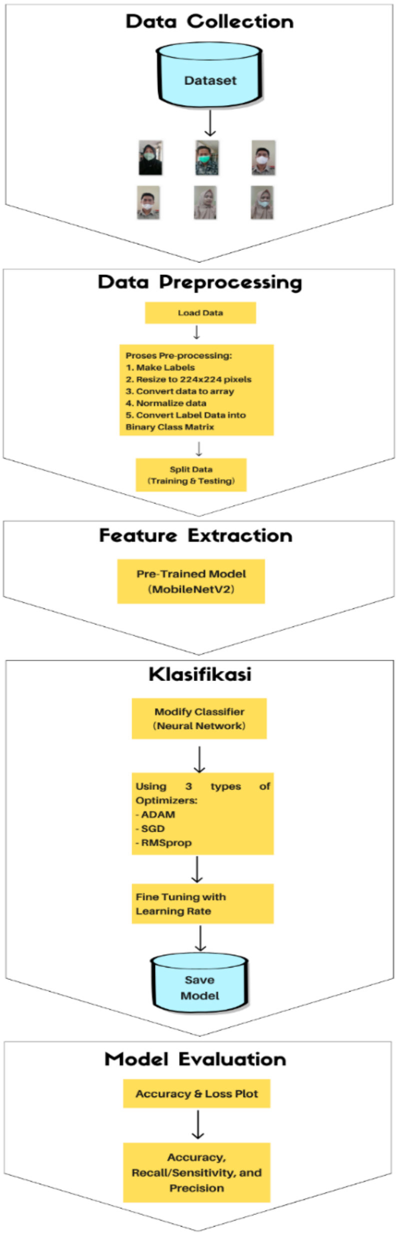

Figure 1. Research Flow

2.1 Data Collection

The data used for this study uses data consisting of a collection of images that have been separated in separate folders between images of people wearing masks and not wearing masks. This folder is also a category for classes, namely “Masked” and “Unmasked”. This study uses 2 dataset sources, the first is from an open access dataset, namely Kaggle [12], the second is from images of people wearing masks and not wearing masks from employees of Regional Office VII BKN Palembang. Figure 2 is an image of Kaggle, while Figure 3 is an image of an employee of Regional Office VII BKN Palembang.

Figure 2. Datasets images from Kaggle

Figure 3. Datasets images from employees of Regional Office VII BKN Palembang

2.2 Data Preprocessing

Data preprocessing is an activity to process and analyze data with several stages. First, resize the image to a size of 224x224 pixels according to the provisions of MobileNetV2 [13], [14].

Then convert the dataset label into a binary class matrix. First the image will be converted in the form of an array containing the pixel values in the image. Because the range of pixel values in the image is too far (between 0-255), then scaling is done to get the pixel values that previously ranged from 0 to 255, now range from -1 to 1. After scaling the pixel values, then the results of the scaling are converted in decimal form so that later it can be converted into a binary class matrix. After becoming a binary class matrix, the image features are ready to be processed and included in the feature extractor as the main ingredient for the process of making the mask detection model. Figure 4 is the result of converting the image into a binary class matrix array.

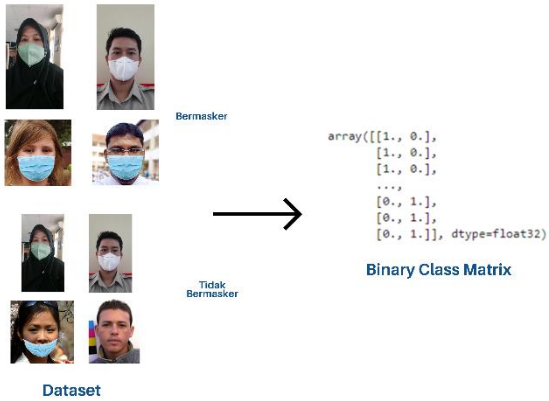

Figure 4. Image converted into binary class matrix

Then, the distribution of the dataset uses the technique of split validation, where this technique divides the data into two parts based on the distribution of data consisting of training data and test data. This technique has been used by Lasniari, et al [15] and Putri Annisa, et al [16] in detecting or classifying an image with a ratio of 80% training data and 20% test data and produces an average accuracy of above 90%. The training data will be used to recognize image patterns and create models, while the test data will be used to test and evaluate the models that have been made. The ratio of training data and test data in this study was 80% for training data and 20% for test data.

Augmentation of the research data is to use a technique to increase the variation of the dataset by applying a random but realistic transformation. This activity is to rotate, zoom, shift the image and flip the image [17]. Figure 5 is one of the results of augmenting the Kaggle dataset numbered 1005.

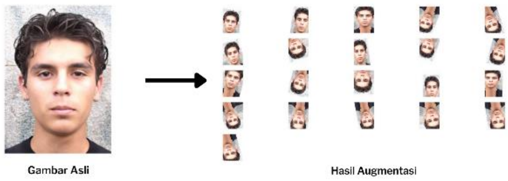

Figure 5. Augmentation Dataset number 1005

2.3 Feature Extraction

Feature Extraction in this study uses existing features on MobileNetV2. MobileNetV2 is an architecture that exists on CNN in overcoming large amounts of data computing [18]. The feature used is a linear bottleneck which contains several layers of convolution depthwise and pointwise and a shortcut connection between bottlenecks. Figure 6 is an overview or business model of the architecture of MobileNetV2 that uses Linear Bottlenecks and shortcut connections between bottlenecks.

DOI: https://doi.org/10.29207/resti.v6i4.4276

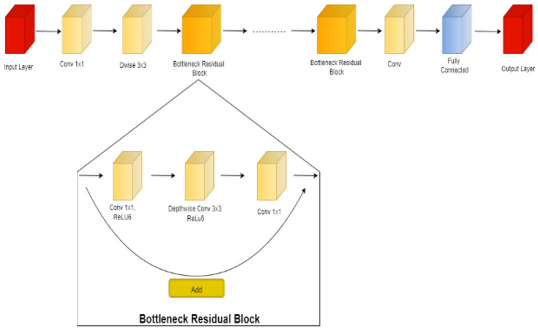

Figure 6. MobileNetV2 Architecture

The layers in the MobileNetV2 architecture perform feature extraction on the input image layer as well as to reduce the dimensions of the image, so that it can be easier and more efficient in the next step, namely classification with Fully Connected Layer.

2.4 Classification

CNN is a type of Neural Network that is most suitable for the classification process on image data [9]. CNN consists of two main parts, namely the feature extraction section and the classification section. In the feature extraction section, there are several processes, namely Convolutional layer, activation function and pooling. The results of the feature extraction process are then carried out with a classification process with a fully connected layer to produce detection results [19], [20]. In figure 7 is the architecture MobileNetV2.

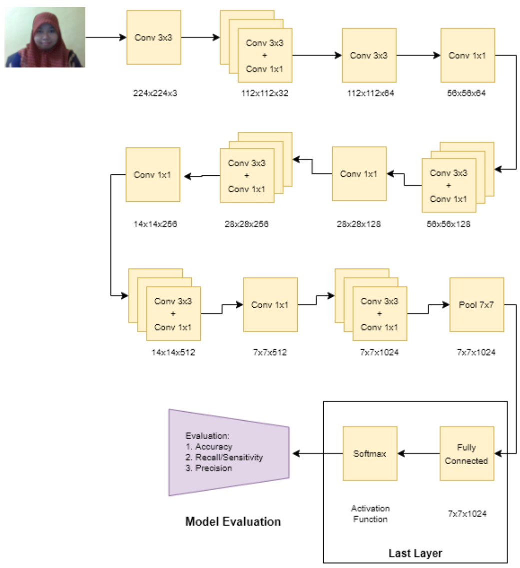

Figure 7. Architecture MobileNetV2

In the training process, an error is often found which is called a loss. To measure the loss obtained in each epoch, a loss function is needed. The loss function used in this study is Binary Cross Entropy. This loss function is used because this research makes a model with two classes. Binary Cross Entropy measures loss by calculating the target probability with the log of the prediction [21].

This study uses three optimizers on CNN. Optimizer is a technique for optimizing the weight value in the backpropagation process in the Neural Network with the aim of getting the classification results with the smallest possible error or loss value [22]. The three optimizations used are Adam, SGD and RMSprop.

The result of the feature extraction is then converted into a vector form called Flattening. After flattening is done, then proceed with the classification process using a Fully Connected Layer with total epochs of 50. To reduce the risk of overfitting, one of the regularization techniques used is Dropout.

Dropout layer works by eliminating one or more neurons in the hidden layer randomly in Neural Network [23]. The dropout value used in this study is 0.5.

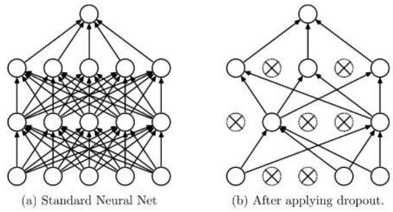

Figure 8. Illustration of Neural Network with Dropout [23]

2.5 Model Evaluation

Pattern recognition techniques and retrieval of information to evaluate the performance of the model to be developed using accuracy, precision, and recall. Accuracy is an illustration of how accurately a machine learning model can classify data correctly. Or the number of correctly classified data divided by the total number of data. Precision is an illustration, of all data that has been predicted as data with a positive label (masked), how many are actually masked. Recall is an illustration, of all the data that is truly positive (masked data), how many are actually predicted as masked data [24]. The form of accuracy, precision, and recall equations can be seen from the equations with numbers 1, 2, and 3.

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{1}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{2}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{3}$$

where TP is the amount of data with positive values and is predicted to be positive, then TN is the amount of data with negative values and predicted to be negative, then FP is the amount of data with negative values but predicted to be positive, and FN is the amount of data with positive values but predicted to be negative.

DOI: https://doi.org/10.29207/resti.v6i4.4276

2.6 Description and System Flow

The designed model is a model that is able to detect the use of masks. This model will be built using the CNN method and the MobileNetV2 architecture. Making the system begins by collecting datasets in the form of images of people wearing face masks. The image data is divided and categorized into two, namely “masked” and “unmasked”. This image dataset was obtained from the Kaggle platform [12] and a photo of employees of the Regional Office VII of the Palembang Civil Service Agency.

The mask detection model using the CNN method was created using Python with the help of Tensorflow and Keras, a framework used to create machine learning and deep learning models. While Keras is a Neural Network API that has the ability to run on Tensorflow [4].

After generating a classification model or detecting the use of masks, the next step is to carry out the testing process on the developed model. Then proceed with evaluating the performance of the model using accuracy, recall, and precision.

3. Results and Discussions

This study makes a classification model to detect images with two classes, namely "masked" and "unmasked". This model was created using the CNN method with the MobileNetV2 architecture with linear bottleneck features and shortcut connections between bottlenecks. The results of this study are:

3.1 Dataset

Total number of datasets for this study is 2,029 images consisting of 1,000 images with masks with details of 968 based on Kaggle data and 32 data from Regional Office VII BKN Palembang employees, then the image dataset without masks amounted to 1,029 images with details 947 data from Kaggle data and 114 data from Office employees. Regional VII BKN Palembang. The results of this dataset can be seen in table 1.

Table 1. Dataset Result Table

Source	Masked Images	Unmasked Images	Total
Kaggle	968	947	1.915
Regional Office VII BKN Palembang employees	32	82	114
Grand Total	1.000	1.029	2.029

3.2 Data Preprocessing

Data preprocessing is an activity to process and analyze data to be converted into a binary class matrix and into augmented data. Figures 4 and 5 show the results of preprocessing image data which is converted into a binary class matrix and into augmented data.

Then the dataset is divided into test data and training data, as many as 1,623 images in the dataset are used as training data and as many as 406 images in the dataset are used as test data.

3.3 Feature Extraction

After the dataset is ready, then proceed with the process of extracting features or feature extraction. The feature extractor used in this process is MobileNetV2. MobileNetV2 is a pre-training model that uses the Convolution Neural Network as the basis for its architecture.

MobileNetV2 is used in this research because of its architecture which is known to be able to overcome the need for over-computing. MobileNetV2 consists of a collection of bottlenecks in which there are 3 layers, namely a pointwise convolution layer equipped with a non-linear ReLU activation function, a depthwise convolution which is also equipped with a ReLU non-linear activation function, and the last one is a pointwise convolution layer but without a ReLU function. non-linear activation.

3.4 Classifications

The proposed CNN architecture using MobilenetV2 can be seen in table 2, where in this architecture we use 18 blocks containing depthwise and pointwise convolution.

Table 2.CNN Architecture

Layer	Output Shape	Param
input_1 (InputLayer)	(None, 224, 224, 3)	0
Conv1_pad (ZeroPadding2D)	(None, 225, 225, 3)	0
Conv1 (Conv2D)	(None, 112, 112, 32)	864
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32)	128
Conv1_relu (ReLU)	(None, 112, 112, 32)	0
expanded_conv_depthwise	(None, 112, 112, 32)	288
expanded_conv_depthwise_BN	(None, 112, 112, 32)	128
expanded_conv_depthwise_relu	(None, 112, 112, 32)	0
expanded_conv_project (Conv2D)	(None, 112, 112, 16)	512
expanded_conv_project_BN	(None, 112, 112, 16)	64
bottleneck_1	(None, 112, 112, 96)	1536
bottleneck_2	(None, 56, 56, 96)	3456
bottleneck_3	(None, 56, 56, 144)	3456
bottleneck_4	(None, 28, 28, 192)	6144
bottleneck_5	(None, 28, 28, 192)	6144
bottleneck_6	(None, 28, 28, 192)	12288
bottleneck_7	(None, 14, 14, 384)	24576
bottleneck_8	(None, 14, 14, 384)	24576
bottleneck_9	(None, 14, 14, 384)	24576
bottleneck_10	(None, 14, 14, 384)	24576

DOI: https://doi.org/10.29207/resti.v6i4.4276

bottleneck_11	(None, 14, 14, 576)	55296
bottleneck_12	(None, 14, 14, 576)	55296
bottleneck_13	(None, 7, 7, 960)	55296
bottleneck_14	(None, 7, 7, 960)	153600
bottleneck_15	(None, 7, 7, 960)	153600
bottleneck_16	(None, 7, 7, 960)	153600
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120
out_relu (ReLU)	(None, 7, 7, 1280)	0
average_pooling2d (AveragePooling)	(None, 1, 1, 1280)	0
Flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 128)	163968
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Based on the test results of the CNN method using a learning rate of 0.001, with an epoch of 50 resulting in accuracy for each optimization used, namely Adam (97.39%), SGD (92.24%), and RMSprop (96.74%).

Figure 9 is the result of the accuracy for each optimization of CNN, it can be seen that the Adam gives a better value than the other two optimizations, while Figure 10 shows the validation loss of the three optimizations, where the valid loss of the three optimizations can be said to be optimal or trained to identify patterns from the dataset, because the more epochs or iterations, the smaller the training loss or when the epochs increase the training error score will decrease.

The x-axis in Figures 9 and 10 is depicted as Epoch or the number of times the model is trained, while the y-axis in Figure 9 shows the magnitude of accuracy and the y-axis in Figure 10 shows the amount of loss.

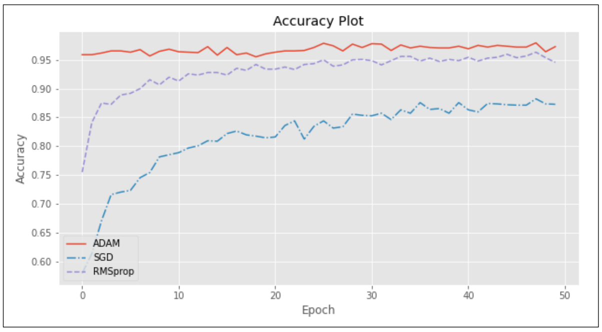

Figure 9. Training Accuracy Plot

Figure 9 shows the process of the accuracy of each model with the optimization of Adam, SGD, and RMSprop in each Epoch. From the picture it can be seen that the movement of Adam's accuracy value has dominated since the first Epoch starting from 95.51% to the last Epoch 97.36% while in RMSprop it started 73.66% to 96.74% and SGD 53.50% to 92.24%. Although the accuracy of RMSprop and SGD has also increased, Adam's performance is better than the other two optimizations.

Figure 10. Training Loss Plot

Figure 10 shows the process of loss results for each model with Adam, SGD, and RMSprop optimizers in each Epoch. From the picture, we can see that the loss value movement of Adam looks quite good since the first Epoch starting from 11.96% to the last Epoch 5.04% while in RMSprop starting from 61.33% to 12.05% and SGD starting from 81.04% to 29.28%. Although the loss value of RMSprop and SGD also decreased, the loss value of Adam was better than the other two optimizers.

3.5 Model Evaluation

The number of objects exposed to light and the distance between the object and the camera will determine the detection results of the model. The amount of data used for testing is 406 images for each model with different optimizations. Based on the test results obtained confusion matrix as shown in Figure 11.

Figure 11 is the result of the confusion matrix, which shows that the mask detection model with Adam's optimizer has succeeded in classifying 193 images of objects that use masks correctly and only has 16 images misclassified. As for the object of people who do not use masks, 190 images were successfully classified correctly, with only 7 images that failed to be classified correctly. Meanwhile, the mask detection model with SGD optimizer succeeded in classifying 190 images of objects that used masks correctly and only experienced a misclassification of 32 images. As for the object of people who do not use masks, 174 images were successfully classified correctly, with only 10 images that failed to be classified correctly. Then the RMSprop optimizer succeeded in classifying 189 object images that used masks correctly and only experienced misclassification of 13 images. As for the object of people who do not use masks, 193 images were successfully classified correctly, with only 11 images that failed to be classified correctly.

Table 2 is a detailed test based on training data, data validation, and data testing with three optimizations from CNN, namely Adam, SGD and RMSprop. From

DOI: https://doi.org/10.29207/resti.v6i4.4276

the table, the average masked and unmasked data was detected at 91.27%.

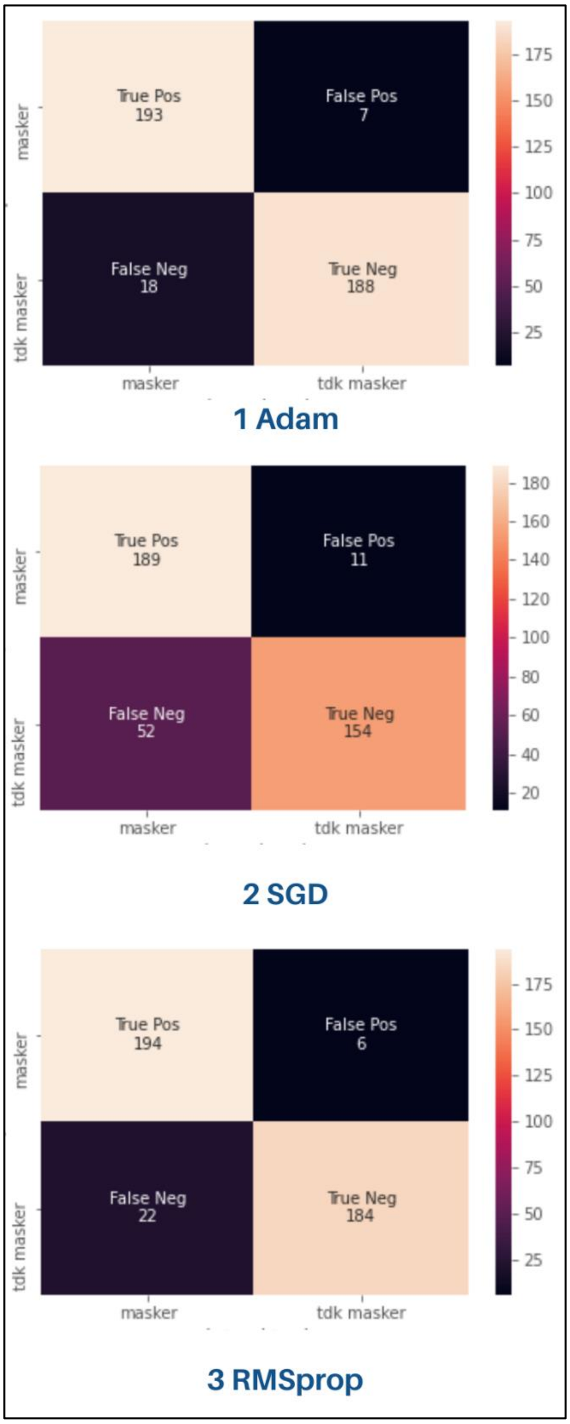

Figure 11. Confusion Matrix

The results of the test evaluation using the proposed model are depicted through a confusion matrix on the recall/sensitivity and precision values for each model with each optimization with details that can be seen in Figure 12 and Table 2.

Table 3.Detection Results Table

Data	Optimizer	Classes	Total Data	Detcted	Undetected
Training	Adam	Masked	696	693	3
		Unmasked	683	651	32
	SGD	Masked	696	679	17
		Unmasked	683	518	165
	RMSprop	Masked	696	679	17
		Unmasked	683	518	165
Validation	Adam	Masked	104	102	2
		Unmasked	140	129	11
	SGD	Masked	104	104	0
		Unmasked	140	114	26
	RMSprop	Masked	104	103	1
		Unmasked	140	122	18
Test	Adam	Masked	200	193	7
		Unmasked	206	188	18
	SGD	Masked	200	189	11
		Unmasked	206	154	52
	RMSprop	Masked	200	194	6
		Unmasked	206	184	22

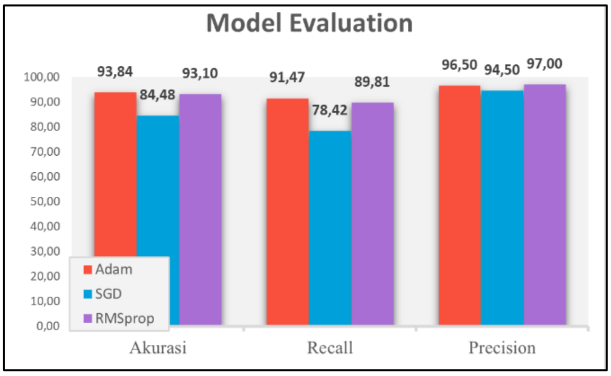

Figure 12. Model Evaluation Plot

Table 4.Model Evaluation Table

Optimizer	Accuracy	Recall	Precision
Adam	93,84%	91,47%	96,50%
SGD	84,48%	78,42%	94,50%
RMSprop	93,10%	89,81%	97,00%

The results of the graph and model evaluation table above show that the detection model can identify whether the object is wearing a mask or not. The model

DOI: https://doi.org/10.29207/resti.v6i4.4276

with Adam optimization seems to get higher accuracy and precision results than the model with SGD or RMSprop optimization. As for recall/sensitivity, RMSprop is able to get the highest value when compared to models using Adam or SGD optimization.

Meanwhile, from Figure 12, it can be seen the performance of the model through the calculation method of precision and recall/sensitivity. From the results of the classification test, it can be seen that the recall/sensitivity value for the Adam optimization model is 91.47% and the precision value is 96.50%. As for the model with SGD optimization, the recall/sensitivity value is 78.42% and the precision value is 94.50%. Then for the model with RMSprop optimization, the recall/sensitivity value is 89.81% and the precision value is 97.00%.

The testing process that has been carried out gives results with an accuracy value of 93.84% for Adam optimization, 84.48% for SGD optimization and 93.10% for RMSprop optimization. Based on the results of the detection of the use of masks using the CNN method, the three optimizations used produce good accuracy values.

These results are obtained based on the Classification Report using the evaluation of the Confusion Matrix as shown in table 5.

Table 5. Analysis of Classification Test Results

Optimizer	Class	Detected	Undetected	Accuracy	Recall	Precision
Adam	Masked	193	7	93,84%	91,47%	96,50%
	Unmasked	188	18			
SGD	Masked	189	11	84,48%	78,42%	94,50%
	Unmasked	154	52			
RMS prop	Masked	194	6	93,10%	89,81%	97,00%
	Unmasked	184	22			

Table 6 provides information on the comparison of the performance we proposed model with related studies that have been carried out.

Table 6.Table of Performance Comparison with Related Research

Author	Dataset	Optimasi	Metodologi	Akurasi
Nyoman P, dkk[4]	Pencarian Internet & Real time	Adam, Gradient Descent	MobileNet V2	90%, 80%
Budiman B, dkk[5]	Kaggle	Tidak disebutkan	MobileNet V2	88.53%
Hairong Wang, dkk[6]	Tidak disebutkan	SGD, Adam RAdam	MobileNet V2	54.14%, 65.4%, 75.23%
Abraham L, dkk[7]	*Real-time dataset* dari kamera *drone*	Adam, RMSprop, SGD, NAdam	MobileNet V2	62.54%, 72.66%, 52.21%, 65.55%
Alsayed A, dkk[8]	Kaggle	SGD, Adam, Adadelta, RMSprop	MobileNet V2	70.60%, 76.64%, 37.08%, 80.49%
Model yang diusulkan	*Kaggle & Real-time dataset dari pegawai Kantor Regional VII BKN Palembang*	*Adam, SGD, RMSprop*	*MobileNetV2*	*93.84%, 84.48%, 93.10%*

Based on table 6, the results of the comparison above show the accuracy value of the model proposed in this study, getting the highest value, namely 93.84% for the Adam optimization model, 84.48% for SGD optimization, and 93.10% for RMSprop optimization. This value is obtained by using the CNN architecture model described in table 2, this architecture which has 18 blocks containing depthwise and pointwise convolution layers is carried out for the three optimizations of Adam, SGD, and RMSprop.

Based on the results of the application of the model that has been carried out and references from several studies, Adam's optimization can provide a higher accuracy value because it can use the best properties of RMSprop optimization to handle sparse gradients on noisy problems [25]. Adam also added the bias-correction feature of RMSprop so that it is proven that this bias-correction feature makes the model results with the Adam optimizer slightly outperform RMSprop [26]. Adam updates the learning rate continuously on each network weight, while in SGD uses a single learning rate for all network weights, because of that Adam is called Adaptive momentum [27].

4. Conclusion

The research model that we have proposed has succeeded in developing a CNN model with the MobilenetV2 architecture using two different dataset sources, namely from Open Access Kaggle and from real time data.

The results given are to see the performance of the Adam, SGD, and RMSprop optimization of the proposed CNN MobilenetV2 model based on a classification report with evaluation of the confusion matrix, namely accuracy, precision and recall.

The results of this study can outperform related research because it provides a higher accuracy value, which is seen from Adam's optimization which is 93.84%. This optimization gives a higher value because of the combination of using the best properties of RMSprop and updating the learning rate on a regular basis which

DOI: https://doi.org/10.29207/resti.v6i4.4276

is different from SGD which is only done once for the entire network weight.

Reference

[1] "WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020." https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020 (accessed Jul. 02, 2021).

[2] Kemenkes, "SE-PENGGUNAAN-MASKER-2020-(2)_1562.pdf." 2020, [Online]. Available: https://covid19.go.id/p/regulasi/surat-edaran-nomor-hk0202i3852020.

[3] world health Organization, "Mask use in the context of COVID-19," *Who*, no. December, pp. 1–10, 2020, [Online]. Available: https://www.who.int/publications/i/item/advice-on-the-use-of-masks-in-the-community-during-home-care-and-in-healthcare-settings-in-the-context-of-the-novel-coronavirus-(2019-ncov)-outbreak.

[4] P. Nyoman and Putu Kusuma Negara, "Deteksi Masker Pencegahan Covid19 Menggunakan Convolutional Neural Network Berbasis Android," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 3, pp. 576–583, 2021, doi: 10.29207/resti.v5i3.3103.

[5] B. Budiman, "Pendeteksian Penggunaan Masker Wajah Dengan Metode Convolutional Neural Network," *J. Ilmu Komput. dan Sist. Inf.*, vol. Vol.9 No.1, 2021.

[6] H. Wang, W. Li, Q. Wan, H. Yan, and R. Xiang, "Research and Application of Facial recognition Algorithm in Audit Investigation," *J. Phys. Conf. Ser.*, vol. 1865, no. 4, p. 42061, 2021, doi: 10.1088/1742-6596/1865/4/042061.

[7] L. Abraham, S. Davy, M. Zawish, R. Mhapsekar, J. A. Finn, and P. Moran, "Preliminary Classification of Selected Farmland Habitats in Ireland Using Deep Neural Networks," *Sensors* , vol. 22, no. 6. 2022, doi: 10.3390/s22062190.

[8] A. Alsayed, A. Alsabei, and M. Arif, "Classification of Apple Tree Leaves Diseases using Deep Learning Methods," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 21, no. 7, p. 324, 2021, [Online]. Available: https://doi.org/10.22937/IJCSNS.2021.21.7.37.

[9] C. Basha, B. N. Pravallika, and E. B. Shankar, "An efficient face mask detector with pytorch and deep learning," *EAI Endorsed Trans. Pervasive Heal. Technol.*, vol. 7, no. 25, p. e4, 2021.

[10] G. S. Jayalakshmi and V. S. Kumar, "Performance analysis of convolutional neural network (CNN) based cancerous skin lesion detection system," in *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, 2019, pp. 1–6.

[11] O. N. Putri, "Implementasi Metode Cnn Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing (Studi Kasus: Gambar Jamur Dengan Genus Agaricus Dan Amanita)," 2020.

[12] "Labeled Mask Dataset (PASCAL_VOC) | Kaggle." https://www.kaggle.com/datasets/techzizou/labeled-mask-dataset-pascal-voc-format (accessed Jun. 15, 2022).

[13] "MobileNet, MobileNetV2, and MobileNetV3." https://keras.io/api/applications/mobilenet/ (accessed Jun. 15, 2022).

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[15] S. Lasniari, J. Jasril, S. Sanjaya, F. Yanto, and M. Affandes, "Pengaruh Hyperparameter Convolutional Neural Network Arsitektur ResNet-50 Pada Klasifikasi Citra Daging Sapi dan Daging Babi," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 3, pp. 474–481, 2022.

[16] A. Putri, B. S. Negara, and S. Sanjaya, "Penerapan Deep Learning Menggunakan VGG-16 untuk Klasifikasi Citra Glioma," *J. Sist. Komput. dan Inform.*, vol. 3, no. 4, pp. 379–383, 2022.

[17] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.

[18] M. F. Naufal and S. F. Kusuma, "Pendeteksi Citra Masker Wajah Menggunakan CNN dan Transfer Learning," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 6, pp. 1293–1300, 2021.

[19] "CS231n Convolutional Neural Networks for Visual Recognition." https://cs231n.github.io/convolutional-networks/#overview (accessed Jun. 11, 2022).

[20] T. Ahmed, P. Das, M. Ali, and M. Mahmud, *A Comparative Study on Convolutional Neural Network Based Face Recognition*. 2020.

[21] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for image classification," *Int. J. Adv. Trends Comput. Sci. Eng*, vol. 9, no. 10, 2020.

[22] A. Wikarta, A. S. Pramono, and J. B. Ariatedja, "Analisa Bermacam Optimizer Pada Convolutional Neural Network Untuk Deteksi Pemakaian Masker," *Semin. Nas. Inform. 2020 (SEMNASIF 2020)*, vol. 2020, no. Semnasif, pp. 69–72, 2020.

[23] G. Hinton, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," vol. 15, pp. 1929–1958, 2014.

[24] T. Shafira, "Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras." Universitas Islam Indonesia, 2018.

[25] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[26] S. Ruder, "An overview of gradient descent optimization," pp. 1–14, 2016.

[27] B. While, N. Stochastic, and F. Imagenet, "W a b sgd," pp. 1–19, 2020.

DOI: https://doi.org/10.29207/resti.v6i4.4276