International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 12– Issue 02, June 2023

Identification of Medical Mask Use by Applying the Convolutional Neural Network Algorithm and the Gabor Filter with Multiclass Classification

Muh Arifandi
University of Technology Yogyakarta: Master of Information Technology
Yogyakarta, Indonesia
Email: muh.arifandi [AT] student.uty.ac.id

Erik Iman Heri Ujianto
University of Technology Yogyakarta: Master of Information Technology
Yogyakarta, Indonesia
Email: erik.iman [AT] uty.ac.id

Abstract— **Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV-2) causes global pandemics and makes countries around the world lock down for tourists. This action is required to prevent the spread of viruses that take 14 days to disappear. SARS-COV-2 can easily infect individuals through a droplet. Thus, the governments of every country worldwide recommend wearing medical masks to prevent the spread of viruses, as well as maintaining distance during activities with others and washing hands frequently. Medical masks become efficient if their application is precise, owing to a lack of knowledge and self-awareness to preserve their distance and wash their hands. This paper proposes a Convolutional Neural Network (CNN) with Gabor filter implementation. The simulation uses a mask on a dataset with over 70,000 individual photos. The results demonstrated that the proposed CNN-Gabor model in this work could effectively classify the position of the mask when compared to the CNN model without the Gabor filter.**

Keywords-convolutional neural network, gabor filter, medical mask classification, multiclass classification.

I. INTRODUCTION

The world is experiencing a pandemic caused by Severe Acute Respiratory Syndrome Coronavirus 2. (SARS- COV-2). The existence of this pandemic prompts several countries to undertake lockdown measures in order to prevent the virus spread and the strain on health workers [1]. Patients infected with Covid-19 require around 14 days to heal orcomplete the isolation period before it is definitively confirmed that they no longer transmit the virus.

Based on epidemiological studies and virology proves that Covid-19 is mainly transmitted from people who are symptomatic to others who are in close range through droplet [2]. Transmission of droplets occurs when someone is at a close distance that is about 1 meter from someone who has breathing symptoms such as coughing or sneezing, so the droplet is at risk of sticking to other people.

Based on these conditions, COVID-19 prevention includes wearing medical masks, keeping a distance in activities, routine hand washing, and constantly applying health protocols.

Machine Learning (ML) is a branch of artificial intelligence in which a machine uses intelligent software to solve problems [3]. Tensorflow is one of many libraries that are collections of functions that can be utilized to develop ML algorithms [4]. ML can carry out specific tasks by studying data and statistical models used during training. In general, ML can be divided into three types: (1) supervised learning, (2) unsupervised learning, and (3) deep learning (DL).

DL is a subset of Artificial Intelligence (AI), including a hidden layer that processes RAW datasets [5]. DL employs a simple representation yet can build complex concepts based on the dataset used during the training process. Several algorithms can be used to train models, such as mobile networks, VGGnet, and so forth. Following that, object detection can use algorithm types such as (1) Yolo, (2) SSD Resnet, and (3) MTCNN.

II. RELATED WORK

Several studies have been done to limit the spread of the Covid-19 virus, including research into the use of artificial intelligence in the pandemic era, such as mask detection. Ejaz conducted one of the studies related to introducing the face of a mask to classify the use of masks in the scope of society [6]. The detection value based on accuracy in the study ranged from 98% to 99%. Venkateswarlu conducted similar research in which the authors employed the mobilenet model.

This pre-trained model uses specific datasets with several global max pooling to detect facial masks [7]. The global pooling block used inthe proposed model converts the input of the image into a vector with a total of 64 features. In the last layer, the author employs the SoftMax Activation Function to conduct binary clarification based on the previously processed 64 types of features. Furthermore, Global Max Pooling is also used in the proposed model to prevent overfitting. In research, evaluation is carried out utilizing two types of classes, also known as binary class classification. The dataset contains photos of individuals with and without masks. The total data used in the research was 11,792, with an accuracy rate of 99%.

XU reported that the results of his research reached an accuracy of 90.2% [8]. The proposed method in this research uses the SSD Mask algorithm to solve the detection problem on an object with a small size. Other research focused on face detection in real-time that can activate the alarm when the detection results show that a person is not using a mask [9]. The proposed system is designed and exercised using a dataset of 35,806 images. The test results showed that the accuracy reached up to 90.1%. Loey also conducted similar research with the number of Dataset 1415, which is also to solve the problem of binary classification [10]. Research conducted by Loey only focused on medical masks. The research results achieved a fairly decent accuracy of 81%. In the same year, similar research was conducted but with adifferent method [11]. The proposed method consists of two components. The first component is intended for feature extraction using Resnet50 from Pretrained Network. The second component is intended for face mask classificationusing the Decision Tree, Support Vector Machine (SVM) technique, and Ensemble Learning. The results of the research can attain an accuracy of 99.64%.

In 2021, Srinivasan's similar research was conducted using MobileNetv2 architecture as a classification technique and Yolo as an object detection [12]. The proposed model uses 11,792 dataset images and is implemented in real-time video. Based on the resulting performance, the proposed model has an accuracy of 90.2%. In the same year, Negi proposed the DL model based on CNN and VGG16 to increase data accuracy to detect the use of face masks on the Simulated Masked Face Dataset (SMFD) (Negi et al., 2021).

The following research was conducted by Sethi [13]. The method proposed in this study uses resnet50 and the conceptof transfer learning. The proposed technique is one-stage and two-stage ensemble detector to reach low inference time and high accuracy. With the transfer learning method on Resnet50 to combine feature information, high accuracy of 98.2% is produced when implemented with Resnet50.

There are 45,000 photos in the research dataset. According to previous research, the accuracy value in the model for mask detection tasks has always remained the same by using feature extraction methods. In this research, we propose using Gabor filters for advanced feature extraction, which was then forwarded to the CNN model as a detector.

III. PROPOSED MODEL

A. *Model Proposed Convolutional Neural Network*

CNN is one of the various types of nerve tissue in DL techniques that can be used for image detection and classification. CNN is formed with a convolutional Layer with a certain weight, bias, and activation function. The main difference between CNN and other nerve tissue is found in the model structure. CNN has a layer to do a feature extraction followed by a detector. Feature Extraction focuses on gathering necessary information from the image dataset. The information is then forwarded to the classification section, which will determine the class type based on the probability value. CNN was chosen to be one of the main options for the classification of images because of its proven ability. In this paper, we propose a CNN-based model with modifications to the feature extraction section to carry out the Multiclass Classification task. *Figure* 1 depicts the architecture of the proposed model. ConvBlock with a given number of neuronsis used in the proposed model. ConvBlock128, for example, comprises 3x1 convolutional2D neurons with 128 neurons. The Rectified Linear Unit (Relu) is used in the activation function, followed by a layer of batch normalization and the max pooling layer with a 2x2 dimension to do a better feature extraction.

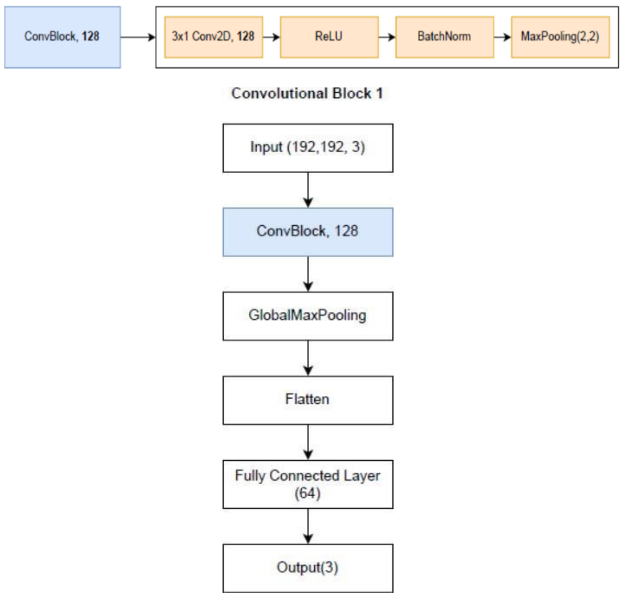

Figure 1: CNN -based architecture proposed for the classification of Multi Class Mask

In addition, the Global Max Pooling layer is also implemented to execute feature extraction sullenly and guarantee the model's generalization. The flattened layer is then added before transmitting data to the classification section to extract and form data in a one-dimensional format used for classification. Consequently, a fully connected layer with 64 neurons was added to the proposed model. A softmax activation function was designed to provide several probabilities based on the number of classes available on the dataset. The highest probability given by the SoftMaxActivation Function will be utilized as a reference to the model's prediction outcomes in a particular image.

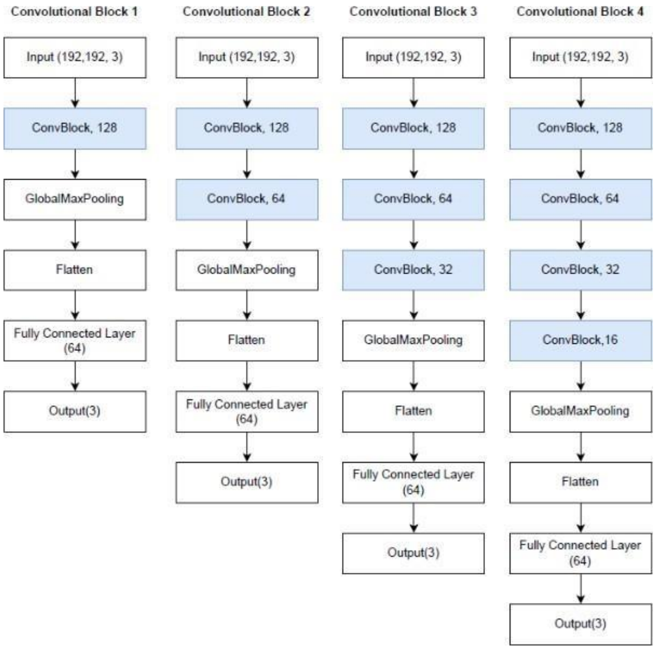

Figure 2: Proposed variations in CNN -based models with several ConvBlock configurations.

In the future, we will vary the number of ConvBlocks configured in the proposed CNN model to achieve the best performance model with the highest efficiency value. In this research, four different block designs were observed. Figure 2 illustrates all of the configurations that were employed. The main differences between each configuration are the total quantity of the ConvBlock and the neuron size provided in each block. As can be observed, we minimize the number ofneurons by adding ConvBlock to ensure that the complexity of computing does not increase significantly.

B. Gabor Filter Implementation

Gabor's filter is one of the texture descriptors that Gabor introduced in 1946 [14]. Gabor filters are implemented to extract image features by analyzing frequency domains [15]. Gabor's filter, in general, is a modified gaussian function that meets the complexity of the sinusoidal from frequency and orientation. The Gabor filter is also the primary choice to be implemented because of its ability to distinguish between two textures and more reliable results [16]. This paper integrates Gabor filters with CNN-based models to carry out features and increase multiple classification accuracy. Figure 3 illustrates the use of Gabor filters in the proposed model.

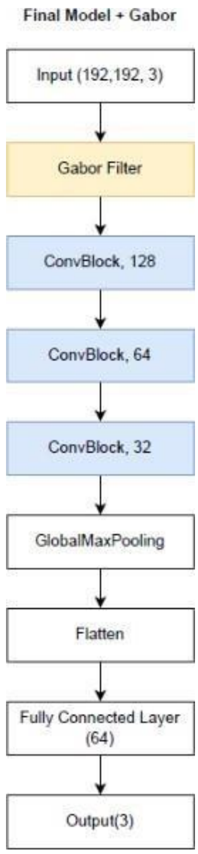

Figure 3: Implementation of Gabor filter on proposed CNN-based model with three ConvBlocks.

Gabor filter is used after the input portion of the proposed CNN model. This choice is due to the Gabor filter having a better performance for feature extraction, so that it will increase the extraction process on the proposed CNN model. As can be observed, the output of the Gabor filter is passed to several ConvBlocks for further feature extraction. Because the results provided by the Gabor filter are more detailed than feature extraction on CNN, the performance values obtained are more accurate.

C. Simulation Configuration

To test the simulation method, Jupyter Notebook was used in the Anaconda virtual environment using Tensorflow and Keras as the main libraries to develop the proposed CNN model. MaskedFace-Net, a dataset with over 70,000 image samples, was used for this research.

The dataset is divided into three parts: (1) 70% for training, (2) 20% for validation, and (3) 10% for testing. The number of epochs used varied from 50, 75, and 100. In addition, libraries in the Python programming language were also used in this research, such as CV2, NumPy, Pandas, Scikit-Learn, and Matplotlib, to implement the Gabor filter, dataset preprocessing, data analysis, and result plotting. The library is installed on the Asus ROG Zeypurs notebook with an Intel i9 processor, 24GB RAM, and an M2 SSD with a size of 512GB. The complete configuration of the simulation parameters used in this research is shown in more detail in Table 1.

Table 1: The proposed CNN-based multiclass classification simulation configuration uses Gabor filters

Software	
Environment	Anaconda with Jupyter Notebook
Deep Learning Library	TensorFlow and Keras
Additional Library	CV2, NumPy, Pandas, Scikit-Learn, Matplotlib

D. Test Matrix

This research employs several metrics to evaluate the proposed CNN-based model with the Gabor filter implementation. First, the accuracy and loss values were obtained from the validation data. The study then investigates precision, recall, and F1-score to provide a complete performance comparison. Furthermore, we consider the complexity of the proposed model; hence a metric based on training time is used to demonstrate the algorithm's efficiency. Lastly, the confusion matrix is used to demonstrate the overall performance of the proposed CNN-based model for multiclass classification with and without Gabor filter implementation. The following formula can be used to calculate the metrics mentioned above:

$$Accuracy\ (\%) = \frac{Tp\ \$\ Tn}{Tp\ \$\ Tn\ \$\ Fn\ \$\ Fp}\ x\ 100\ , \quad (1)$$

$$Precision\ (\%) = \frac{Tp}{Tp\ \$\ Fp}\ x\ 100\ , \quad (2)$$

$$Recall\ (\%) = \frac{Tp}{Tp\ \$\ Fn}\ x\ 100\ , \quad (3)$$

$$F1 - Score\ (\%) = \ 2\ x\ < \frac{Precision\ x\ Recall}{Precision\ \$\ Recall} = , \quad (4)$$

Tp represents a *true positive*, followed by *Tn*, a *true negative*, and *Fp* and *Fn*, a *false positive* and a *false negative*, respectively

IV. Result and Discussion

A. Evaluation of Total Iterations

We simulated the first test by changing the number of iterations used to train the proposed CNN model. This evaluation investigated the optimal trade-off between the number of selected iterations and overall performance based on loss and accuracy. The test results of the various iteration values are shown in Figure 4 for the loss with the sparse categorical cross-entropy method, followed by Figure 5 for the training and validation data accuracy.

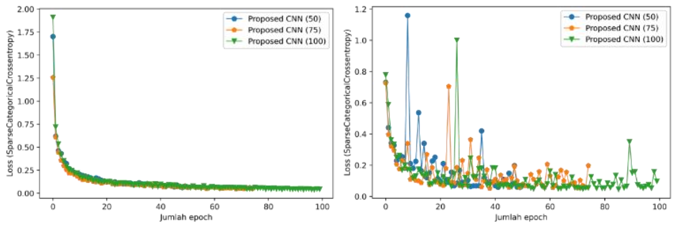

Figure 4: Training loss (left) and validation (right) of the epoch size variation of the proposed CNN model

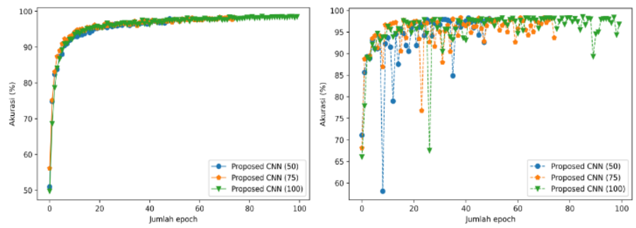

Figure 5: Training accuracy (left) and validation (right) of the variation of the epoch sizes of the proposed CNN model

Based on the three types of variations tested, we were starting from 50, 75 and 100 iterations. We observe that testing with 75 training iterations yields the most significant increase in performance when using a single ConvBlock. Using an image size of 192 x 192 x 3 as input, 32 batch sizes and an Adam optimizer, 75 training epochs can provide an accuracy of 98.31% and a loss of 0.048. This performance is the best trade-off compared to the other epoch variations, 50 and 100. Nevertheless, while a higher number of iterations say 100, gives higher performance, the system's training duration increases dramatically as the numberof iterations increases. Based on the epoch investigation findings, we used 75 epochs as the optimal number of iterations to train the proposed CNN-based model for multiclass classification in this research's use case of mask detection.

B. CNN Optimizer Evaluation

After determining the best iteration size for the training process, we simulate to improve the performance of the proposed CNN-based model by modifying the optimizer type utilized in model training. In this scenario, we employ three types of optimizers that are popularly used for image classification problems on CNN:

(1) Adam, (2) Stochastic Gradient Descent (SGD), and (3) Root Mean Squared Propagation (RMSProp). In this scenario, the test is conducted for 75 epochs with one ConvBlock, a batch size of 32, and a learning rate of 0.001 for all optimizer types.

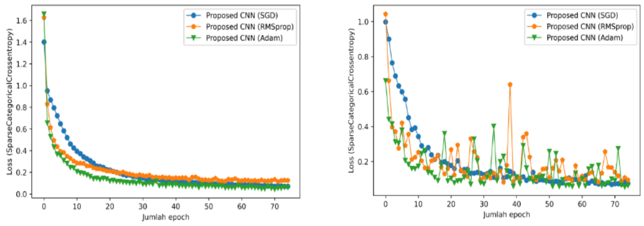

Figure 5: Training loss (left) and validation (right) of the optimizer's variation in the proposed CNN-based model

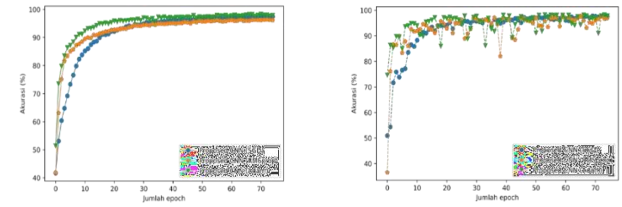

Figure 6: Training accuracy (left) and validation (right) of the optimizer's variations on the CNN-based model

According to the evaluation results in Figure 6, Adam Optimizer outperforms SGD and RMSPROP regarding loss for training and validation data. Furthermore, Adam Optimizer provided consistent performance with a lower loss value than the initial training iteration. Similar findings are seen based on loss from dataset validation, where Adam Optimizer constantly outperforms other optimizers (SGD and RMSProp).

The accuracy value of the model was also investigated utilizing three different optimizers. *Figure* 7 shows that Adam Optimizer outperforms other optimizers, such as SGD and RMSProp, in terms of the performance of the proposed CNN model. Adam Optimizer has a validation accuracy of roughly 97.84%, followed by SGD and RMSPOP with 97.56% and 96.93%, respectively. According to these results, we employ Adam as the primary optimizer in theproposed CNN model because it provides the best consistent performance.

C. The size of the convolutional block Evaluation

As described in *Figure* 2, four types of variations of CNN-based models are proposed for this work. These variations are based on the amount of ConvBlock added to the feature extraction process. In this section, the investigation of the variation of the model is discussed. The optimal configuration for EPOCH and Optimizer is obtained from the results of previous tests, then the amount learning rate is set by 0.001, and the batch size is set at 32.

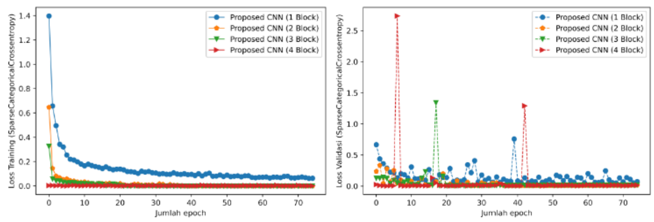

Figure 7: Training loss (left) and validation (right) of the variations of the conversion block in the CNN-based proposal model

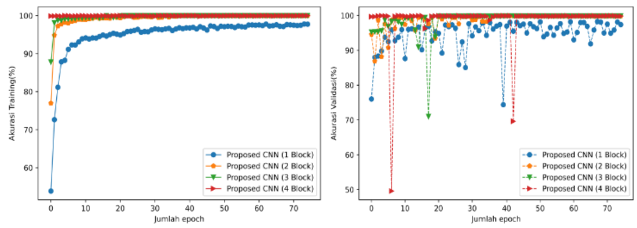

Figure 8: *Training accuracy (left) and validation (right) of the variations of the conversion block in the CNN-based proposal model*

First, we investigate all losses in the training and loss process in the validation process with various variations of ConvBlock. *Figure* 8 illustrates the loss value with 75 EPOCHs for training. The larger size of the ConvBlock is expected to provide better performance. The findings generally show that the highest performance is obtained using four layers of ConvBlock. ConvBlock 2-level gives a loss of 0.063, followed by ConvBlock 3 with a loss of 0.0003 and ConvBlock 4 reaching a loss of 0.001. ConvBlock 4 performance is worse than ConvBlock 3 in terms of loss from the dataset for training and validation. We argue that the configuration of models with ConvBlock 4 experiences overfitting and causes the performance value produced by the model to decline.

Then, as shown in *Figure* 9, we investigate the accuracy of the dataset used for training and validation. The findings show that Convblock 3 can achieve better than other configurations. The accuracy produced by the proposal model employing a single ConvBlock is 97.74%, while ConvBlock 2 and ConvBlock 3 achieved 100% accuracy. Nevertheless, ConvBlock 4 can only provide 99.98% accuracy due to an overfitting model.

Table 2: The performance results of the proposed CNN-based model are based on the ConvBlock variation

Convolutional block	***Precision (%)***	***Recall (%)***	***F1-Score (%)***	***Training time(s)***
1	97	0.97	0.97	947.92
2	99	0.99	0.99	1164.56
3	**100**	**1.0**	**1.0**	**1210.61**
4	100	1.0	1.0	1235.59

In order to further compare the various block configurations observed in this study, we use precision, recall, and f1-score metrics to understand the overall model performance. Table 2 shows that only ConvBlock 3 and 4 variations can achieve 100% precision and recall compared to other model configurations. Referring to the loss, accuracy, and training time, it can be observed that the optimal configuration for the CNN-based model proposed in this research uses three layers of ConvBlock. Based on the results obtained, a three-layer ConvBlock further improves the model performance in the following evaluation.

D. Learning Rate Evaluation

We adjust the parameters utilized in training in order to enhance the performance of the proposed CNN model. Based on previous results, we found that the Adam optimizer worked and produced the best performance with the proposed CNN-based model for the mask classification problem. In the previous evaluation, the learning rate was set at a constant value of 0.001. In this section, we provide four types of configurations of the learning rate in this section, namely 0.01,0.001, 0.005, and 0.0001.

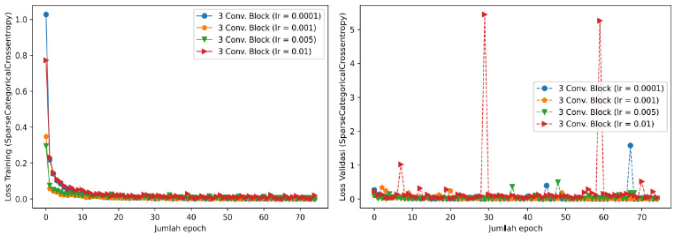

Figure 9: Training loss (left) and validation (right) of various learning rates using three ConvBlocks

First, we investigated the resulting loss of the dataset for training and validation. *Figure* 10 shows the variation of loss based on the different learning rates. We have found that a learning rate of 0.0001 can give a loss of 0.0013. Similar results are obtained using a learning rate of 0.001, with a loss of 0.0033. Then, learning rates of 0.005 and 0.01 produce losses of 0.0001 and 0.02, respectively. Based on these results, the learning rate of 0.005 is superior to other variations in the loss metric for training and validation.

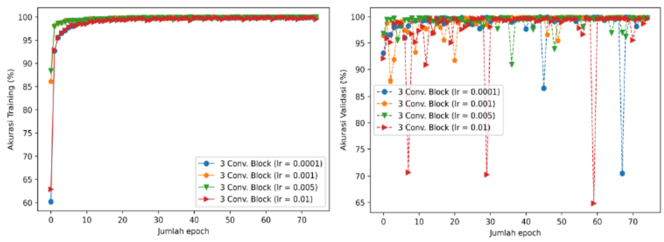

Figure 10: Training accuracy (left) and validation (right) of various learning rates using three ConvBlocks

We also investigate the impact of learning rate on accuracy using training and validation datasets to provide a comprehensive evaluation. Four different outcomes from varied learning rate configurations are shown in *Figure* 11. The outcomes demonstrated that the 0.005 learning rate performance outperformed other configurations. It should be noticed that learning rates of 0.001, 0.005, and 0.0001 result in more steady performance with minimum fluctuation values than learning rates of 0.01.

Table 3: Performance results of the proposed CNN-based model basedon learning rate variations on Adam optimizer

Learning rate (lr)	***Precision (%)***	***Recall (%)***	***F1-Score (%)***	***Training time(s)***
lr = 0.0001	100	100	100	747.11
lr = 0.001	100	100	100	744.45
lr = 0.005	**100**	**100**	**100**	**744.37**
lr = 0.01	99	99	99	740.07

Moreover, the precision, recall, f1-score, and training duration values for each learning rate setting are noted. Table 3 shows all of the test's detailed results. Similar to the initial findings on the accuracy, learning rate values of 0.001, 0.005, and 0.0001 achieved the best results with 100% f1-scores, precision, and recall. These results indicate that the three learning rate variations can support the proposed model to achieve the value with the highest performance. In thisresearch, we also considered training time so that the learning rate that required the lowest training time, which is a learning rate of 0.005 with a total training time of 744.37 s, was selectedand used as the learning rate for the proposed model.

E. Gabor Filter Implementation Analysis

Eventually, we investigated the performance of the proposed CNN-based model with and without the Gabor filter. Evaluation is based on the latest parameter configurations from the previous scenario. First, the number of training iterations is set to 75, and the number of ConvBlocks is set to three layers, followed by the Adam optimizer with a learning rate of 0.0005.

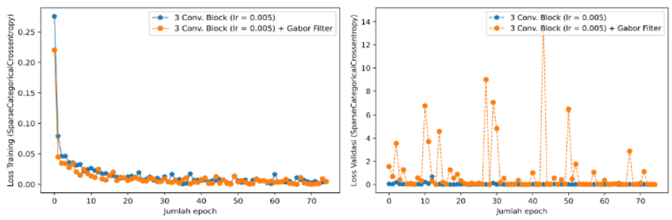

Figure 11: Training loss (left) and validation (right) of the proposed CNN model with and without the Gabor Filter

Figure 12 shows the loss from the proposed model's training and validation using the Gabor filter. According to the simulation results, the model with the Gabor filter can provide higher performance outcomes than the model without the filter. For example, in the first ten training iterations, the proposed CNN Gabor filter produces a loss of 0.22, while CNN without the Gabor filter results in a loss of 0.27, which is 0.05 greater.

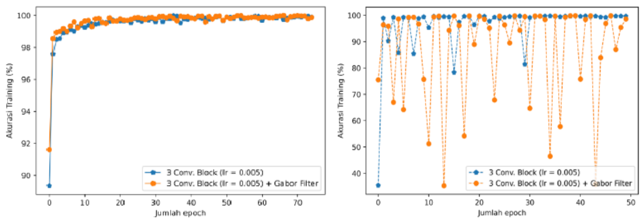

Figure 12: Training accuracy (left) and validation (right) of the proposed CNN model with and without Gabor Filter

Furthermore, regarding accuracy results for the training and validation datasets, the proposed CNN with Gabor filter can perform better at lower epoch values. The learning process of the proposed CNN-based model can be improved using the Gabor filter to give better results in a shorter training period. In the last round of the epoch, CNN Gabor obtained an accuracy of 99.89%, higher than CNNwithout a Gabor filter, with an accuracy of 99.71%.

Table 4: The results of the proposed CNN-based model performance are based on the use of the Gabor filter

Model Configurations	***Precision (%)***	***Recall (%)***	***F1-Score (%)***	***Training time (s)***
Convolutional block 3 Learning rate = 0.005	100	100	100	842.37
Convolutional block 3 Learning rate = 0.005 + ***Gabor Filter***	**100**	**100**	**100**	**406.35**

We also investigate precision, recall, f1-score, and training time to demonstrate the quality of the proposed model. As shown in Table 4, the proposed CNN-based model is very high, with precision, recall, and f1-score reaching 100%. Training time has increased significantly. The proposed CNN-based model can provide a much more efficient classification training period by employing the Gabor filter. The results show that Gabor CNN requires just 406.35 s to training time, whereas CNN without the Gabor filter requires 842.37 s. An efficiency value of 50% is obtained by implementing the Gabor filter with the proposed CNN-based model. In addition, we also investigate the confusion matrix of the proposed CNN-based model with and without Gabor filter, where the results are depicted in *Figure* 14. It can be concluded that the performance of the CNN model with Gabor filter is superior to CNN without adopting the Gabor filter in terms of multiclass classification.

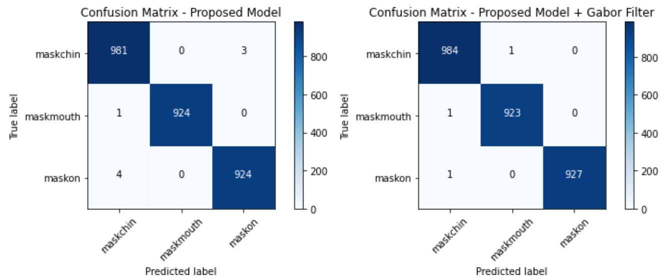

Figure 13: Confusion matrix of proposed CNN-based model without (left) and with Gabor filter (right)

CONCLUSSION

Based on the results of tests using five types of scenarios to assess the performance produced by the proposed CNNmodel to resolve the mask calcification problem under multiclass conditions, it was discovered that combining the CNN model with the Gabor filter can increase performance in terms of accuracy and lowering training time. First, testing is carried out to find optimal values for several hyperparameters, such as (1) epoch, (2) optimizer, (3) number of ConvBlocks, and (4) learning rate that can be configured in the proposed model. The resulting optimal hyperparameter values are then used to test the proposed model without additions and with the addition of the Gabor filter. Using the Gabor filter, tests show that better accuracy, precision, recall, f1-score, training time, and confusion matrix values are produced. In addition, the Gabor filter can significantly reduce training time and maintain optimal accuracy values. Furthermore, the results of testing with the confusion matrix shown in more detail that the CNN model with the Gabor filter can perform better mask classification than the CNN model without the Gabor filter. So, the Gabor filter has a significant influence and can be implemented on the CNN model to improve model performance and significantly reduce training time.

REFERENCES

[1] Subhamastan Rao, T., Anjali Devi, S., Dileep, P., & Sitha Ram, M.(2020). A Novel Approach to Detect Face Mask to Control Covid Using Deep Learning. European Journal of Molecular and Clinical Medicine, 7(6), 658–668. https://doi.org/10.1109/ICISS49785.2020.9315927

[2] Isbaniah, F. (2020). Pedoman Pencegahan dan Pengendalian Coronavirus Disease (COVID-19). In Germas. https://infeksiemerging.kemkes.go.id/download/REV-04_Pedoman_P2_COVID-19_27_Maret2020_TTD1.pdf

[3] Mohammed, M. (2016). Machine learning: Algorithms and Applications. https://doi.org/10.1201/9781315371658

[4] Wiranda, N., Purba, H. S., & Sukmawati, R. A. (2020). Survei Penggunaan Tensorflow pada Machine Learning untuk Identifikasi Ikan Kawasan Lahan Basah. IJEIS (Indonesian Journal of Electronics and Instrumentation Systems), 10(2), 179. https://doi.org/10.22146/ijeis.58315

[5] Evolve, M. L. (2019). Mengenal Machine Learning. Medium.Com. Fukushima, K., & Miyake, S. (1982). Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition (pp. 267–285). https://doi.org/10.1007/978-3-642-46466-9_18

[6] Ejaz, Md. S., & Islam, Md. R. (2019). Masked Face Recognition Using Convolutional Neural Network. 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 1–6. https://doi.org/10.1109/STI47673.2019.9068044

[7] Venkateswarlu, I. B., Kakarla, J., & Prakash, S. (2020). Face mask detection using MobileNet and Global Pooling Block. 2020 IEEE 4th Conference on Information & Communication Technology (CICT), 1–5. https://doi.org/10.1109/CICT51604.2020.9312083

[8] Xu, M., Wang, H., Yang, S., & Li, R. (2020). Mask wearing detection method based on SSD-Mask algorithm. 2020 International Conference on Computer Science and Management Technology (ICCSMT), 138–143. https://doi.org/10.1109/ICCSMT51754.2020.00034

[9] Vinh, T. Q., & Anh, N. T. N. (2020). Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier. 2020 International Conference on Advanced Computing and Applications (ACOMP), 146–149. https://doi.org/10.1109/ACOMP50827.2020.00029

[10] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021a). A hybrid deep transfer learning model with machine learning methods forface mask detection in the era of the COVID-19 pandemic. Measurement, 167, 108288. https://doi.org/10.1016/j.measurement.2020.108288

[11] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021b). Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustainable Cities and Society, 65, 102600. https://doi.org/10.1016/j.scs.2020.102600

[12] Srinivasan, S., Rujula Singh, R., Biradar, R. R., & Revathi, S. (2021). COVID-19 Monitoring System using Social Distancing and Face Mask Detection on Surveillance video datasets. 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 449–455. https://doi.org/10.1109/ESCI50559.2021.9396783

[13] Sethi, S., Kathuria, M., Kaushik T. (2021). Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread, Journalof Biomedical Informatics, Volume 120, 2021. https://doi.org/10.1016/j.jbi.2021.103848

[14] Gabor. (1946). Theory of Communication. Institution of Electrical Engineering, 93(3), 429–457. https://doi.org/10.1049/sqj.1970.0021

[15] Jing Yi, Yong, & Phooi. (2007). Gabor Filters and Grey-level Cooccurrence Matrices in Texture Classification. MMU International Symp. on Information and Communications Technologies.

[16] Hammoud. (2000). Texture segmentation using Gabor filters. KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516), 109–112.https://doi.org/10.1109/KES.2000.885770