Science in Information Technology Letters
Vol. 4., No. 1, May 2023, pp. 54-64
ISSN 2722-4139
http://pubs2.ascee.org/index.php/sitech

Optimizing CNN hyperparameters with genetic algorithms for face mask usage classification

Awang Hendrianto Pratomo [a,1,*], Nur Heri Cahyana [a,2], Septi Nur Indrawati [a,3]

[a] Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia
[1] awang@upnyk.ac.id; [2] nur.hericahyn@upnyk.ac.id; [3]123160070@student.upnyk.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received April 13, 2023
Revised April 26, 2023
Accepted May 10, 2023

Keywords
Face mask usage
CNN
Genetic algorithms
Hyperparameters selection

ABSTRACT

Convolutional Neural Networks (CNNs) have gained significant traction in the field of image categorization, particularly in the domains of health and safety. This study aims to categorize the utilization of face masks, which is a vital determinant of respiratory health. Convolutional neural networks (CNNs) possess a high level of complexity, making it crucial to execute hyperparameter adjustment in order to optimize the performance of the model. The conventional approach of trial-and-error hyperparameter configuration often yields suboptimal outcomes and is time-consuming. Genetic Algorithms (GA), an optimization technique grounded in the principles of natural selection, were employed to identify the optimal hyperparameters for Convolutional Neural Networks (CNNs). The objective was to enhance the performance of the model, namely in the classification of photographs into two categories: those with face masks and those without face masks. The convolutional neural network (CNN) model, which was enhanced by the utilization of hyperparameters adjusted by a genetic algorithm (GA), demonstrated a commendable accuracy rate of 94.82% following rigorous testing and validation procedures. The observed outcome exhibited a 2.04% improvement compared to models that employed a trial and error approach for hyperparameter tuning. Our research exhibits exceptional quality in the domain of investigations utilizing Convolutional Neural Networks (CNNs). Our research integrates the resilience of Genetic Algorithms (GA), in contrast to previous studies that employed Convolutional Neural Networks (CNN) or conventional machine learning models without adjusting hyperparameters. This unique approach enhances the accuracy and methodology of hyperparameter tuning in Convolutional Neural Networks (CNNs).

1. Introduction

Masks are mandatory for the public to wear when carrying out activities in public spaces during the Covid-19 pandemic [1]. The use of masks can help inhibit the spread of Covid-19 because it can be used as a barrier for droplets that come out of the mouth [2]. Therefore, the correct use of masks that cover the nose, mouth and chin needs to be applied so that the mask functions optimally [3]. Even so, based on facts in the field, there are still often people who do not use masks or use masks incorrectly [4].

Authorities have attempted to monitor and regulate the use of masks, but due to the large monitoring area and the large number of people to monitor, these efforts have been ineffective.

Artificial intelligence can be used to classify the use of masks automatically due to differences in distinctive characteristics between the faces of mask users and not [5], [6]. Research related to the classification of mask use has been carried out using three types of machine learning algorithms, namely Decision Tree, Support Vector Machine (SVM), and Ensemble [7]. From this study, the SVM algorithm showed the highest accuracy result, which was 99%. To obtain this accuracy, the researchers concluded that image classification using machine learning algorithms requires additional processes to extract features and requires a long training time. Another study was conducted by designing and using the SRCNet deep learning algorithm [3]. SRCNet is a combination of CNN which is known to have a high ability to classify two-dimensional data and Super Resolution networks so that it is able to classify low-resolution images well. The results obtained from the study amounted to 98.70%.

CNN itself performs well in classification due to its flexible architecture [8], [9]. The ability to extract features automatically and the number of layers and neurons that can be adjusted according to needs are the advantages of CNN. But because of this flexibility, finding the best combination of hyperparameters to produce the optimal model was a challenge at CNN [10]. Manual adjustment of hyperparameters is a difficult and time-consuming task due to the large number of possible combinations sought [11]. Therefore, some studies utilize search and optimization algorithms to automatically search for optimal hyperparameter combinations.

Some studies that have been conducted to look for combinations of CNN hyperparameters include using the Particle Swarm Optimization (PSO) method [12], Bayesian Optimization (BO) [13], and Parameter-Setting-Free Harmony Search (PSF-HS) [14]. Although they work differently, the results obtained from each study show that automatic hyperparameter tuning can improve the accuracy of the model created. In addition to these methods, there is also a Genetic Algorithm (GA) which has been widely applied to optimize other classification algorithms such as LSTM [15] as well as SVM [16], RF [17], Adaboost [18], and KNN [11]. Compared to other optimization algorithms, GA has a more reliable ability to optimize complex search problems and has a large search space because it works by forming a population [11].

Enhanced technical solutions are needed to facilitate efficient and real-time monitoring of mask usage in public settings, given the global health landscape and the significance of this practice. However, the in the previous study, Observational data highlighted issues of non-compliance and improper mask usage among the public, and determining the optimal combination of Convolutional Neural Network (CNN) hyperparameters presents challenges due to its vast possibilities, prompting research into automation solutions. In order to fulfill this requirement, our study integrates two sophisticated algorithms, namely CNN and Genetic Algorithm (GA), to develop an innovative categorization model.

CNNs have demonstrated significant advancements in the field of picture categorization, particularly in the domain of facial recognition. In this experiment, CNN has been meticulously designed to identify masks in facial picture data. CNNs possess significant computational capabilities; yet, optimizing their hyperparameters has been shown to enhance their overall performance. Genetic algorithms (GA), renowned for its utilization of natural evolution as a source of inspiration, are employed to optimize the hyperparameters of Convolutional Neural Networks (CNN). The integration of these factors guarantees an enhanced level of sensitivity and specificity in the model's classification of mask usage. This distinctive combination serves two objectives. The proposed methodology has the potential to significantly decrease both false positives and false negatives by improving accuracy in precisely measuring mask compliance. Furthermore, the utilization of optimization techniques enhances the accuracy and efficiency of categorization, rendering it well-suited for real-time applications. The present study has promise for enhancing the monitoring of mask wear and promoting public health safety.

2. Method

There are several main stages carried out in research, namely data collection, data preprocessing, model making, testing, and system creation. Fig. 1 illustrates the process of developing the proposed model. The initial stage of the process involves the collection of data, which encompasses the capture and tagging of datasets. In the process of data preprocessing, the data undergoes three specific operations: grayscaling, resizing, and normalization. Following the preprocessing stage, the process of Model Development commences by implementing data augmentation techniques and subsequently dividing the dataset into separate training and validation sets. The data that has been supplied is subjected to processing using a Convolutional Neural Network (CNN) that has been genetically tweaked.

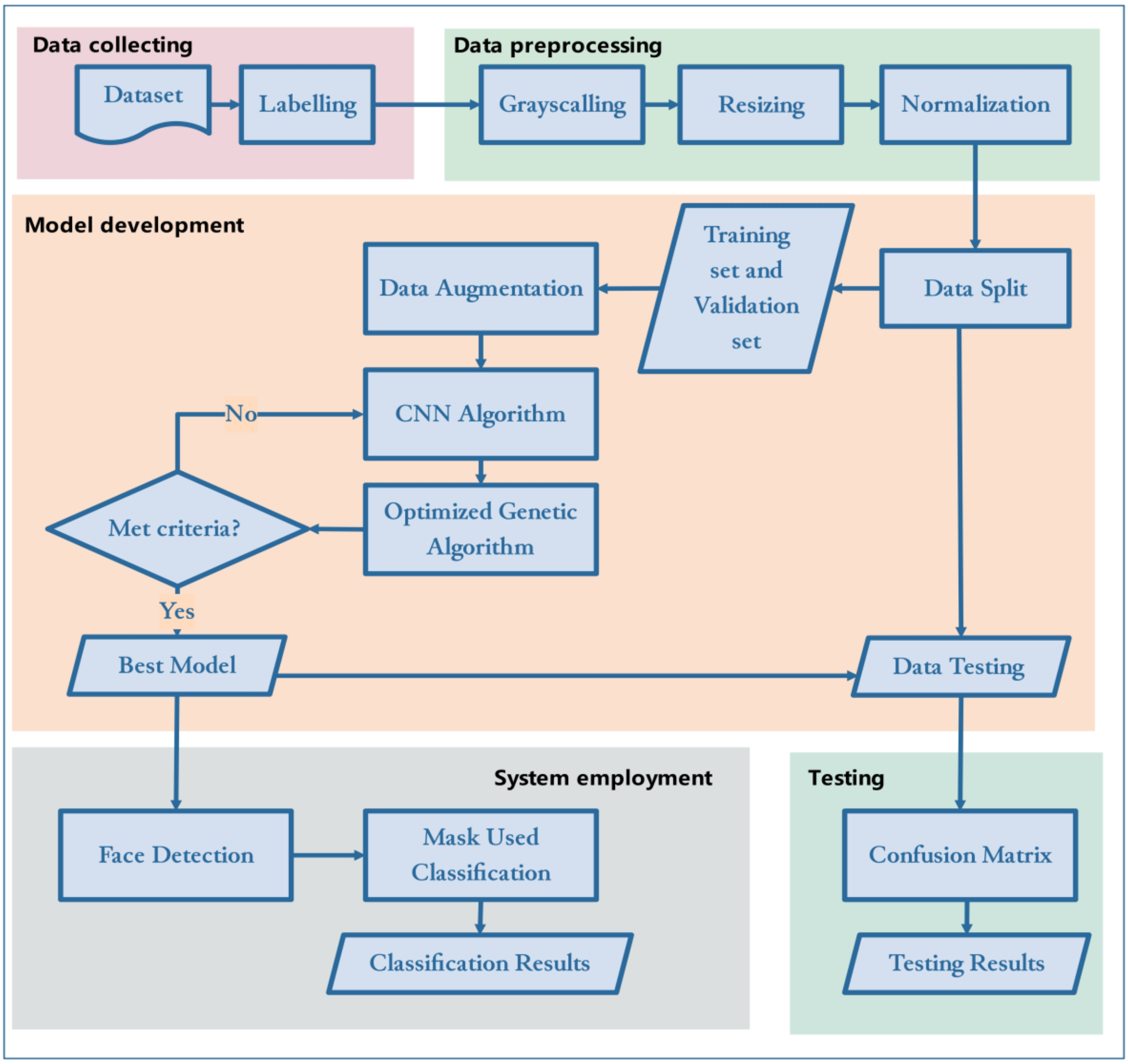

Fig. 1. Research Method

The model undergoes continuous evaluation based on established criteria, and if it does not meet the criteria, more optimization efforts are pursued. The optimal model undergoes data validation following its derivation. During the System employment phase, the model is capable of detecting faces and accurately classifying the usage of masks, subsequently presenting the obtained classification findings. The evaluation of the model's performance is conducted during the Testing phase, where a confusion matrix is utilized to derive the ultimate results.

2.1. Data Collection

The data used in this study is facial image data with three conditions, namely faces without masks (no mask), faces with the use of the right mask (right mask) and faces with the use of the wrong mask (wrong mask). This study used 3,223 data sourced from the internet (Real-World Masked Face Dataset and Larxel Dataset) as well as primary data taken directly by researchers. Data is differentiated based on mask wearing conditions with details of 1000 no mask data, 1277 right mask data, and 946 wrong mask data. An example of the data used can be seen in Fig. 2.

No Mask

Right Mask

Wrong Mask

Fig. 2. Sample Data

2.2. Data Preprocessing

Data preprocessing is a stage to process data before being processed into the algorithm used [19]. The purpose of data preprocessing is so that the data has conformity with the rules on the algorithm used [20]. Data preprocessing in this study consists of three stages, namely grayscaling, resizing, and normliztion [21]. Grayscaling is a process for changing the color of RGB images to gray. Resizing is the process of changing the pixel size of an image. In this study, all image data will be converted into a size of 96 x 96 pixels. While normalization is the process of changing pixels that have a value between 0-255 to 0-1.

2.3. Model Creation

Making models in this study was carried out to produce machine learning models that will be used to classify the use of masks. Before the model is created, the data is first divided into three parts, namely training set of 70%, validation set of 20%, and testing of 10 %. After that, the training set and validation set that will be used for training are subjected to a 20-degree rotation augmentation process, 0.2 times shear, 0.2 times zoom, and horizontal flip.

2.3.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm specifically designed to handle two-dimensional data [22]. In general, CNN architecture is divided into two parts, namely feature learning and classification [23]. Feature learning is composed of several layers such as convolution layer, activation layer, and pooling layer. The function of each layer in feature learning is to take characteristics from an input. While the classification section is composed of several layers that are fully connected with other layers and function to produce output. The architecture designed in this study consists of 4 feature extraction blocks where each block is composed of a convolution layer, activation layer, and pooling layer. In addition, the classification section is composed of an input layer / flatten, two hidden layers, and an output layer. An illustration of CNN's network architecture [24] in this study can be seen in Fig. 3.

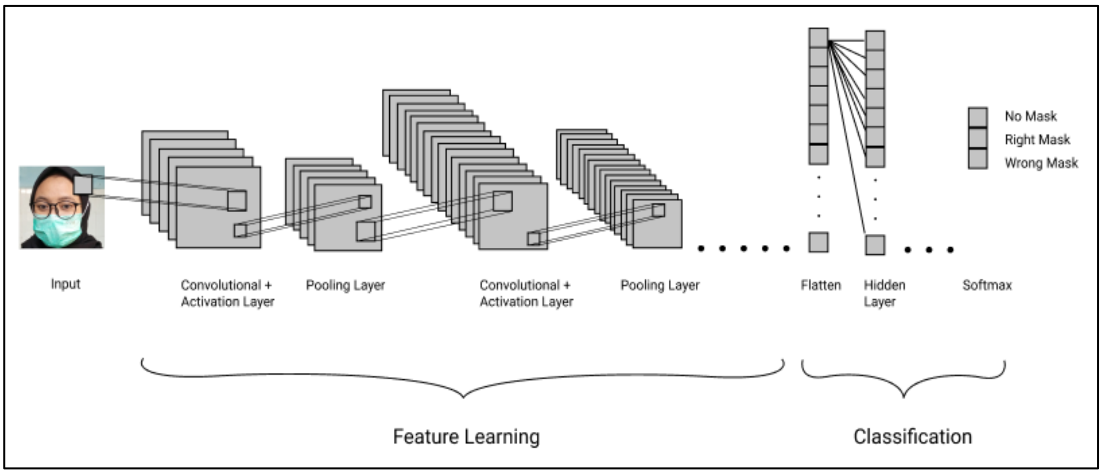

Fig. 3. CNN Architecture

The input is a preprocessed image that has a size of 96 x 96 pixels. The pixel value in the input image will be inserted into the convolution layer and convolution operations are performed with n x n size filters so that it will produce a feature map. The equation for carrying out the convolution process at the convolution layer can be seen in equation 1.

$$C_n = \sum \left(N_{(i,j)} F_{(i,j)} \right) + bF \tag{1}$$

The variable C_n represents the value of the feature map at the nth index, which is obtained by applying a filter to an image. The matrix N represents the input image, with each member denoting the brightness or color information of a pixel. The filter matrix, denoted as F, is utilized to perform image processing operations aimed at extracting specific features. The symbol bF is used to denote the refractive value within the filter, which may function as an offset or bias after convolution. The indices i and j are employed to traverse the pixel positions within the input image matrix N and the filter matrix F, respectively, enabling operations such as convolution.

After getting a feature map , the next process is the use of the activation function at the activation layer. One of the existing activation functions is ReLU which will produce positive outputs. The equation of the ReLU activation function can be seen in equation 2.

$$\mathrm{ReLU}_x = \begin{cases} 0 \text{ if}, x < 0 \\ x \text{ if}, x \geq 0 \end{cases} \quad (2)$$

Then the results of the activation layer will be entered and processed in the pooling layer. The type of pooling operation used in this study is max pooling with an equation formula as contained in equation 3.

$$\hat{c} = max\ \{C\} \quad (3)$$

After finishing with the feature extraction layer, the next step is to change the output form that was originally two-dimensional to one-dimensional and perform calculation operations on the hidden layer with the equation formula as stated in Equation 4.

$$z_j = \sum_{i=1}^{n} x_i\ w_{i,j} + b_j, j = 0,1,2,3, \dots t \quad (4)$$

The variable z_j denotes the output, which is a function dependent on multiple parameters within a neural network layer. The term b represents the bias, which is a fixed value added to the product of inputs and weights in order to adjust the activation function and impact the output of the network. The variable n denotes the overall quantity of data following the flattening procedure, whereas x signifies the specific flattened value, typically employed to convert a multi-dimensional input into a one-dimensional format. The symbol w denotes the weight parameter in a neural network, serving to scale the input and influence the magnitude of the signal. Finally, the symbol j represents the quantity of target nodes or units that exist in the completely connected layer. This layer gets input from all neurons in the preceding layer.

And the last process in the CNN Algorithm is the calculation of the probability of each class using the equation in Equation 5.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (5)$$

Information :

The symbol σ represents the likelihood of an input being assigned to a particular class, as determined by a designated activation or classification function. The symbol z represents the linear equation that corresponds to the pre-activation function. This equation calculates the weighted sum of inputs and biases prior to the application of the activation function. The symbol j is used to represent a particular class from the set of available classes, which is represented as K. In the context of the classification issue, the symbol k represents the cardinality of the set of distinct classes. This value serves as a reference point for determining the number of potential outcomes or labels that can be ascribed to a given input.

2.3.2. Algorithm Genetics

Genetic Algorithms (GA) [25], [26] are used to search for combinations of CNN hyperparameters [27], [28] that produce the most optimal accuracy. Based on the predetermined CNN architecture, in this study there are fourteen hyperparameters that will be optimized.

The hyperparameters consist of four filters, four kernel sizes, two hidden layer units, two dropouts, an activation function, and an optimizer. Details of hyperparameters and ranges of values to be searched can be seen in Table 1.

Table 1. Hyperparameter CNN

Hyperparameter	Value
Filter	32, 64, 128, 256
Kernel size	3, 5
Dropout	0.1 – 0.7
Unit hidden layer	32, 64, 128, 256
Activation layer	ReLU, ELU
Optimizer	RMSprop, Adam

When attempting to optimize the hyperparameters of a Convolutional Neural Network (CNN) using the Genetic Algorithm (GA), it is necessary to adhere to a structured methodology. The population is initially established by generating 'n' random chromosomes. Subsequently, the fitness value of each chromosome is computed in order to assess their efficacy. New populations are generated by employing selection reproduction operators, crossovers, and mutations. The aforementioned cyclical process persists until a sufficient number of chromosomes is attained, hence enabling the formation of a new population. The process of assessing chromosomes is thereafter reiterated. It is necessary to determine whether the termination condition has been satisfied. Once the user is content with the outcome, the process comes to a stop, resulting in the optimal solution. In addition, the process of generating new populations and subsequently evaluating them is repeated.

In this study the number of chromosomes used is 10, the function to calculate the fitness value of a chromosome is validation loss, the parent chromosome selection technique is rank selection or selecting parent chromosomes based on the highest fitness value, the number of parent chromosomes is 5, and the criteria for stopping or maximum generation is 8.

3. Results and Discussion

This section will describe testing of completed research. The testing phase aims to determine the success rate of the built model. Testing in this study was carried out by creating a confusion matrix to calculate the value of the evaluation indicator used, namely accuracy. The confusion matrix table in this study will display the values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) of each class, namely TM (Without Mask), MB (True Mask), and MS (Wrong Mask).

Tests were carried out on CNN models with manually tuned hyperparameter values and CNN models with hyperparameter values sought using GA optimization. Model testing conducted in this study was each repeated 10 times to determine the stability of the model against data distribution. The data used to test the model is testing data which amounts to 10% of the total data, which is 323.

3.1. CNN Confusion Matrix without GA optimization

CNN hyperparameters that are manually tuned without GA optimization have values of 32 filters at convolutional layer 1, 64 filters at convolutional layer 2, 128 filters at convolutional layer 3, 256 filters at convolutional layer 4, kernel size at 3, ReLU activation function at each convolutional layer, 128 units in hidden layer 1, 64 units in hidden layer 2, 0.5 for each dropout, and optimizer Adam. From this architecture, the CNN model produces 1,576,067 parameters and the accuracy results of each experiment can be seen in Table 2 .

Table 2. Confusion Matrix CNN

Iteration	TP			TN			FP			FN			Accuracy
	TM	*MB*	*MS*	*TM*	*MB*	*MS*	*TM*	*MB*	*MS*	*TM*	*MB*	*MS*	
1	96	111	92	215	193	214	5	12	7	7	7	10	92.56%
2	100	117	79	210	188	221	5	9	13	8	9	10	91.64%
3	93	126	84	221	182	223	5	8	7	4	7	9	93.80%
4	104	115	74	202	185	229	16	9	5	1	14	15	90.71%
5	104	107	95	211	204	214	5	6	6	3	6	8	94.73%
6	84	121	99	230	186	211	6	4	9	3	12	4	94.11%
7	99	141	59	215	164	243	7	11	6	2	7	15	92.56%
8	98	117	87	214	191	220	10	3	8	1	12	8	93.49%
9	99	131	66	206	172	241	6	15	6	12	5	10	91.64%
10	101	130	68	212	173	237	3	14	7	7	6	11	92.56%
Average													92.78%

Table 2 displays the outcomes of 10 iterations conducted for a classification job. It gives the breakdown of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) among three distinct categories: TM, MB, and MS. Throughout the 10 repetitions, the accuracy of the classification exhibits minor fluctuations. As an illustration, during the initial version, TM possessed 96 TP, MB possessed 111 TP, and MS possessed 92 TP. In a similar vein, the true negatives (TN), false positives (FP), and false negatives (FN) for the aforementioned categories were recorded as 215, 5, and 7; 193, 12, and 7; and 214, 7, and 10, respectively. Consequently, the resulting accuracy was calculated to be 92.56%. The accuracy of each iteration exhibits variability, with the minimum recorded at 90.71% during the fourth iteration and the maximum seen at 94.73% during the fifth iteration. Upon calculating the mean accuracy across all iterations, it is determined that the total average accuracy is roughly 92.78%.

3.2. Confusion Matrix CNN with GA optimization

The application of GA to the CNN architecture that was built produces hyperparameter values of 32 filters on convolutional layer 1, 256 filters on convolutional layer 2, 64 filters on convolutional layer 3, 64 filters on convolutional layer 4, kernel size of 3, ReLU activation function on each each convolutional layer, 256 units for hidden layer 1 and hidden layer 2, 0.5 for dropout 1, 0.3 for dropout 2, and Adam optimizer. From this architecture, the CNN model produces 915,395 parameters and the accuracy results for each experiment can be seen in Table 3.

Table 3. Confusion Matrix CNN-GA

Iterasi	TP			TN			FP			FN			Accuracy
	TM	*MB*	*MS*	*TM*	*MB*	*MS*	*TM*	*MB*	*MS*	*TM*	*MB*	*MS*	
1	92	122	89	222	185	219	2	6	12	7	10	3	93.80%
2	88	128	94	229	186	218	4	8	1	2	1	10	95.97%
3	86	123	96	228	188	212	5	4	9	4	8	6	94.42%
4	82	132	86	231	173	219	7	14	2	3	4	16	92.87%
5	90	128	91	228	185	219	3	6	5	2	4	8	95.66%
6	108	119	79	204	195	230	7	7	3	4	2	11	94.73%
7	86	133	88	229	178	223	4	5	7	4	7	5	95.04%
8	102	113	91	213	200	216	1	3	13	7	7	3	94.73%
9	113	118	75	201	193	224	7	8	2	2	4	11	94.73%
10	97	121	93	220	194	220	5	3	4	1	5	6	96.28%
Average													94.82%

Table 3 is a table that displays the prediction results of the CNN model with hyperparameter values obtained from optimization using GA and accuracy values from each experiment. The table shown offers a detailed summary of the observed outcomes of categorization throughout 10 iterations. The document provides a detailed breakdown of the quantities of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) pertaining to three unique categories, namely TM, MB, and MS. The performance evaluation metrics of the classifier are presented in every iteration. To provide an example, in the first iteration, TM had a cumulative score denoted as total points (TP) of 92, MB had a score of 122, and MS had a score of 89. In the present iteration, the values of true negatives (TN), false positives (FP), and false negatives (FN) for TM, MB, and MS were 222, 2, and 7; 185, 6, and 10; and 219, 12, and 3, correspondingly. As a result, the achieved accuracy rate was 93.80%. There is variability in the amount of accuracy across the iterations, with the highest recorded figure of 96.28% occurring during the 10th iteration. After doing a thorough analysis of all iterations, it has been ascertained that the mean accuracy rate stands at 94.82%.

3.3. Results of Mask Use Classification

Fig. 4 is a classified image for each class, whether the system detects one object or many objects. The results of the No Mask class classification are shown using a red box around the face, the results of the True Mask class classification are shown using green boxes around the face, and the results of the False Mask class classification are shown using a blue box around the face.

No mask Masks are correct Wrong mask

Fig. 4. Classification results

4. Conclusion

Based on the results of testing and discussion of CNN hyperparameter optimization using GA for mask use classification, it can be concluded that the use of GA to find the value of CNN hyperparameter produces a model with better accuracy than the model obtained from searching using the trial and error method. The accuracy results obtained by the model by adjusting the hyperparameter value using the trial and error method were 92.78%, while the accuracy results obtained by the model with the hyperparameter value of GA optimization results were 94.82 %. Improved accuracy between manual tuning using GA optimization by 2.04%. Although the resulting increase in accuracy is not so large, optimization using GA was able to reduce the number of parameters generated from 1,576,067 parameters to 915,395. The test results show that CNN hyperparameter optimization using GA has proven successful in making a better CNN model and simplifying the work in tuning CNN hyperparameters because GA searches for CNN hyperparameter combinations automatically so it does not need to be done manually and repeatedly. But even so, there are still shortcomings in this study that can be corrected in future studies. Because in this study the CNN algorithm distinguishes one class from another based on the presence or absence of nose, mouth, and chin features on the face, all objects that cover these features are considered masks. Therefore, the suggestion for future research is to add a process to distinguish masks from other objects so that the system can be more accurate and flexible when applied to the original state.

References

[1] K. K. Cheng, T. H. Lam, and C. C. Leung, "Wearing face masks in the community during the COVID-19 pandemic: altruism and solidarity," *Lancet*, vol. 399, no. 10336, pp. e39–e40, Apr. 2022, doi: https://doi.org/10.1016/S0140-6736(20)30918-1.

[2] L. Bubbico *et al.*, "Community Use of Face Masks against the Spread of COVID-19," *Int. J. Environ. Res. Public Health*, vol. 18, no. 6, p. 3214, Mar. 2021, doi: 10.3390/ijerph18063214.

[3] B. Qin and D. Li, "Identifying Facemask-wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19," pp. 1-15, May 2020, doi: 10.21203/rs.3.rs-28668/v1.

[4] D. J. Carragher and P. J. B. Hancock, "Surgical face masks impair human face matching performance for familiar and unfamiliar faces," *Cogn. Res. Princ. Implic.*, vol. 5, no. 1, p. 59, Dec. 2020, doi: 10.1186/s41235-020-00258-x.

[5] A. Waheed and J. Shafi, "Successful Role of Smart Technology to Combat COVID-19," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Oct. 2020, pp. 772–777, doi: 10.1109/I-SMAC49090.2020.9243444.

[6] J. Tomás, A. Rego, S. Viciano-Tudela, and J. Lloret, "Incorrect Facemask-Wearing Detection Using Convolutional Neural Networks with Transfer Learning," *Healthcare*, vol. 9, no. 8, p. 1050, Aug. 2021, doi: 10.3390/healthcare9081050.

[7] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, p. 108288, Jan. 2021, doi: 10.1016/j.measurement.2020.108288.

[8] G. Kaur *et al.*, "Face mask recognition system using CNN model," *Neurosci. Informatics*, vol. 2, no. 3, p. 100035, Sep. 2022, doi: 10.1016/j.neuri.2021.100035.

[9] M. T. Hailesellasie and S. R. Hasan, "MulNet: A Flexible CNN Processor With Higher Resource Utilization Efficiency for Constrained Devices," *IEEE Access*, vol. 7, pp. 47509–47524, 2019, doi: 10.1109/ACCESS.2019.2907865.

[10] P. Doke, D. Shrivastava, C. Pan, Q. Zhou, and Y.-D. Zhang, "Using CNN with Bayesian optimization to identify cerebral micro-bleeds," *Mach. Vis. Appl.*, vol. 31, no. 5, p. 36, Jul. 2020, doi: 10.1007/s00138-020-01087-0.

[11] A. S. Wicaksono and A. Afif, "Hyper Parameter Optimization using Genetic Algorithm on Machine Learning Methods for Online News Popularity Prediction," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 12, pp. 263–267, 2018, doi: 10.14569/IJACSA.2018.091238.

[12] Y. Guo, J. Y. Li, and Z. H. Zhan, "Efficient Hyperparameter Optimization for Convolution Neural Networks in Deep Learning: A Distributed Particle Swarm Optimization Approach," *Cybern. Syst.*, vol. 52, no. 1, pp. 36–57, 2020, doi: 10.1080/01969722.2020.1827797.

[13] R. J. Borgli, H. Kvale Stensland, M. A. Riegler, and P. Halvorsen, "Automatic Hyperparameter Optimization for Transfer Learning on Medical Image Datasets Using Bayesian Optimization," in *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, May 2019, vol. 2019-May, pp. 1–6, doi: 10.1109/ISMICT.2019.8743779.

[14] W. Y. Lee, S. M. Park, and K. B. Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," *Opt. - Int. J. Light Electron Opt.*, vol. 172, no. July, pp. 359–367, 2018, doi: 10.1016/j.ijleo.2018.07.044.

[15] N. Gorgolis, I. Hatzilygeroudis, Z. Istenes, and L. N. G. Gyenne, "Hyperparameter Optimization of LSTM Network Models through Genetic Algorithm," *10th Int. Conf. Information, Intell. Syst. Appl. IISA 2019*, vol. 00, no. c, pp. 1–4, 2019, doi: 10.1109/IISA.2019.8900675.

[16] C. Agarwal, I. Kaur, and S. Yadav, "Hybrid CNN-SVM Model for Face Mask Detector to Protect from COVID-19," in *Lecture Notes in Computational Vision and Biomechanics*, vol. 37, Springer Science and Business Media B.V., 2023, pp. 419–426, doi: 10.1007/978-981-19-0151-5_35.

[17] Yang Xu, C. Boone, and L. T. Pileggi, "Metal-mask configurable RF front-end circuits," *IEEE J. Solid-State Circuits*, vol. 39, no. 8, pp. 1347–1351, Aug. 2004, doi: 10.1109/JSSC.2004.831798.

[18] S. Zhang, Y. Xue, M. Chen, and S. Niu, "Detection of Face Wearing Mask Based on AdaBoost and YCrCb," in *2022 4th International Conference on Electrical Engineering and Control Technologies (CEECT)*, Dec. 2022, pp. 349–353, doi: 10.1109/CEECT55960.2022.10030579.

[19] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, "Big Data Preprocessing," *Cham: Springer International Publishing*, pp. 1-186, 2020, doi: 10.1007/978-3-030-39105-8.

[20] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, p. 22, Jun. 2018, doi: 10.29099/ijair.v2i1.42.

[21] R. K. Kodali and R. Dhanekula, "Face Mask Detection Using Deep Learning," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2021, pp. 1–5, doi: 10.1109/ICCCI50826.2021.9402670.

[22] Q. Wang, B. Zhao, H. Ma, J. Chang, and G. Mao, "A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional convolutional neural network with signal conversion," *J. Mech. Sci. Technol.*, vol. 33, no. 6, pp. 2561–2571, Jun. 2019, doi: 10.1007/s12206-019-0504-x.

[23] D. M. Hakim and E. Rainarli, "Convolutional Neural Network untuk Pengenalan Citra Notasi Musik," *Techno.Com*, vol. 18, no. 3, pp. 214–226, 2019, doi: 10.33633/tc.v18i3.2387.

[24] D. Bhatt *et al.*, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, p. 2470, Oct. 2021, doi: 10.3390/electronics10202470.

[25] A. Lambora, K. Gupta, and K. Chopra, "Genetic Algorithm- A Literature Review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Feb. 2019, pp. 380–384, doi: 10.1109/COMITCon.2019.8862255.

[26] S. Mirjalili, "Genetic Algorithm," in *Studies in Computational Intelligence*, vol. 780, Springer Verlag, 2019, pp. 43–55, doi: 10.1007/978-3-319-93025-1_4.

[27] T. N. Fatyanosa and M. Aritsugi, "Effects of the Number of Hyperparameters on the Performance of GA-CNN," in *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, Dec. 2020, pp. 144–153, doi: 10.1109/BDCAT50828.2020.00016.

[28] M. M. A. Monshi, J. Poon, V. Chung, and F. M. Monshi, "CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR," *Comput. Biol. Med.*, vol. 133, p. 104375, Jun. 2021, doi: 10.1016/j.compbiomed.2021.104375.