DOI: 10.14421/ijid.2021.3038 IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 10, No. 2, 2021, Pp. 72-81

Face Mask-e Wearing Detection Using Soft-Margin Support Vector Machine (SVM)

Muhammad Nur Yasir Utomo
Computer and Network Engineering Study Program
Department of Electrical Engineering
Politeknik Negeri Ujung Pandang
Makassar, Indonesia
yasirutomo@poliupg.ac.id

Fajrin Violita
Department of Health Promotion
Faculty of Public Health
Universitas Cendrawasih
Jayapura, Indonesia
fajrin.violita@alumni.ui.ac.id

Article History
Received Sep 6th, 2021
Revised Oct 22nd, 2021
Accepted Oct 22nd, 2021
Published Dec, 2021

***Abstract*— As an effort to prevent the spread of the Covid-19, various countries have implemented health protocol policies such as work-from-home, social distancing, and face mask-wearing in public places. However, monitoring compliance with the policy is still difficult, especially for the face mask policy. It is still managed by humans and is costly. Thus, this research proposes a face mask-wearing detection using a soft-margin Support Vector Machine (SVM). There are three main stages: feature selection and preprocessing, model training, and evaluation. During the first stage, the dataset of 3833 images (1915 images with face masks and 1918 images without face masks) was prepared to be used in the training stage. The training stage was conducted using SVM added with the soft-margin objective to overcome images that could not be separated linearly. At the final stage, evaluation was conducted using a confusion matrix with 10 folds cross-validation. Based on the experiments, the proposed method shows a performance accuracy of 91.7%, a precision of 90.3%, recall of 93.5%, and an F-measure of 91.8%. Our method also worked fast, taking only 0.025 seconds to process a new image. It is 7.12 times faster than Deep Learning which requires 0.18 seconds for one classification.**

Keywords*—*face region extraction; image classification; image processing; confusion matrix; covid-19

1 INTRODUCTION

The COVID-19 pandemic has impacted not only the health sector, but also the social, educational and economic sectors globally [1]. Prevention efforts are continuously being carried out by various countries by urging the public to work from home, keep a distance from other people when leaving the house and use masks in public places. [2]. However, monitoring compliance of the policy is still difficult especially for the use of masks [3].

Face mask-wearing monitoring in public places is still mostly done manually by involving humans [4]. This is not effective in terms of energy, time and cost. Moreover, humans' involvement in face mask monitoring also has the potential to make humans a spreader to the community. To solve this problem, the use of technology, especially image processing, is needed [5], [6].

Image processing has been proposed by several researchers as a solution for monitoring the use of masks. These researches combine image processing and Artificial Intelligence (AI) to classify images of people with face mask and without face mask. Research conducted by Yadav et al. shows AI-based face mask detection using Deep Learning [7]. This research process image by extracting face area as feature from complete image and process it with deep learning. It claims that the proposed method provides an accuracy of up to 95%. The same method was also used in the Militant's research to make real-time face mask detection using the Convolutional Neural Network (CNN) with an accuracy of 96% [8]. However, the use of Deep Learning requires a high computational process with large resources [3] so that these studies use GPUs for the training process.

The high computational process in the case of face mask detection is then tried to be solved in Chowdary's research which proposes face mask detection using Transfer Learning [3]. This study utilizes a pre-trained Deep Learning model so that the computational training process can be minimized. The evaluation shows that Chowdary's research has 99.92% of accuracy. The concept of Transfer Learning is also used by Bhuiyan et al. [9] with 96% accuracy. The problem is that the computational process for these studies is still carried out using Google Colab's resources. Consequently, the efficiency of the computing process on local resources still needs to be proven.

Previous research has shown that Deep Learning is very often used in the case of face mask detection because of its high level of accuracy. However, the challenge of the face mask detection system is not only about achieving high accuracy, but also having computational efficiency so that it can be implemented easily and inexpensively with minimum resource requirement in various public places. Therefore, further research related to face mask detection methods that are accurate and computationally efficient is needed.

This research proposes a face mask-wearing detection system using a Machine Learning algorithm, namely Support Vector Machine (SVM). SVM is chosen because it has a simpler computational process than Deep Learning [10] so it is easier to apply but with good accuracy [11]. Furthermore, this research also adds a soft-margin objective to the SVM to overcome images that cannot be separated linearly both in the training process and in the implementation stage.

This research is structured as follows: Section 2 shows the proposed method. Section 3 describes the results and analysis. Section 4 concludes this study and the possibilities for further research.

2 METHOD

There are four main stages that are conducted in this research: collection of datasets, pre-processing, classification with SVM and evaluation. These stages are shown in Fig. 1.

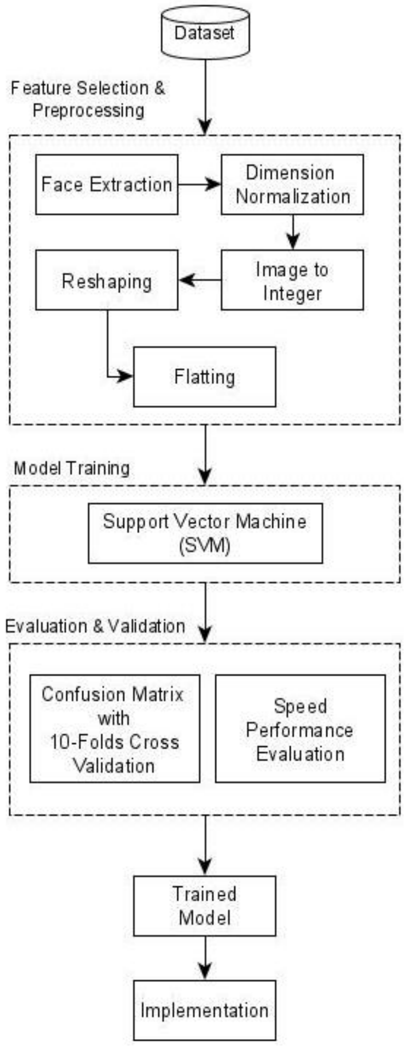

Figure 1. Proposed face mask detection method

2.1 Dataset

The dataset used is an image dataset from Kaggle, namely the Face Mask Detection (FMD) dataset. This dataset has a total of 3833 images which are divided into two categories, namely images with mask (1915 images) and without mask (1918 images).

This dataset is chosen because it has images of people with original masks (not simulated face masks), so the dataset is very representative in real world condition. The image examples for people wearing masks and not is shown in Fig. 2.

Figure 2. Dataset examples of people with and without face mask

2.2 *Feature Selection and Preprocessing*

Feature selection in this study is conducted by cropping the face area from a complete image (face region extraction). This process is carried out to reduce features and leave only an important part of the region of interest (ROI) of this study, namely the face. The face cropping process is carried out in two stages which are facial region detection and facial region cropping process.

In the first stage of feature selection, the system needs to detect facial regions in an image, this process is done using Haar Cascades Frontal Face. Haar Cascades Frontal Face is a pre-trained face detector from OpenCV that stores the coordinates of face shapes in XML format. If the facial region is successfully detected from the image, the second stage is then carried out by capturing/cropping the face from the full image using OpenCV. The illustration of the face selection process on the image data is illustrated in Fig. 3.

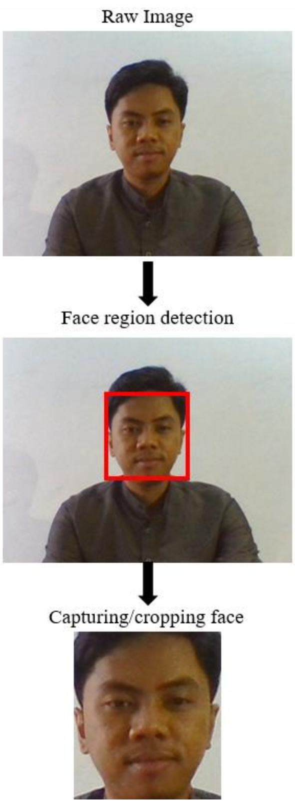

Figure 3. Feature selection process

After the face image has been successfully extracted from the complete image, the pre-processing is then carried out to normalize the image to the form that can be processed in SVM which is numerical data [2]. In this research, preprocessing includes four stages. The four stages are resizing the dimensions of the dataset image, transforming the image into a matrix of integers, reshaping the matrix into a list of arrays and simplifying the array dimensions through a flattening process, so it can be processed using SVM. All pre-processing stages can be explained as follows:

2.2.1. *Resizing the dimensions images:* The stage to uniform the size/dimensions of the image in the dataset. This is necessary because the dimensions of the images in the dataset used vary. Varied dimensions can cause bias in intraining and reduce accuracy. Uniformity of image dimensions is done by changing all image dimensions to 100x100 pixels.

2.2.2. *Image to Integer:* The process of converting an image into a matrix of integers. This process is done by utilizing the OpenCV library in Python. The result is as shown in Fig. 4.

```
[[[225 223 222]                 [[167 167 167]
  [226 224 223]                   [160 160 160]
  [228 226 225]                   [159 160 158]
  ...               . . .         ...
  [213 214 212]                   [114 122 129]
  [215 215 215]                   [163 163 163]
  [212 212 212]]                  [172 175 179]]]
```

Figure 4. Illustration of image to Integer result

2.2.3. *Reshaping:* The reshaping process aims to convert the matrix of integers into a list of integers. The reshaping results are shown in Fig. 5.

```
[[[225 223 222]                 [[167 167 167]
  [226 224 223]                   [160 160 160]
  [228 226 225]                   [159 160 158]
  ...               . . .         ...
  [213 214 212]                   [114 122 129]
  [215 215 215]                   [163 163 163]
  [212 212 212]]                  [172 175 179]]]
```

```
[225 223 222
 226 224 223
 228 226 225
    ...
 114 122 129
 163 163 163
 172 175 179]
```

Figure 5. Illustration of reshaping result

2.2.4. Flatting: This process aims to simplify the dimensions of the integer to a lower number ranging from 0 to 1, so it is easier for SVM to process. Flatting is done by dividing each integer by 255. The result illustration is shown in Fig. 6.

```
[225 223 222
 226 224 223
 228 226 225
    ...
 114 122 129
 163 163 163
 172 175 179]
```

⬇

```
[0.88235294 0.87450980 0.87058824
 0.88627450 0.87843137 0.87450980
 0.89411764 0.88627450 0.88235294
    ...
 0.44705882 0.47843137 0.50588235
 0.63921568 0.63921568 0.63921568
 0.67450980 0.68627450 0.70196078]
```

Figure 6. Illustration of flatting result

2.3 *Support Vector Machine (SVM)*

In this study, the SVM method was used as an image classification machine that can classify images of people wearing face masks or not. SVM works by finding the best hyperplane by maximizing the distance between classes [12], [13]. The distance between classes is the maximum distance between data objects in each class [14]. An illustration of determining the hyperplane in SVM is shown in Fig. 7.

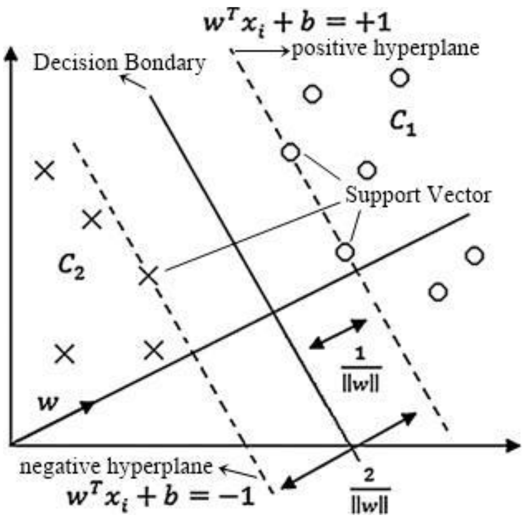

Figure 7. Illustration of how SVM defines hyperplane for classification

In Fig. 7, the patterns X and O are representations of classes 0 and 1. Pattern X describes the distribution of class 0 data and pattern O describes class 1. SVM then works to find the best dividing line (decision boundary) between the two groups of data, this process becomes the learning process in SVM [15]. The result of the process of finding the best dividing line is called a hyperplane [16].

This research uses SVM with a linear kernel. This setup is chosen because it only aims to detect two classes: images with face mask and without face mask. The SVM learning process in this study can be described. For example, the data is denoted as $x, y \in R^n$ and the label of each data is notated $y_i \in \{1,0\}$ for $i = 1,2,\dots,n$ where n m is the number of data, and $b \in R$. The class assumption used is 1 and 0 which can be completely separated by a hyperplane on dimension d defined as in Equation 1.

$$w^T x_2 + b = 0 \tag{1}$$

The pattern of x data that belongs to class 1 can be formulated as a pattern that meets Equation 2.

$$w^T x_2 + b = +1 \tag{2}$$

While the data pattern x which belongs to class 0 is formulated by Equation 3.

$$w^T x_2 + b = -1 \tag{3}$$

The largest margin of the two classes can be found by maximizing the value of the distance between the hyperplane and its closest point through Equation 4.

$$\frac{1}{\|w\|} \tag{4}$$

For linear classification cases, the SVM conducts optimization on primal space using Equation 5.

$$min \frac{1}{2}\|w\|^2$$
$$\text{for} \tag{5}$$
$$y_i(wx_i + b) \geq 1, i = 1, \dots l$$

Where x_i is the input data, y_i is the output of x_i and w, b are the parameters whose values need to be known. The goal to be achieved from the equation is the minimization of the objective function $\frac{1}{2}\|w\|^2$ or maximize quantity $\|w\|^2$ $(w^T w)$ with due regard to the limits $y_i(x_i + b) \geq 1$ for output $y_i = +1$ and $y_i(x_i + b) \leq -1$ for output $y_i = -1$.

In order to avoid bias during training due to data that is not feasible (data that cannot be separated linearly), the slack factor (ξ) is used. Slack factor is used to provide optimization on the data that causes bias or overlapping of the supposed data class constraints [17]. Equation 6 shows the use of slack in linear functions.

$$min \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} \xi$$
$$\text{for} \tag{6}$$
$$y_i(wx_i + b) + \xi \geq 1$$

Where $\xi \geq 0$ and$i = 1, \dots l$. The solution to the data bias problem using the help of slack is also known as the soft margin optimization method [18]. Fig. 8 shows the illustration of the use of slack to solve non feasible data problems or biased data.

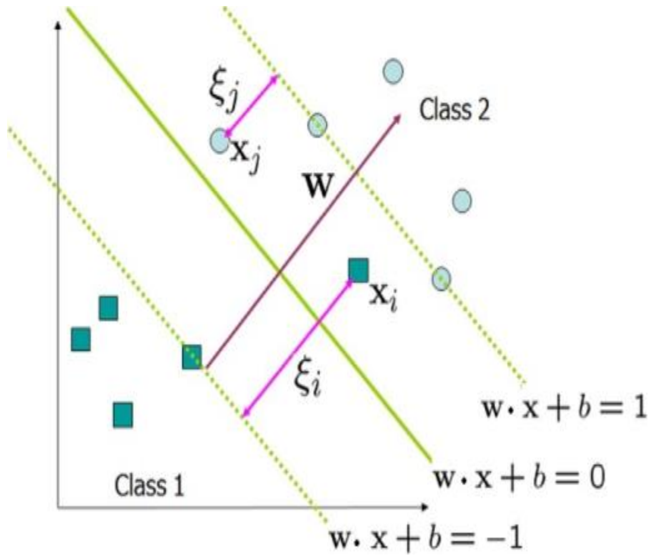

Figure 8. Soft-margin technique with slack variable in SVM

In addition to the problem of biased data, in real-world cases, non-linear data are often found. Non-linear data in this case refer to data that cannot be separated correctly by ordinary linear hyperplane [19]. To solve this problem, the Lagrangian method is used, and it can be seen in Equation 7.

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{l} \alpha_i [y_i (w^t x_i + b) - 1] \quad (7)$$

Where α_i is a non-negative variable, called the Lagrangian Multiplier. The purpose of (7) is to find the saddle point of the Lagrangian function $J(w, b, \alpha)$. The saddle point in this case is the saddle point or balance between the two data classes. To achieve this equilibrium point, the Lagrangian function must minimize the variables w and b, and at the same time the variable must be maximized for the variable α. To get an optimized solution for the dual problem equation from the Lagrangian function, Equation 7 can be improved to Equation 8.

$$\begin{aligned} w^T w &= \sum_{i=1}^{l} \alpha_i y_i w^T x_i \\ &= \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i{}^T x_j \end{aligned} \quad (8)$$

The final equation is obtained, and it can be seen in Formula 9.

$$max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j x_i{}^T x_j$$

$$\text{for} \quad (9)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0$$

Where $\alpha_i \geq 0$ and $i = 1, \ldots, l$. This equation is a quadratic programming solution with a linear constraint. In the classification method using SVM, algorithm training with this solution is also known as convex optimization. In this research, SVM is used as a classification algorithm to detect images of people with face masks or not.

2.4 Experimental Setup and Evaluation

Experiment in this research is carried out using standard specification of a personal computer shown in Table 1.

Table 1. The Hardware Used

No	*Hardware Type*	*Specification*
1	Processor	Intel Core i5
2	Memory	8 GB DDR3
3	Hard-disk	1 TB HDD

In addition to the hardware, several software are also used to perform the experiment. Software used in this research are shown in Table 2.

Table 2. The Software Used

No	*Software*	*Version*
1	Windows OS	Version 10 64-bit
2	Python	3.7.6
3	OpenCV	4.4.0
4	Scikit-Learn	0.22.2
5	Numpy	1.19.4
6	Pickle	4.0
7	Time	3.7.6

The experimental setup in Table 1 and Table 2 is used to train and test the classification model. The evaluation itself is carried out using a confusion matrix with 10-fold cross validation. The confusion matrix is used to measure the values of accuracy, precision, recall and F-measure of the classification model while 10-folds cross validation is used as validation measurement [20]. The illustration of the confusion matrix is shown in Fig. 9.

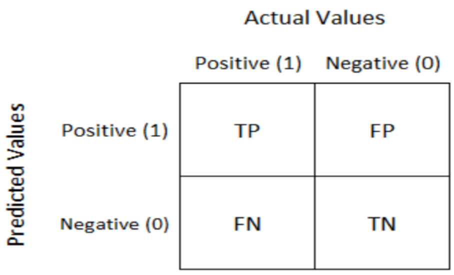

Figure 9. Confusion matrix

Based on the confusion matrix in Fig. 9, the accuracy value of the proposed method is measured as in Equation 10.

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

TP in Formula 10 is a true positive prediction, TN is a true negative, P is a positive class population and N is a negative class population. The second evaluation to be measured is precision which is calculated using Equation 11.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

The third evaluation measurement is Recall. Recall is calculated by finding the percentage of the truth of the positive class prediction compared to. The total data with an actual positive label. Recall calculation is done by Equation 12.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

After the precision and recall are known, the F-measure is then calculated using Equation 13.

$$F - measure = \frac{2\ X\ Precision\ X\ Recall}{Precision + Recall} \quad (13)$$

This research also compares computational efficiency in terms of training and classification speed between the proposed method and CNN (Deep Learning). The calculation process of the speed performance is done using the Time library in Python. The process is shown in Fig. 10.

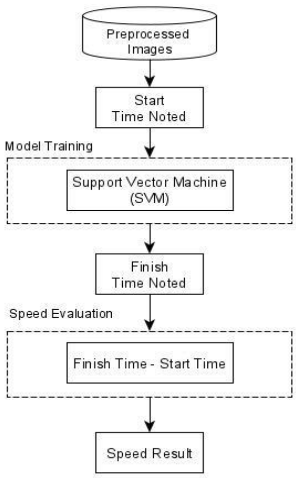

Figure 10. Speed Evaluation Process

Fig. 10 shows that time calculation starts right before the training process is executed and stops right after it. The same concept is also used to measure prediction speed of the model. By knowing the speed performance, recommendations for the best method for real world implementation can be concluded.

2.5 *Implementation*

In addition to developing the face mask detection method, this research also proposes ways to implement it.

This is important so that the proposed method can actually be used correctly and efficiently. Implementation is done by combining three main components, namely the trained model, OpenCV and Haar Cascades Frontal Face. Each component can be explained as follows:

- The trained model used is an image classification model based on the results of training and evaluation of the proposed method. The trained model is used as a classification model on new images captured in real time.
- OpenCV is used to access the camera and take real-time image data. In addition, OpenCV also plays a role in the conversion of image to integer [21].
- Haar Cascades Frontal Face which is a pre-trained face detector in the form of XML file provided open source by the OpenCV developers [22]. This file is useful for detecting faces in an image [23].The third component is combined with the flow as shown in Fig. 11.

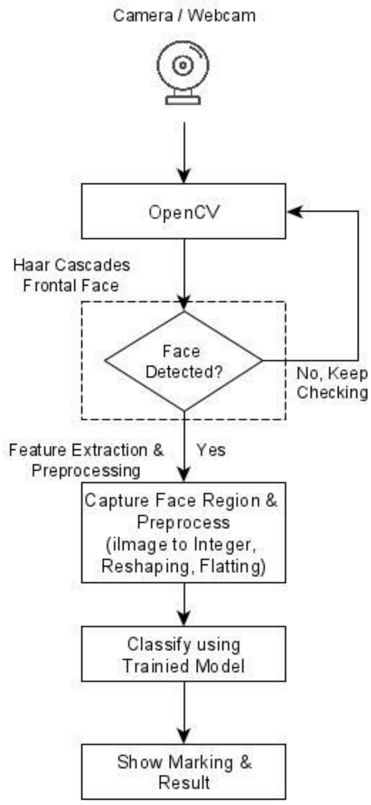

Figure 11. Implementation flow of the proposed method

Fig. 11 above shows the implementation flow of the proposed method. The flow in Fig. 11 can be described as follows:

- Camera / webcam accessed using OpenCV to get images in real-time.
- Detection of faces in the image is done using Haar Cascades Frontal Face. This process is part of feature selection and carried out continuously as long as the camera is open.
- Feature extraction is conducted if a face is detected from the full image. The detected face image is

captured, extracted and converted into an integer, reshaping and flatting (preprocessing).

- The results of preprocessing are then classified using a trained model from the proposed method. The classification results are divided into two categories: with face mask or without face mask image.
- After the classification process is complete, the results are displayed on the camera image in real time.

3 RESULT AND DISCUSSION

3.1 Evaluation of Proposed Method (SVM)

The proposed method is evaluated in terms of accuracy, precision, recall and F-measure. The evaluation is conducted using a confusion matrix with 10-folds cross validation, and the result is shown in Table 3.

Table 3. Cross Validation Result of Proposed Face Mask Detection Method

Fold	*Accuracy*	*Precision*	*Recall*	*F-measure*
1	0.933	0.906	0.958	0.931
2	0.916	0.900	0.945	0.922
3	0.899	0.879	0.923	0.901
4	0.912	0.900	0.930	0.915
5	0.917	0.895	0.941	0.917
6	0.916	0.901	0.938	0.919
7	0.908	0.895	0.931	0.913
8	0.922	0.920	0.932	0.926
9	0.927	0.923	0.935	0.929
10	0.913	0.909	0.914	0.911
Max	0.933	0.923	0.958	0.931
Min	0.899	0.879	0.914	0.901
Avg.	**0.917**	**0.903**	**0.935**	**0.918**

Table 3 shows the evaluation results conducted with 10 folds cross validation. Based on the table, it can be seen that the accuracy value of the proposed method is very good, ranging from 89.9% to 93.3%. The precision evaluation value also shows satisfactory results, which is ranging from 87.9% to 92.3%. In terms of recall, the proposed method shows a convincing performance with a minimum value of 93.5% and a maximum of 95.8%. The final evaluation is then carried out by calculating the average value of each evaluation component. The results are shown in Fig. 12.

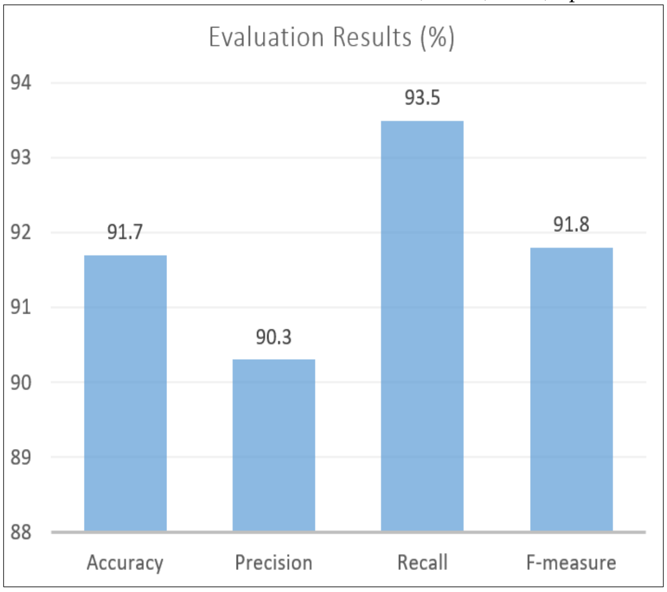

Figure 12. Evaluation results of proposed method

Fig. 12 shows that the proposed method gets 91.7% of accuracy, 90.3% of precision, 93.5% of recall and 91.8% of F-measure. These results indicate that the proposed model works very well and consistently gets results above 90% for all evaluation categories.

3.2 Comparison with the State of the Art Method (CNN)

To get an idea of how well the proposed method works, this research then compares the proposed method with the current state of the art method, namely Deep Learning [24]. Research by Chowdary et. al [3] is used as a baseline. This research uses the Deep Learning method (CNN) to conduct face mask detection. Comparison between the proposed method and Deep Learning is calculated in three aspects: the accuracy level, training and prediction speed. The comparison of accuracy can be seen in Fig. 13.

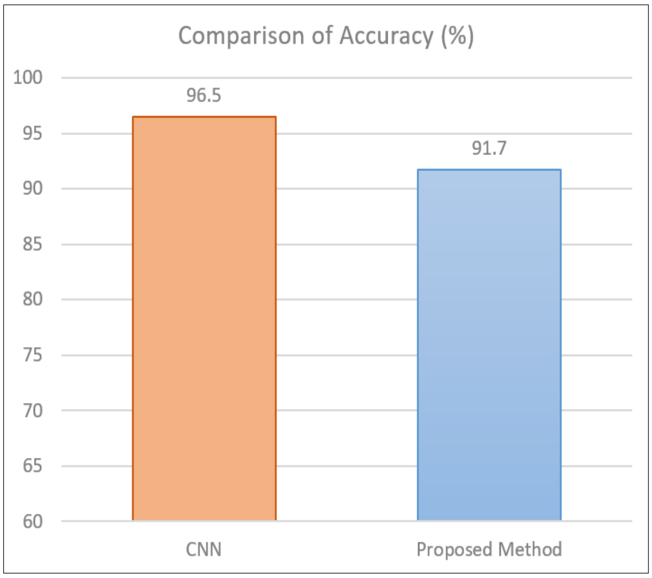

Figure 13. Comparison of accuracy between proposed method and CNN

Based on Fig. 13 above, it can be seen that CNN/Deep Learning has a higher accuracy than the proposed method, which is 96.5% compared to 91.7%. However, the training process performance of the proposed method is much faster than CNN as shown in Fig. 14.

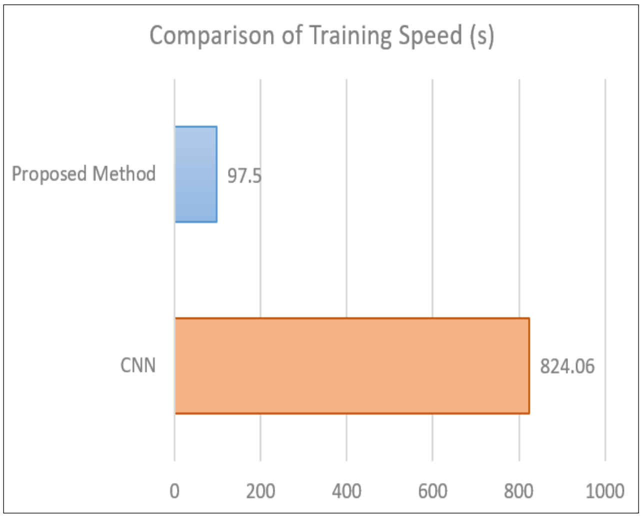

Figure 14. Training speed comparison between proposed method and CNN (Smaller is Better)

Based on Fig. 14 above, it can be seen that the training speed of CNN takes up to 824.06 seconds with 10 Epoch setup (number of epoch recommendations for deep learning [25], [26]). The CNN training time is much longer than the training speed of the proposed method, which is only 97.5 seconds.

In addition to the slow training process time, CNN also requires a longer time than the proposed method in terms of classifying face masks on new images. This is shown in Fig. 15 which illustrates the difference in the prediction speed of each method.

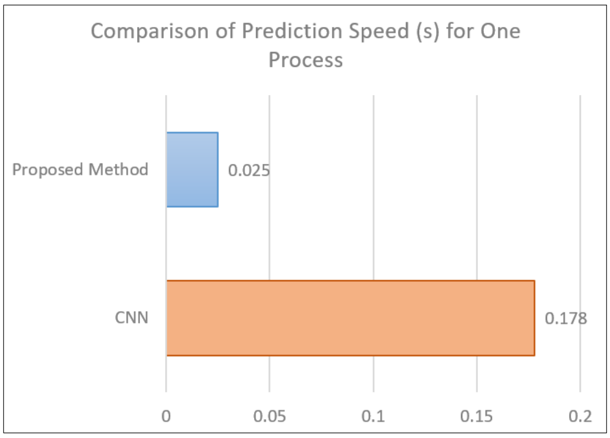

Figure 15. Prediction speed comparison between proposed method and CNN (Smaller is Better)

Based on Fig. 15, the proposed method can work much faster in performing face mask detection on an image. The proposed method manages to get a speed of 0.025 seconds for one face mask detection process, while CNN takes up to 0.178 seconds. The proposed model is 7.12 times faster than CNN. These results indicate that the computational process of the proposed method is lighter and works faster. Therefore, it is more likely to be used in real implementations.

3.3 Implementation of Proposed Face Mask Detection

The implementation of the face mask detection system in this study is implemented by combining OpenCV library and the trained model. In this research, OpenCV is used to open the webcam, and the images that appear are then classified using the trained model. The results are shown in Fig. 16 to Fig. 18.

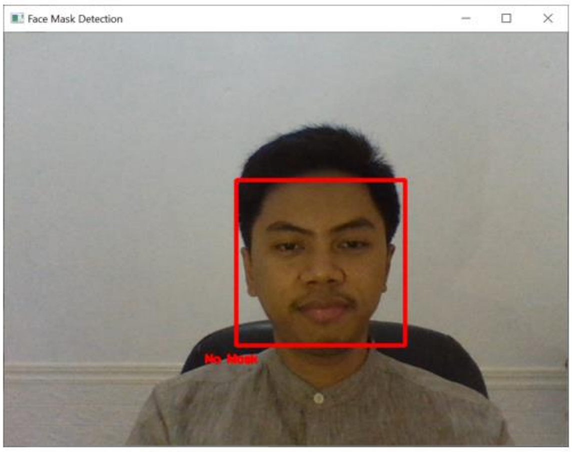

Figure 16. Detection of person not wearing face mask

Fig. 16 above shows an example detection of a person who is not wearing a face mask taken in real time via a webcam. An example of a person with face mask detection is shown in Fig. 17.

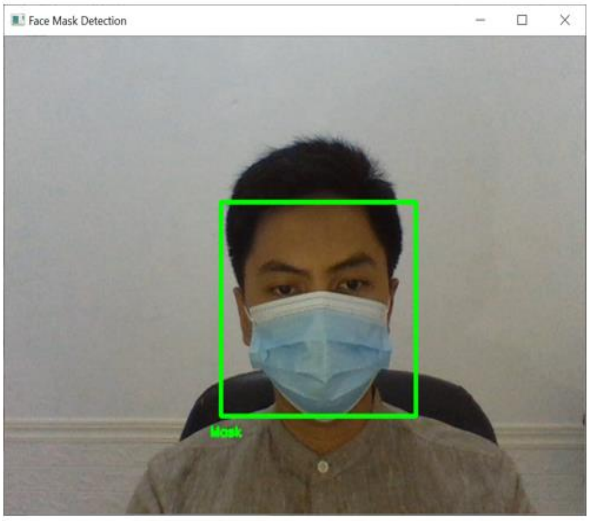

FIgure 17. Detection of person wearing face mask

This study also tested the implementation of the image with two people simultaneously. The implementation results show that the proposed method can also work very well on multi face mask detection as shown in Fig. 18.

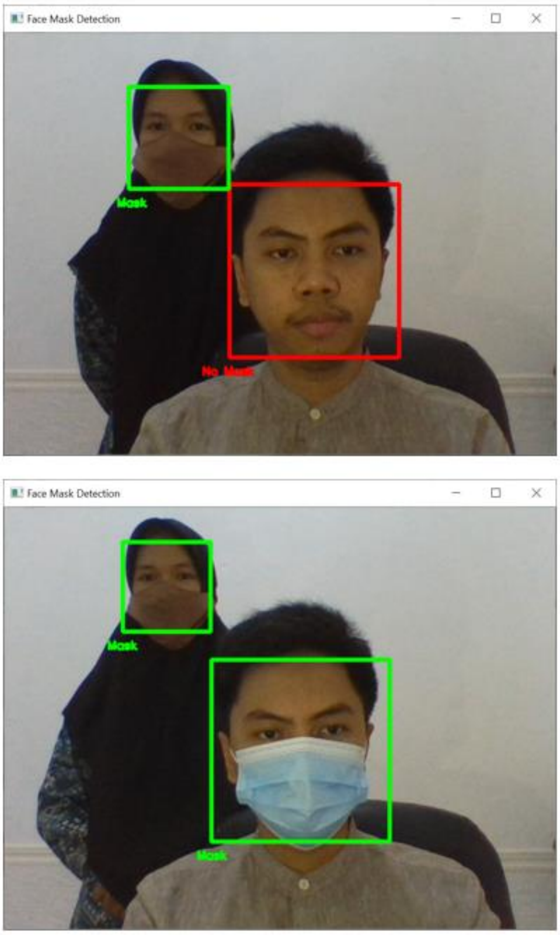

Figure 18. Multi-face mask detection

4 CONCLUSION

This research proposes a face mask detection system to support the efforts to prevent the spread of COVID-19/coronavirus. The proposed method uses the SVM as an image classification algorithm that can detect whether an image contains a picture of people wearing or not wearing a face mask. Based on the experiments that have been conducted, the proposed method succeeded in obtaining 91.7% of accuracy, 90.3% of precision, 93.5% of recall and 91.8% of F-measure.

In addition, the proposed method also shows fast computational performance. The speed evaluation shows that the proposed method only takes 0.025 seconds to classify one image. It is 7.12 times faster than Deep Learning which needs 0.178 second to do the same task. It is very possible to implement a proposed method for monitoring the locations of public facilities with minimum resource requirements.

Further research can be done by adding complexity calculation of the model, looking for other variations of pre-processing and different algorithms to get more comprehensive analysis.

AUTHOR'S CONTRIBUTION

Fajrin Violita realized that the health industry needed a breakthrough technology to monitor the use of face masks in public places to prevent the spread of COVID-19. Muhammad Nur Yasir Utomo then proposed a solution: a face mask detection system (image processing) using SVM. Fajrin Violita searched for the dataset needed. Muhammad Nur Yasir Utomo then performed the coding needed to train face mask detection system using SVM. At the end, both authors carried out the experiment together. They contributed to this paper.

COMPETING INTERESTS

Muhammad Nur Yasir Utomo and Fajrin Violita as authors of this paper stated that the paper is free from conflict of interests (COI) or competing interests (CI).

ACKNOWLEDGMENT

This work is partially supported by IDalamat.com and Politeknik Negeri Ujung Pandang (PNUP), Makassar, Indonesia.

REFERENCES

[1] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting Against COVID-19 - A Novel Deep Learning Model Based on YOLO-v2 with ResNet-50 for Medical Face Mask Detection," *Sustain. Cities Soc.*, vol. 65, pp. 1–8, 2021, doi: https://doi.org/10.1016/j.scs.2020.102600.

[2] M. N. Y. Utomo, M. Sudaryanto, and K. Saddhono, "Tools and Strategy for Distance Learning to Respond COVID-19 Pandemic in Indonesia," *Ing. des Syst. d'Information*, vol. 25, no. 3, pp. 383–390, 2020, doi: 10.18280/isi.250314.

[3] G. Jignesh Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face Mask Detection Using Transfer Learning of InceptionV3," *Lect. Notes Comput. Sci. (Artificial Intell. Bioinformatics) Springer*, vol. 12581, no. 1, pp. 1–11, 2020, doi: 10.1007/978-3-030-66665-1_6.

[4] J. Yu and W. Zhang, "Face Mask-wearing Detection Algorithm Based on Improved Yolo-v4," *Sensors*, vol. 21, no. 9, pp. 1–21, 2021, doi: 10.3390/s21093263.

[5] 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," *J. Int. Meas. Confed.*, vol. 167, 2021, doi: 10.1016/j.measurement.2020.108288.

[6] C. Z. Basha, B. N. L. 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, pp. 1–8, 2021, doi: 10.4108/eai.8-1-2021.167843.

[7] S. Yadav, "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 7, pp. 1368–1375, 2020, doi: 10.22214/ijraset.2020.30560.

[8] S. V. Militante and N. V. Dionisio, "Real-Time Facemask Recognition with Alarm System using Deep Learning," *IEEE Control Syst. Grad. Res. Colloq.*, pp. 106–110, 2020, doi: 10.1109/ICSGRC49013.2020.9232610.

[9] M. R. Bhuiyan, S. A. Khushbu, and M. S. Islam, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," *2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020*, pp. 1–5, 2020, doi: 10.1109/ICCCNT49239.2020.9225384.

[10] B. Xu, A. Shirani, D. Lo, and M. Amin Alipour, "Prediction of Relatedness in Stack Overflow: Deep Learning vs. SVM," *Proc. 12th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas.*, pp. 1–10, 2018, doi: https://doi.org/10.1145/3239235.3240503.

[11] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith, and N. Kumar, "Image Classification using SVM and CNN," *2020 Int.*

Conf. Comput. Sci. Eng. Appl., pp. 1–5, 2020, doi: https://doi.org/10.1109/ICCSEA49143.2020.9132851.

[12] M. A. I. Mahmoud and H. Ren, "Forest Fire Detection and Identification using Image Processing and SVM," *J. Inf. Process. Syst.*, vol. 15, no. 1, pp. 159–168, 2019, doi: 10.3745/JIPS.01.0038.

[13] H. I. Dino and M. B. Abdulrazzaq, "Facial Expression Classification Based on SVM, KNN and MLP Classifiers," *2019 Int. Conf. Adv. Sci. Eng. ICOASE 2019*, pp. 70–75, 2019, doi: 10.1109/ICOASE.2019.8723728.

[14] L. Hu and J. Cui, "Digital Image Recognition Based on Fractional-order-PCA-SVM Coupling Algorithm," *J. Int. Meas. Confed.*, vol. 145, no. February, pp. 150–159, 2019, doi: 10.1016/j.measurement.2019.02.006.

[15] Y. Guo, X. Jia, and D. Paull, "Effective Sequential Classifier Training for SVM-Based Multitemporal Remote Sensing Image Classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3036–3048, 2018, doi: 10.1109/TIP.2018.2808767.

[16] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "A Novel Face Detection Approach using Local Binary Pattern Histogram and Support Vector Machine," *2018 Int. Conf. Adv. Syst. Electr. Technol. ICASET 2018*, pp. 28–33, 2018, doi: 10.1109/ASET.2018.8379829.

[17] W. Shu and K. Cai, "A SVM Multi-Class Image Classification Method Based on de and KNN in Smart City Management," *IEEE Access*, vol. 7, pp. 132775–132785, 2019, doi: 10.1109/ACCESS.2019.2941321.

[18] M. L. Dantas Dias and A. R. Rocha Neto, "Training Soft Margin Support Vector Machines by Simulated Annealing: A Dual Approach," *Expert Syst. Appl.*, vol. 87, pp. 157–169, 2017, doi: 10.1016/j.eswa.2017.06.016.

[19] J. F. Tuttle, L. D. Blackburn, and K. M. Powell, "On-line Classification of Coal Combustion Quality using Nonlinear SVM for Improved Neural Network NOx Emission Rate Prediction," *Comput. Chem. Eng.*, vol. 141, p. 106990, 2020, doi: 10.1016/j.compchemeng.2020.106990.

[20] W. Xiong, J. Xu, Z. Xiong, J. Wang, and M. Liu, "Degraded Historical Document Image Binarization using Local Features and Support Vector Machine (SVM)," *Opt. - Int. J. Light Electron Opt.*, vol. 164, pp. 218–223, 2018, doi: 10.1016/j.ijleo.2018.02.072.

[21] H. Adusumalli, D. Kalyani, R. K. Sri, M. Pratapteja, and P. V. R. D. P. Rao, "Face Mask Detection Using OpenCV," *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob. Networks, ICICV 2021*, pp. 1304–1309, 2021, doi: 10.1109/ICICV50876.2021.9388375.

[22] Z. Arya and V. Tiwari, "Automatic Face Recognition and Detection Using OpenCV, Haar Cascade and Recognizer for Frontal Face," *Int. J. Eng. Res. Appl. www.ijera.com*, vol. 10, no. 6, pp. 13–19, 2020, doi: 10.9790/9622-1006051319.

[23] N. Prasad, B. Rajpal, K. K. Rao Mangalore, R. Shastri, and N. Pradeep, "Frontal and Non-Frontal Face Detection using Deep Neural Networks (DNN)," *Int. J. Res. Ind. Eng.*, vol. 10, no. 1, pp. 9–21, 2021.

[24] S. Balaji, B. Balamurugan, T. A. Kumar, R. Rajmohan, and P. P. Kumar, "A Brief Survey on AI Based Face Mask Detection System for Public Places," *Irish Interdiscip. J. Sci. Res. 2021*, vol. 5, no. 1, pp. 108–117, 2021.

[25] S. Zimmerman, C. Fox, and U. Kruschwitz, "Improving Hate Speech Detection with Deep Learning Ensembles," *Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval.*, pp. 2546–2553, 2019.

[26] S. Abu Jwade, A. Guzzomi, and A. Mian, "On Farm Automatic Sheep Breed Classification using Deep Learning," *Comput. Electron. Agric.*, vol. 167, p. 105055, 2019, doi: 10.1016/j.compag.2019.105055.