CommIT Journal 16(2), 159–166, 2022

Modified Multi-Kernel Support Vector Machine for Mask Detection

Muhammad Athoillah[1]*, Evita Purnaningrum[2], and Rani Kurnia Putri[3]
[1]Statistics Department, Faculty of Science and Technology, Universitas PGRI Adi Buana Surabaya 60234, Indonesia
[2]Management Department, Faculty of Economic and Business, Universitas PGRI Adi Buana Surabaya 60234, Indonesia
[3]Mathematics Education Department, Faculty of Science and Technology, Universitas PGRI Adi Buana Surabaya 60234, Indonesia
Email: [1]athoillah@unipasby.ac.id, [2]purnaningrum@unipasby.ac.id, [3]rani@unipasby.ac.id

***Abstract*—Indonesia is one of the countries most affected by the Coronavirus pandemic with millions confirm cases. Hence, the government has increased strict procedures for using face masks in public areas. For this reason, the detection of people wearing face masks in public areas is needed. Face mask detection is a part of the classification problem. Thus Support Vector Machine (SVM) can be implemented. SVM is still known as one of the most powerful and efficient classification algorithms. The research aims to build an automatic face mask detector using SVM. However, it needs to modify it first because it only can classify linear data. The modification is made by adding kernel functions, and a Multi-kernel approach is chosen. The proposed method is applied by combining various kernels into one kernel equation. The dataset used in the research is a face mask image obtained from Github. The data are public datasets consisting of faces with and without masks. The results present that the proposed method provides good performance. It is proven by the average value. The values are 83.67% for sensitivity, 82.40% for specificity, 82.00% for precision, 82.93% for accuracy, and 82.77% for F1-score. These values are better than other experiments using single kernel SVM with the same process and dataset.**

***Index Terms*—Modified Multi-Kernel, Support Vector Machine (SVM), Mask Detection**

I. Introduction

THE coronavirus (COVID-19) outbreak has spread throughout the world and transformed into several new variants since its emergence. Although researchers, practitioners, and the government suppress the growth rate of positive confirmed cases and the number of virus spreads, the faster spread of the new variant makes the pandemic even more alarming. Indonesia is one of many countries affected by the pandemic, with 5.8 million confirmed cases [1]. One of the reasons for this situation is a low rate of vaccination in Indonesia for the third vaccination with only 6.68% [2]. Accordingly, other efforts are needed to reduce the COVID-19 death rate, including increasing strict procedures and using face masks. In addition, World Health Organization (WHO) recommends that people in public areas wear double masks to suppress the spread of the virus. For this reason, detecting people wearing face masks in public areas is essential [3]. Regular use of face masks is crucial to help to prevent the spread of infection and contracting viruses transmitted in the air or through droplets [4].

Exploration of technology to detect mask users in public places is needed to identify easier people who do not comply with the rules. Various studies have applied many methods to test the program's effectiveness for face detection [5]. Specifically, there are more and more studies to make automatic mask detection. For example, the previous studies show a detector of face masks using Single Shot MultiBox Detector and MobileNetV2 [6], deep learning for face mask detection to reduce Coronavirus spread [7], and a similar face mask detector but with a semantic segmentation method [8]. There is a study using the Squeeze and Excitation (SE)-YOLOv3 method to detect the use of masks with relatively balanced effectiveness and efficiency in the case of real-time datasets with an mAP value of 8.6% higher than YOLOv3 [9]. Then, users are detected without masks in real-time using a deep learning method based on a dataset from Live Video processed in a Raspberry Pi 4 [10]. Similarly, a model is designed with the PCS algorithm to recognize a human face without a mask with an accuracy of 96.25% and face recognition with a mask with an accuracy of

Received: Nov. 08, 2021; received in revised form: April 05, 2022; accepted: April 06, 2022; available online: Sep. 14, 2022.
*Corresponding Author

68.75% [11]. Last, the transfer learning model applies Inception V3 to detect masked faces [12].

The classification model or method has various types of models, including K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Deep Learning (DL), or Artificial Neural Network (ANN). The KNN, SVM, ANN, and DL algorithms have relatively stable performance in detecting users with masks. Moreover, KNN and SVM are relatively faster in the dataset training process [13]. A previous study has proven that the accuracy of masked face detection using SVM is 99% and above [14]. Then, a transfer learning method is applied to classify masked and unmasked faces with a combination of the MobileNet-V2 model with SVM. It has a good performance of 97.11% [15]. An enormous number of studies have been actively applied in certain domains of SVM and their application in various fields of science. Today, SVM is still known as one of the most powerful and efficient classification algorithms. This algorithm has the strongest regression and classification mathematical model. This powerful foundation lets new guidance for further research in the prodigious field of regression and classification for years [16, 17].

The purpose of this framework is to provide another reference for SVM to detect mask users. Traditionally, SVM is a linear classification method. Thus, it has to be modified using a kernel function to classify non-linear data. However, there are many kinds of kernels with different characteristics, so it is difficult to determine the best kernel for every classification problem. This problem has been proven by some research [18]. It aims to detect lung cancer on CT scan datasets using various types of single kernel SVM, namely Linear kernel, Radial Basis Function (RBF) kernel, and Polynomial kernel. The results show that the three models provide different accuracy, and the RBF kernel SVM ranks the lowest with only 55.55%. Meanwhile, Linear and Polynomial kernels have 78.33% and 80.01%, respectively. In addition, the previous research studies empirically the different kernels of SVM. It provides different results in the case of facial expression recognition. SVM Linear Kernel has the worst accuracy with 49.93% compared to the quadratic kernel and cubic kernel with an accuracy of 77.51% and 97.11%, respectively [19]. Based on that problem, the research intends to provide another alternative to overcome confusion in choosing the best kernel for every problem by combining existing kernels into a unified equation called Multi-kernel SVM. In simple terms, this approach involves combining various kernels into a single formula.

(a) (b)

Fig. 1. Example of the face without a mask (a) and face with a mask (b) [9].

II. Research Method

A. Dataset

The dataset used in the research is a face mask image obtained from Github (https://github.com/prajnasb and https://www.kaggle.com/andrewmvd). The data are a public dataset consisting of faces with and without masks. Figure 1 shows an example of a face with mask (Fig. 1a) and without a mask (Fig. 1b). The number of datasets is 1,500, with details of 750 with masks and 750 without masks. The dataset is divided into training data with 1,350 images (90%) and testing data with 150 images (10%).

B. Data Processing

The researchers collect the dataset of face mask images as mentioned previously. Then, data are transformed into matrix calculations using histogram formulas. An image histogram is a collection of graphs that comprehensively describe the distribution of the intensity values of each pixel in an image or a certain part of the image. A histogram describes the relative frequency of occurrence (relative) of the intensity in the image. Moreover, a histogram also represents a lot about the contrast and the brightness of an image [20]. A histogram is used because this approach is very conscious but powerful, especially if it is used to identify mask images with close color blocks and the same intensity of spread for each mask image [21].

C. Multi-Kernel Support Vector Machine (SVM)

SVM is one of the machine learning algorithms used for classification problems. The SVM algorithm has been applied to various tasks, including biology, medicine, and meteorology [22–24]. A simple explanation of the thinking concept of the SVM algorithm is an attempt to find the best hyperplane that separates the two data classes in the input space. The pattern of the two classes is translated as $(+1)$ class and (-1) class which can be obtained from various alternative boundaries (discrimination limits). Meanwhile, the margin line is translated as the distance formed between the hyperplane and the closest pattern from each class that has been separated. This closest pattern is called a support vector. Briefly, it can be explained that the effort to find the hyperplane position is the core of the learning process in SVM. Traditionally, SVM is an algorithm that can only classify linear data. Soft margin SVM cannot find the separator on the hyperplane, so it cannot have high accuracy and be generalized properly. Therefore, it takes a kernel to transform data into a higher-dimensional space called kernel space which is helpful for separating data linearly [25]. The kernel function of $K(x_i, x_j)$ is formulated as $K(x_i, x_j) =< \phi(x_i).\phi(x_j) >$.

The ϕ transforms data from the original dimension to a higher dimension, resulting in the new dimension where the data can be separated linearly by the formed hyperplane. In the research, the researchers combine various kernel functions, namely the third-order polynomial kernel function, the linear kernel function, and the RBF kernel function, as the most commonly used kernel function. Here are the equations of the three functions.

- Linear Kernel:
$$K(x_i, x_j) = x^T.x^{'}. \tag{1}$$
- Polynomial Kernel:
$$K(x_i, x_j) = (x.x^{'} + 1)^d. \tag{2}$$
- RBF Kernel:
$$K(x_i, x_j) = \exp(\frac{\|x_i - x_j\|^2}{2\sigma^2}). \tag{3}$$

As explained earlier, the main idea of Multi-kernel learning is to combine several kernel functions of $K(x, x^{'})$ so that the kernel functions can be formulated as $K(x, x^{'}) = \sum_{m=1}^{M} d_m K_m(x, x^{'})$ which subjects to $d_m \geq 0, \sum_{m=1}^{M} d_m = 1$. The M is the number of kernel functions, while d_m is the weight chosen to represent data [26]. This function becomes a reference or basis for combining the three kernels in Eqs. (1)–(3). Then, it can be implemented in the system. Furthermore, in a single kernel function, the optimal distance from the hyperplane is obtained by solving the equation of $\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$, which is subject to the following Eqs. (4) and (5).

$$y_i(x_i.w + b) \geq 1 - \xi_i \ \forall i, \tag{4}$$
$$\forall \geq 0 \ \forall i. \tag{5}$$

The w is the normal line (a weight vector perpendicular to the hyperplane). At the same time, ξ and C are additional variables as a boundary between maximizing margins and reducing the number of errors during the classification process. Furthermore, in Multi-kernel functions, the decision function of $(x_i.w) + b = f(x) + b$ becomes $\sum_{m=1}^{M} f_m(x) + b$. The b is the bias (position of the plane relative to the center of the coordinates), and each f_m is a different Reproducing Kernel Hilbert Space (RKHS) function based on the K_m kernel function [27]. So, the optimal hyperplane can be obtained by solving the equation of $\min_d J(d)$, which is subject to $d_m \geq 0, \sum_{m=1}^{M} d_m = 1$. The $J(d)$ is an equation of the objective function with the following constraints in Eq. (6).

$$J(d) = \min \frac{1}{2} \sum_{m=1}^{M} \frac{1}{d_m} \|f_m\|^2 + C \sum_{i=1}^{n} \xi_i. \tag{6}$$

It is subject to $y_i \sum_{m=1}^{M} f_m(x_i) + y_i b \geq 1 - \xi_i$ and $\xi_i \geq 0$. Then, the dual problem is as follows.

$$J(d) = \max_{\alpha} -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \sum_{m}^{M} d_m K_m(x_i, x_j) + \sum_{i=1}^{n} \alpha_i, \tag{7}$$

with $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C, \forall i$.

Equation (7) is a Quadratic Programming (QP) problem with a linear constraint. Training SVM is the same as solving convex optimization problems. So, by referring to Eq. (7), $J(d)$ can also be written as Eq. (8) with α^* being the optimal solution during the training process of SVM.

$$J(d) = -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i^* \alpha_j^* y_i y_j \sum_{m}^{M} d_m K_m(x_i, x_j) + \sum_{i=1}^{n} \alpha_i^*. \tag{8}$$

D. Validating the Model

Next, the researchers validate the model by using K-Fold Cross Validation. It is a validation method by dividing the data into k-subsets. Then, it is repeated k-times for learning and testing. In each repetition, one subset is used as test data and the other subset as

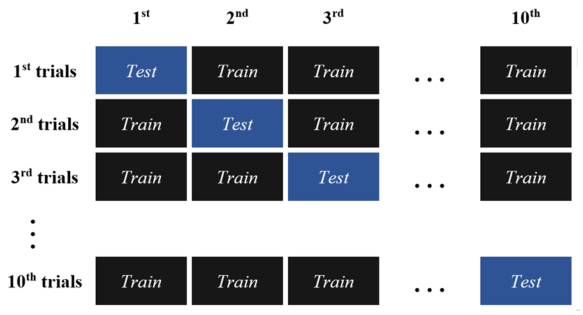

Fig. 2. Example of cross-validation scheme.

TABLE I
THE RESULT OF ALL TRIALS USING MULTI-KERNEL METHOD.

Trials	Sensitivity /Recall	Specificity	Precision	Accuracy	F1-Score
1	82.05%	84.72%	85.33%	83.33%	83.66%
2	80.77%	83.33%	84.00%	82.00%	82.35%
3	80.82%	79.22%	78.67%	80.00%	79.73%
4	85.07%	78.31%	76.00%	81.33%	80.28%
5	83.33%	**86.11%**	**86.67%**	84.67%	84.97%
6	78.95%	79.73%	80.00%	79.33%	79.47%
7	85.51%	80.25%	78.67%	82.67%	81.94%
8	85.33%	85.33%	85.33%	85.33%	85.33%
9	83.56%	81.82%	81.33%	82.67%	82.43%
10	**91.30%**	85.19%	**84.00%**	**88.00%**	**87.50%**
Avg	83.67%	82.40%	**82.00%**	82.93%	82.77%

learning data, obtained from a combination of different subsets [28]. In this section, the researchers divide the dataset into ten groups of dataset combinations. Then, the program runs repeatedly with different combinations of training and test data according to cross-validation rules. Figure 2 shows the explanation of the K-Fold Cross Validation approach used in the research.

E. Calculation Performance

The researchers calculate the system performance by calculating the value of accuracy, precision, specificity, sensitivity, and F1-score. This calculation starts with creating a confusion matrix containing the number of the classification result. First, True Positive (TP) is a group of positive data results that the system correctly detects. Second, False Positive (FP) is the result that is actually negative but is detected as positive by the system otherwise. Third, False Negative (FN) is a group of data results that are actually positive but are detected as negative by the system. Fourth, True Negative (TN) consists of data result that is actually negative and correctly detected as negative by the system. The value of accuracy, precision, specificity, sensitivity, and F1-score are obtained by operating addition, division, or multiplication of the results number of predicted and actual class as the following in Eq. (9) and confusion matrix (see Fig. 3) [29].

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}. \tag{9}$$

III. RESULTS AND DISCUSSION

The experiment is carried out using K-Fold Cross Validation. Each experiment is carried out ten times using a different combination of training and testing data, as mentioned previously. In Table I, the researchers present the results of trials using the proposed method to describe the research results comprehensively.

The result shows that the system can recognize the image of people with masks and no masks well. The average of the result is good, with 82.00% of precision. It means the program can recognize that people, which are labeled with masks, actually use a face mask. Then, the average sensitivity (aka recall) is 83.67% which means from all the samples of the face with a mask, 83.67% are correctly recognized. Then, the program can correctly predict 82.40% specificity of all face samples without a mask. Moreover, the average accuracy value shows that 82.93% of samples are correctly recognized. The average value of the F1-score (82.77%) represents the harmonic average of sensitivity and precision.

Table I also shows that the system runs well with the constant result of all the trials made. The lowest sensitivity value is 78.94% in the sixth trial, while the highest is in the tenth trial with 91.30%. In the other aspect, the lowest specificity value is in the fourth trial, with 78.31%, and the highest value is 86.11% in the fifth trial. Similarly, the precision gains the most value with 86.67% in the fifth trial, but the lowest value is made in the fourth trial with 76.00%. It can be said that the tenth trial is the best, as proven by the highest value of accuracy and F1-score of 88.00% and 87.50%, respectively. Contrastly, in the sixth trial, accuracy and F1-score only gain 79.33% and 79.47%, respectively.

The Multi-kernel algorithm is constructed by combining all three kernels of linear, polynomial, and RBF. Thus, the result will be provided by showing the experimental results. It uses the proposed method compared to the experiments with a single kernel with the same dataset and trial method. Table II presents the average of all comparisons.

Table II shows that the proposed method has the best result in all aspects. It has 83.67% of sensitivity, 82.40% of specificity, 82.00% of precision, 82.93% of accuracy, and 82.77% of F1-score. Meanwhile, the worst result is obtained by RBF Kernel with 53.03% of sensitivity, 51.28% of specificity, 31.47% of precision, 51.80% of accuracy, and 39.50% of F1-score. With that result, it can be concluded that, in this case of

Cite this article as: M. Athoillah, E. Purnaningrum, and R. K. Putri, “Modified Multi-Kernel Support Vector Machine for Mask Detection”, CommIT Journal 16(2), 159–166, 2022.

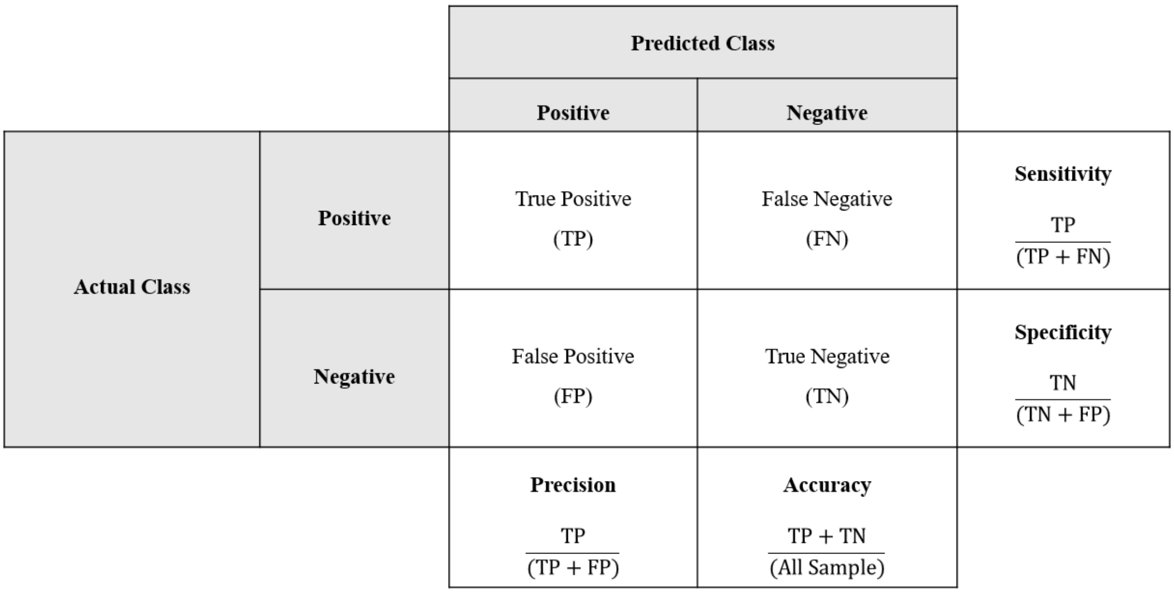

Fig. 3. Confusion matrix.

TABLE II
THE COMPARISON RESULT OF ALL EXPERIMENTS.

Aspect	Method (Average)		
	Multi Kernel	Polynomial Kernel	RBF Kernel
Sensitivity/Recall	**83.67%**	81.13%	53.03%
Specificity	**82.40%**	80.47%	51.28%
Precision	**82.00%**	80.27%	31.47%
Accuracy	**82.93%**	80.80%	51.80%
F1-Score	**82.77%**	80.70%	39.50%

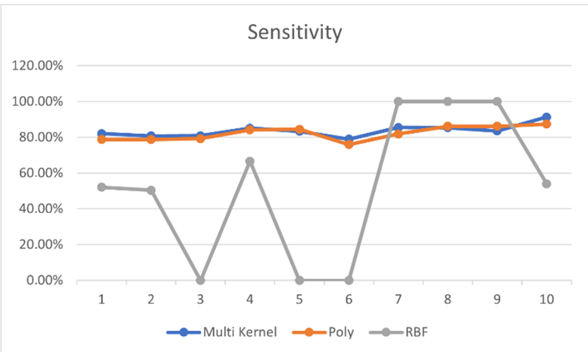

Fig. 4. Sensitivity/recall of all experiments.

recognizing a face with a mask or not, the proposed method (Multi-kernel SVM algorithm) can recognize better than other single kernel SVM. In addition, the results of a system with a single linear kernel SVM cannot be displayed because it has an error during the calculation. So, the system cannot reach convergence at the maximum number of iterations.

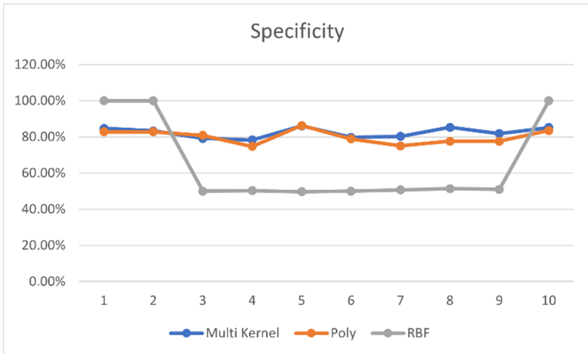

Fig. 5. Specificity of all experiments.

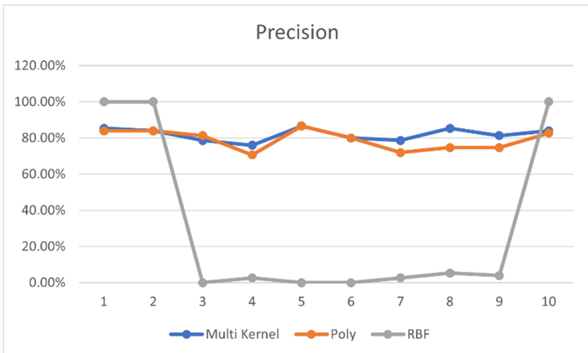

Fig. 6. Precision of all experiments.

Figures 4 to 6 show the results of each trial on each aspect of the assessment in the entire experiment to

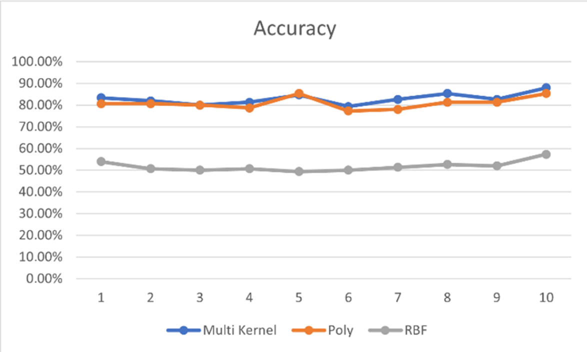

Fig. 7. Accuracy of all experiments.

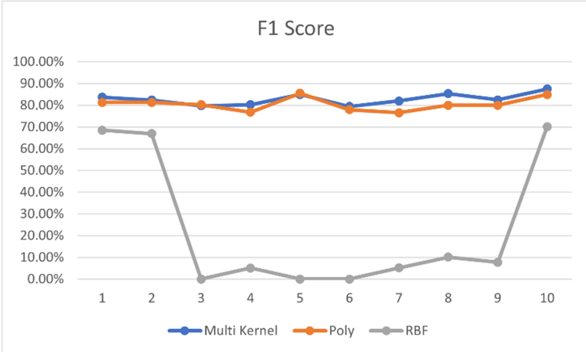

Fig. 8. F1-Score of all experiments.

understand better the differences in the results of mask identification using the Multi-kernel method with other single-kernel methods tested. Figures 4 to 6 show the performance graph of sensitivity/recall, specificity, and precision of all experiments. The value of the proposed method is good and constant from all experiments, around 80%. Moreover, it can also be seen that the sensitivity/recall, specificity, and precision value of Multi-kernel SVM are similar to the result of the SVM algorithm using Polynomial Kernel. Conversely, the SVM method using RBF kernel not only has a lower value of sensitivity/recall, specificity, and precision in general but is also very fluctuating, especially in sensitivity/recall value. It can be seen from several experiments that the value is 0%, yet some turn out to be 100%. It proves that the results of the SVM classification using the RBF kernel are strongly influenced by the characteristics of the data because of the experiments that have been carried out using the K-Fold Cross Validation approach. Each data group formed is different from one another, and it is very possible to obtain very different data characteristics.

Figures 7 and 8 show the performance graph of accuracy and F1-score of all experiments. The value of the Multi and Polynomial kernels in SVM are good and constant from all experiments, around 80% in contrast with the result of RBF kernel SVM. The graph shows that the accuracy value using RBF kernel SVM is the lowest value of all trials compared to others. Furthermore, the value of the F1-score using RBF kernel SVM is not only the lowest but is also fluctuating the same as the other performance values.

IV. Conclusion

The research creates a program that can detect a face with or without a mask using SVM modified with Multi-kernel. The approach is built by combining three kernel kinds: the third-order polynomial kernel function, the linear kernel function, and the RBF kernel function to be one single kernel equation. The researchers test the program with 1,500 images and use 90% of them for training and 10% for testing. This research focuses on analyzing the results of face image recognition with masks or without masks by using histogram values on these original images dataset without any pre-processing of noise cleaning or other pre-processing on the dataset. The model classifier is validated using the K-Fold Cross Validation method. The researchers arrange the dataset into ten groups of datasets. Subsequently, the researchers conduct the experiments repeatedly with different combinations of training and testing data.

The result shows that the performance of Multi-kernel SVM to detect face masks is good. It is proven by the average value consisting of sensitivity with 83.67%, specificity with 82.40%, precision with 82.00%, accuracy with 82.93%, and F1-score with 82.77%. In addition, the proposed method is compared with SVM using a single kernel. The result reveals that the proposed method has the best result in all aspects compared to a program with a single polynomial kernel or RBF kernel using the same procedure and dataset. Moreover, the program that uses a single linear kernel cannot be done with this process and dataset because it has an error when finding convergence of iterations. It certainly indicates the previous hypothesis that it is difficult to choose a fit kernel for every classification problem because every kernel has characteristics. Consequently, not every kernel will give maximum results. It can even be null.

Further research can be done using other kernel combination processes, such as non-linear combination and the data-dependent combination method which assigns a certain kernel weight to each data instance. Another suggestion that can be made in further research is to process the image dataset first, such as cleaning noise or by using other image transformation methods, in addition to the hisogram value so that other perspectives can be obtained from face mask identification analysis research.

References

[1] WHO, “WHO Coronavirus (COVID-19) dashboard,” 2022. [Online]. Available: https://covid19.who.int/table

[2] Kementerian Kesehatan Republik Indonesia, “Vaksinasi COVID-19 nasional,” 2022. [Online]. Available: https://vaksin.kemkes.go.id/#/vaccine

[3] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, “Rational use of face masks in the COVID-19 pandemic,” *The Lancet Respiratory Medicine*, vol. 8, no. 5, pp. 434–436, 2020.

[4] S. Singh, U. Ahuja, M. Kumar, K. Kumar, and M. Sachdeva, “Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment,” *Multimedia Tools and Applications*, vol. 80, no. 13, pp. 19 753–19 768, 2021.

[5] A. Kumar, A. Kaur, and M. Kumar, “Face detection techniques: A review,” *Artificial Intelligence Review*, vol. 52, no. 2, pp. 927–948, 2019.

[6] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, “SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2,” *Sustainable Cities and Society*, vol. 66, pp. 1–11, 2021.

[7] S. Sethi, M. Kathuria, and T. Kaushik, “Face mask detection using deep learning: An approach to reduce risk of coronavirus spread,” *Journal of biomedical informatics*, vol. 120, pp. 1–12, 2021.

[8] T. Meenpal, A. Balakrishnan, and A. Verma, “Facial mask detection using semantic segmentation,” in *2019 4th International Conference on Computing, Communications and Security (ICCS)*. Rome, Italy: IEEE, Oct. 10–12, 2019, pp. 1–5.

[9] X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, “Real-time face mask detection method based on YOLOv3,” *Electronics*, vol. 10, no. 7, pp. 1–17, 2021.

[10] S. Yadav, “Deep learning based safe social distancing and face mask detection in public areas for COVID-19 safety guidelines adherence,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 7, pp. 1368–1375, 2020.

[11] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, “Implementation of principal component analysis on masked and non-masked face recognition,” in *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*. Dhaka, Bangladesh: IEEE, May 3–5, 2019, pp. 1–5.

[12] G. Jignesh Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, “Face mask detection using transfer learning of inceptionv3,” in *International Conference on Big Data Analytics*. Sonepat, India: Springer, Dec. 15–18, 2020, pp. 81–90.

[13] M. F. Naufal, S. F. Kusuma, Z. A. Prayuska, A. A. Yoshua, Y. A. Lauwoto, N. S. Dinata, and D. Sugiarto, “Comparative analysis of image classification algorithms for face mask detection,” *Journal of Information Systems Engineering and Business Intelligence*, vol. 7, no. 1, pp. 56–66, 2021.

[14] 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, pp. 1–11, 2021.

[15] A. Oumina, N. El Makhfi, and M. Hamdi, “Control the covid-19 pandemic: Face mask detection using transfer learning,” in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. Kenitra, Morocco: IEEE, Dec. 2–3, 2020, pp. 1–5.

[16] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[17] M. A. Chandra and S. S. Bedi, “Survey on SVM and their application in image classification,” *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, 2021.

[18] H. F. Kareem, M. S. Al-Huseiny, F. Y. Mohsen, and K. Al-Yasriy, “Evaluation of SVM performance in the detection of lung cancer in marked CT scan dataset,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1731–1738, 2021.

[19] S. Saeed, J. Baber, M. Bakhtyar, I. Ullah, N. Sheikh, I. Dad, and A. A. Sanjrani, “Empirical evaluation of SVM for facial expression recognition,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, pp. 670–673, 2018.

[20] A. Ajam, M. Forghani, M. M. AlyanNezhadi, H. Qazanfari, and Z. Amiri, “Content-based image retrieval using color difference histogram in image textures,” in *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. Shahrood, Iran: IEEE, Dec. 18–19, 2019, pp. 1–6.

[21] Y. N. Nabuasa, “Pengolahan citra digital perbandingan metode histogram equalization dan spesification pada citra abu-abu,” *J-Icon: Jurnal Komputer dan Informatika*, vol. 7, no. 1, pp. 87–95, 2019.

[22] M. A. Khan, “Detection and classification of plant diseases using image processing and multiclass support vector machine,” *International Journal of Computer Trends and Technology*, vol. 68, no. 4, pp. 5–11, 2020.

[23] P. Nanglia, S. Kumar, A. N. Mahajan, P. Singh, and D. Rathee, “A hybrid algorithm for lung cancer classification using SVM and Neural Networks,” *ICT Express*, vol. 7, no. 3, pp. 335–341, 2021.

[24] E. Purnaningrum and M. Athoillah, “SVM approach for forecasting international tourism arrival in East Java,” *Journal of Physics: Conference Series*, vol. 1863, no. 1, pp. 1–6, 2021.

[25] A. Patle and D. S. Chouhan, “SVM kernel functions for classification,” in *2013 International Conference on Advances in Technology and Engineering (ICATE).* Mumbai, India: IEEE, Jan. 23–25, 2013, pp. 1–9.

[26] M. Gönen and E. Alpaydın, “Multiple kernel learning algorithms,” *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[27] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, “SimpleMKL,” *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[28] Y. Zhang and Y. Yang, “Cross-validation for selecting a model selection procedure,” *Journal of Econometrics*, vol. 187, no. 1, pp. 95–112, 2015.

[29] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, “Part 1: Simple definition and calculation of accuracy, sensitivity and specificity,” *Emergency*, vol. 3, no. 2, pp. 48–49, 2015.