International Journal of Innovative Computing, Information and Control
Volume **19**, Number **4**, August **2023**

ICIC International ©2023 ISSN 1349-4198
pp. 1087–1101

MASKED FACE RECOGNITION BY ZEROING THE MASKED REGION WITHOUT MODEL RETRAINING

Roberto Johan Salim[1] and Nico Surantha[1,2]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ roberto.salim; nico.surantha }@binus.ac.id

[2]Department of Electrical, Electronic and Communication Engineering
Faculty of Engineering
Tokyo City University
1-28-1 Tamazutsumi, Setagaya-ku, Tokyo 158-8557, Japan

Received November 2022; revised March 2023

Abstract. *With the recent global pandemic event, the requirement to use masks, especially in public spaces, has become a challenge to the existing face recognition system. To overcome this challenge, previous studies have performed transfer learning and finetuning of the existing model with masked face datasets. Others have performed a preprocessing by cropping the masked face and then fine-tuning the model with the newly cropped datasets. However, retraining with preprocessed or masked faces may be costly or even unavailable for some with limited resources. Furthermore, these methods of preprocessing are ill-advised to be used directly using models that are not retrained as was found in this study. Therefore, this study explores and presents a way of cropping which shows increases in performance without the requirement of any training to the existing face recognition model. This method managed to increase the performance of the existing model by up to 9.09% when presented with masked-face scenarios.*
Keywords: Masked face recognition, Face recognition, Without retraining, Data augmentation

1. **Introduction.** With the recent pandemic situation, many preventive measures are taken to mitigate infections of the COVID-19 virus. Extensive hygiene protocol, social distancing, and usage of face masks are some measures that are taken to anticipate the spread of the diseases. While the main method of transmission is by air, some studies have found that it is possible to be infected by surface contacts [1,2]. With this, some people even developed insecurity about touching things that are in public spaces [3]. Surveys before the pandemic show that fingerprint sensor is the most popular type of biometric system [4]. Many systems implement fingerprint sensors as a form of biometric security to access restricted areas within public spaces. Given the current situation and public behavior of touching things in public, this type of biometric needs to be replaced. A form of contactless biometric such as face recognition can replace these methods. However, face recognition is also affected by the measures that are taken during the pandemic situation. Wearing a mask blocks more than half of the facial features that may be needed for the recognition process. While taking the mask off is still an option, it is ill-advised as the main virus transmission is by air, presenting a higher chance of infection. Therefore, it is very important to explore solutions that allow for masked face recognition.

DOI: 10.24507/ijicic.19.04.1087

Masked Face Recognition (MFR) has become an active research topic since the pandemic hit. MFR is a kind of occlusion FR where the occluded area is predicted to be between the chin and the nose. Studies have been done to try to improve the existing solution by tuning or retraining the face recognition model with masked faces [5,6]. Given scenarios, and the knowledge of the occluded location, it allows the usage of certain methods such as cropping images. This method preprocesses the image to only leave the part which is not covered by the mask to use as the input image [6-8]. However, the previous study has always involved at least a form of tuning to the face recognition model. As the cost of training or finetuning a face recognition model can be expensive, the option of retraining may not be feasible for everyone. [5] mentioned the finetuning process requires around 42 hours to train each of their models using a GTX 1080, while models in [9] were trained around 30 hours while using a GTX Titan X.

With the trending of the topic, much research performed on finetuning or training new face recognition models that perform better on the masked face. However, this does not reflect well on the public availability of these models. Much of this research does not provide the tuned model for easily accessible by the public, as most studies do not provide any links or any means other than contacting the author to obtain the trained mode [5,6,10,11].

This study explores and proposes an image preprocessing method to improve the performance of pre-trained face recognition models without the need of retraining the model. This method of cropping preserves the position of the facial landmark location by zeroing the parts that should be covered by a mask. Besides recording the performance as the evaluation metrics, this experiment also records the preprocessing time of the experimented methods. This aspect was considered important as most of biometric devices placed in public spaces usually operate using portable devices. Thus, this experiment presents the following contributions.

1) Present a preprocessing method that improves the performance of the existing pre-trained model by cropping and maintaining facial landmark location.
2) Evaluate the improvement of performance using accuracy and F1 on the proposed preprocessing method given masked face recognition by evaluated on several classifiers given the classification cases.

This paper is structured as follows. Section 2 presents the related works on masked face recognition. Section 3 introduces the research stages and experimental design; here we detail the proposed cropping method as part of the data preprocessing. The result is then presented and discussed in Section 4. And a conclusion is drawn in Section 5.

2. **Related Works.** Given the situation between face recognition and pandemic measure presenting the raise of the masked face recognition problem, some have experimented with the effect of the existing solution when faced with masked face recognition. Saib and Pudaruth [7] asked the question of whether it is possible to do face recognition given a masked face. Within their study, they experiment on VGG16 and MobileNetV2 architecture with a softmax layer. This study also presents a scenario where the face image is cropped, leaving only the eyes and forehead region as their scenario. Their results present that the best model with 91.37% of accuracy was the MobileNetV2 architecture with a softmax layer where they trained the model again using a mixture of masked and unmasked images. While this paper does not mention the exact use of the datasets and only mentions the use of PINS Face Recognition dataset, it is assumed that the model was transfer-learned using this dataset and tested against a different split of these datasets; therefore, it is required to train using masked and not masked type of images every time a new identity introduced into the recognized list.

The idea of transfer learning using masked face images is also presented by Anwar and Raychowdhury [5] who present a tool called MaskTheFace to help create datasets to be used when training a model. The authors claim that this additional masked face in the training sets manages to increase the performance of the model by up to 34%. This type of model improvement was also experimented by Wirianto and Mauritsius [10] that performed transfer learning on a pre-trained ResNet100 model using an ArcFace loss function with a self-collected dataset called Indonesia Labelled Face in the Wild (ILFW). This study claims to obtain an accuracy of 92% during the training process.

Table 1. Related works

Works	Method	Dataset	FR model	Classifier	Results
[5]	Dataset Augmentation + Transfer Learning	LFW-Simulated	FaceNet	Verification	93.43%
[6]	Cropping + Transfer Learning	RMFR-CEN	VarGFaceNet ShuffleNet MobileFaceNet	Verification	77.2%
[7]	Cropping + Transfer Learning	PINS	MobileNetV2 VGG16 HOG	Softmax SVM	91.37%
[8]	Cropping + Train from Scratch	SMFRD Extended YALE B	Convolutional Attention Module	Verification	81.42%
[10]	Data Augmentation + Transfer Learning	ILFW (Indonesia LFW)	ResNet100 + ArcFace	–	92%

The other idea of improving existing models by cropping the input image as used by Saib and Pudaruth [7] was also evaluated by Martínez-Díaz et al. [6] and further studied by Li et al. [8]. By cropping the face and leaving parts not covered by the mask, make the recognition process similar between masked and not masked cases. With this method of preprocessing, Martínez-Díaz et al. [6] experiments manage to capture an accuracy of 77.2% when performing close identification using MobileFaceNet which has been finetuned using WebFace Dataset. Li et al. [8] further studied the cropping method and parameter with findings of the optimum cropping ratio between eye distance and cropping height. Within the study, they claim to obtain an accuracy of 81.42% given train on not masked and tested on the masked image.

This review concludes that most of the study that has been done on masked face recognition requires some degree of tuning. This can be in a form of training the model from scratch as well as transfer learning on a pre-trained model using simulated or preprocessed by cropping datasets. To the best of our knowledge, there has not been any study that presents the performance of the model without any form of tuning given the masked face scenario. Therefore, this study explores the effect of masked as well as experiments on the cropped method given a pre-trained model without tuning. This study also proposes and explores another cropping method to improve performance without the need for retraining.

3. **Research Methodology.** This section presents the research method that has been performed to obtain the experiment results. The process of this experiment is presented in Figure 1. The first step is data collection. This step shows which dataset was used in the experiment and its properties before being augmented and preprocessed for this experiment. Next is data augmentation and preprocessing. Here we present a detailed

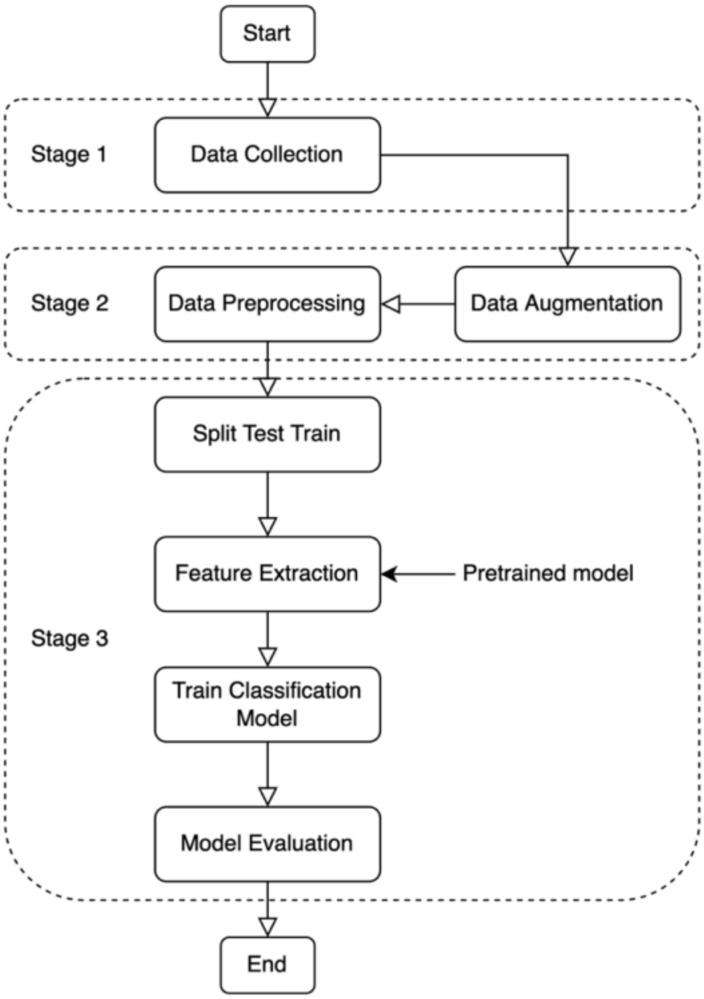

FIGURE 1. Research stages

process of augmenting normal face recognition datasets to fit our masked scenario, as well as the preprocessing method and our proposed method for improvement without retraining the face recognition model. Lastly, present the research methodology and the evaluation. This section shows the model and classification method used in the study, the experiment scenarios that represent the masked face recognition cases, as well as the evaluation method used in the study.

3.1. **Data collection.** For this experiment, four publicly available datasets were in the evaluation. These datasets are COMASK-20 [12], PINS [13], YALE [14], and MFR-2 [5]. COMASK-20 dataset provides some masked face images. These images can be a real masked face or a generated one. MFR-2 contains both real-life masked and not masked images without any generated masked face images. On the other hand, PINS and YALE do not contain any masked face image. Therefore, an augmentation process was done on these datasets creating a copy of the masked face image. This step will be discussed further in the next section. Table 2 shows the properties of each dataset collected.

3.2. **Data augmentation.** As mentioned before, some datasets do not contain any masked faces. Therefore, an augmented version of PINS and YALE was created by simulating face masks upon each of its images. To solve this issue, a masking tool by the name of MaskTheFace [5] was employed to perform mask augmentation on the dataset without a masked face. This tool utilizes the dlib face landmark to cover the mouth area

Table 2. Dataset properties

Dataset	Identity	Images
COMASK-20 [12]	312	2,824
MFR-2 [5]	53	269
PINS [13]	105	17,534
YALE [14]	15	165

Table 3. Dataset grouped by masked and not masked images

Dataset	Class	Not masked	Masked
COMASK-20	312	1,371	1,300
MFR-2	53	98	167
PINS (Augmented)	105	14,545	10,664
YALE (Augmented)	15	165	161

with an image of a mask. This tool generates masked images using random mask options excluding the gas masks that are available within the tools. This augmentation process separates the masked and not masked datasets. Therefore, a separation process of masked and not masked images in COMASK-20 and MFR-2 into separate folders has also been performed. This separation of masked and not masked will be used in the creation of test and train splitting for each scenario in the evaluation stage.

3.3. **Data preprocessing.** After the augmentation and the separation of the masked and not masked dataset, preprocessing steps were taken to find the face location, align, and crop based on the preprocessing method on the masked area. For this preprocessing, we employ an implementation of RetinaFace [15] model by [16]. Using the face area and facial landmark generated by the face detection model, we perform the alignment process using the eye locations to rotate each image accordingly and then isolate the face area to get the image for input without the preprocessing. For the cropping preprocess, using the y-axis of the nose landmark, we cropped the image height from the top of the face area to the y-axis of the nose. This cropping scenario represents the experiment that was done in [6,7].

In this study, we proposed another masked area cropping method that preserves the height of the face area. Rather than cropping the masked area and changing the image height, we keep the height of the face area but map every value below the y-axis of the nose to zero value. We call this method of cropping by zeroing. Figure 2 shows an example of every preprocessing method used in this experiment.

3.4. **Experimental design.** This section presents in detail the scenarios and parts on which the evaluation performed. This includes the pretrained model that was used in this study and where we obtain them, the classifier used in the identification process, and detailed scenarios that we created for this study.

3.4.1. *Face recognition models.* For the face recognition model, FaceNet [17], ArcFace [18], and MobileFaceNets [19] are chosen to generate the face embeddings for the classifiers. These models are chosen as they are generally publicly available. The FaceNet and ArcFace model that was used in this study is obtained from [20] which is a collection of models from [9] and [21] respectively, while the MobileFaceNets model was taken from [22]. These models will be used to perform feature extraction from the preprocessed image into a form of vector representation of the image called face embeddings.

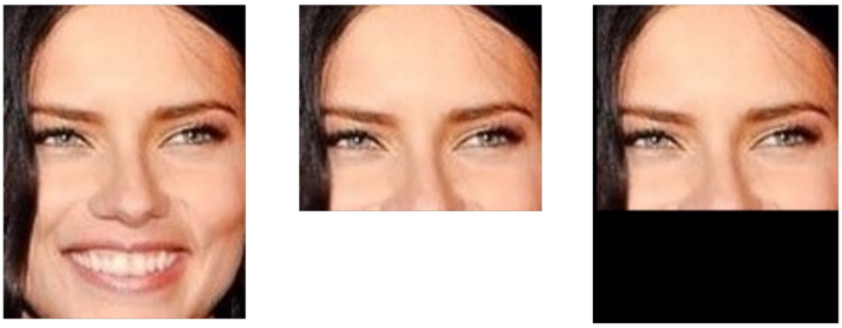

Figure 2. Dataset types after preprocessing

3.4.2. *Classifiers.* In this experiment, we want to test the performance of the pre-trained face recognition model given a masked face scenario without any process of retraining or fine-tuning of the face recognition model. To get the results, we evaluated the performance of classifiers with the input of face embeddings generated by the face recognition model. The classifiers we have chosen are KNN, SVM with One vs Rest strategy, and Mean Embeddings (ME) [23]. These classifiers are chosen as it is the common option and has relatively faster training time than training a classifier layer.

K-Nearest Neighbour (KNN) performs classification by selecting the K number of the closest distance between the candidate and trained data. The identification process then counts the most appearing label in the K closest to the predicted value.

Support Vector Machine (SVM) creates hyperplane boundaries between trained classes to perform prediction by deciding which side of the candidate falls relative to the trained boundaries. One vs Rest strategies make a boundary for each label with each data in a given label as positive data and every other labels as negative data. With this approach, the comparison should only be performed at most based on the number of labels which should be faster than the other strategy of One vs One.

Mean Embeddings (ME) [23] performed identification by performing distance comparison using the means of each trained label against the candidate. This algorithm performs distance comparison only to each means of labels. This method was implemented using the NearestCentroid module in scikit-learn.

3.4.3. *Scenarios.* The experiment was performed based on scenarios of masked face recognition. For this experiment, four scenarios representing different kinds of preprocessing were created. Within each scenario, first, we perform feature extraction using a face recognition model, next use the training split to train the classifier, and lastly perform testing by predicting the subject of each image in the testing set. These scenarios evaluate the prediction of the trained classifier using each combination of preprocessing based on the scenario. Each classifier in every scenario was trained using not masked data and tested accordingly using masked data except in the first scenario.

The four scenarios that were used in this experiment are detailed as follows. The first scenario acts as the baseline where the train and testing are images without masks. The purpose of this scenario is to benchmark the performance of the face recognition model given its intended purpose. This scenario also acts as a validation for the usability of the pre-trained models gathered for this study. The second scenario presents the testing data using a dataset containing both masked and not masked data. This scenario represents the reality of the existing system when faced with masked faces. Next, the third scenario experiment performed train and test given the dataset has been preprocessed, cropping the masked area. This type of preprocessing has been previously studied and has shown an increase in performance compared to the second scenario [6-8]. Lastly, the fourth scenario presents a similar condition to the third scenario but uses zeroing instead of cropping.

Table 4 shows that every scenario other than the first is tested using a masked dataset and the preprocessing method used in each scenario. Figures 3-6 show an example of images used in the training and testing process after being augmented and preprocessed. The test example shown in Figures 5 and 6 use the masked type of data before being preprocessed as can be seen by the little leftover of the mask part near the subject nose.

Table 4. Scenario properties

Scenario	Masked test data	Preprocessing (train & test data)
1	No	Without preprocessing
2	Yes	Without preprocessing [6,7]
3	Yes	Cropping [6,7]
4	Yes	Zeroing (proposed method)

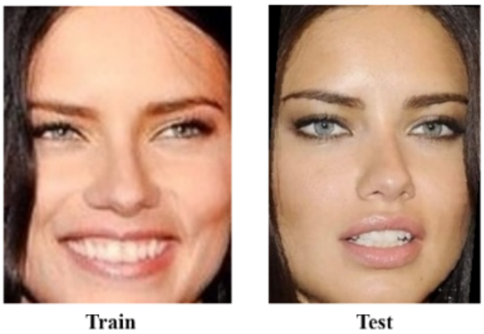

Figure 3. Scenario 1

Figure 4. Scenario 2

Figure 5. Scenario 3

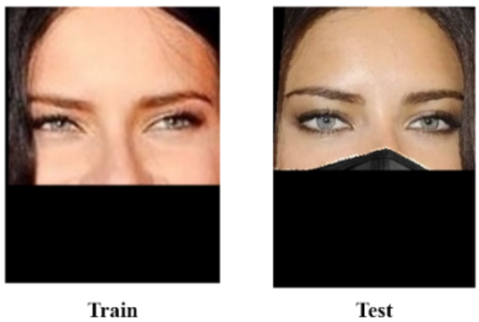

Figure 6. Scenario 4

3.5. **Evaluation.** As mentioned in the research method, the train will be used to train the classifier in identification cases. This experiment split the training and testing image based on the individuals, this ensures that each test image will have at least one of its images as part of the training dataset.

The splitting process was performed with a 0.33 ratio for testing on the not masked part of the dataset. Then for each individual in the test set, images from the masked set are randomly picked to match the number of test images for that subject. This ensures that the number of masked and not masked images in the test image will always be the same. This process often oversamples the masked images as it is likely that the masked image dataset contains fewer images than its not masked counterpart. This could be due to the limitations of the dataset or the inability to process certain images during the preprocessing step. Some labels in datasets COMASK-20 and MFR-2 may only contain one not masked image that will be attempted to be split. This resulted with an error in the splitting process that will cause the number of labels to be less than the original dataset. Table 5 shows the splitted dataset used in this experiment.

Table 5. Scenario properties

Dataset name	Label	Train (Not masked)	Test	
			Not masked	Masked
COMASK-20 [12]	293	897	454	454
MFR-2 [5]	45	45	45	45
PINS [13]	105	9,695	4,850	4,850
YALE [14]	15	105	60	60

The evaluation was performed using the train-test split shown in Table 5 that record evaluation matrix of accuracy and weighted F1 score. The record accuracy was calculated by dividing the correct prediction of the trained classifier by the total image in the test part of the dataset (1). The F1 score was calculated using the weight of each label due to the imbalance of the dataset (2). The weight of each class is calculated using the ratio of the portion of the class within all images in the test dataset (3).

$$Accuracy = \frac{Ture\ Positive}{Total\ Images} \times 100\% \tag{1}$$

$$Weight\ avg\ F1 = F1_{Class\ 1} \times W_1 + F1_{Class\ 2} \times W_2 + \cdots + F1_{Class\ n} \times W_n \tag{2}$$

$$W_n = \frac{Image\ Count\ in\ Class\ n}{Total\ Data} \tag{3}$$

This experiment also records the processing time as a general comparison between each method and combination. For this, the time recording was performed using the Python "time" built-in library. A timestamp is recorded before calling the predict function both in face recognition model inference and the classifier. Another timestamp is then called after each "predict" function is completed. Time difference between both timestamps is subtracted to get the total computational time given processing a dataset. This value is divided by the total of the predicted image for an average inference time for an image.

4. **Result and Discussion.** This section shows the results of the study which have been obtained. This part shows the results of the classification scenario obtained in this experiment.

4.1. **Classification scenarios.** The results of the experiment are shown in Table 6. The table shows the recorded accuracy of each face recognition model and classifier for every testing scenario. These scenarios have been previously presented in Section 3.4.3 which contains improvements attempt when facing masked faces in scenarios 3 and 4. Data in Table 6 shows comparable results when comparing results with the same dataset.

Excluding the first scenario that acts as a baseline, scenario 4 shows the best result compared to scenarios 2 and 3. Scenario 4 shows mostly positive improvement over the other scenario excluding the first. With the highest difference between scenarios 2 and 4 of 29.38% in accuracy when tested on COMASK-20 dataset with ArcFace as its face

Table 6. Classification results

Datasets	Face recognition model	Classifier	Accuracy (%)			
			Scenario 1	Scenario 2 [6,7]	Scenario 3 [6,7]	Scenario 4 (Proposed)
COMASK-20	MobileFaceNet	ME	95.81	54.84	74.75	78.17
		KNN	83.70	53.19	64.02	65.97
		SVM	96.47	64.31	77.31	78.04
	ArcFace	ME	97.35	54.40	57.31	83.78
		KNN	81.49	47.24	53.78	70.97
		SVM	98.01	74.33	69.63	**86.34**
	FaceNet	ME	92.51	52.20	66.82	76.34
		KNN	74.22	42.95	55.60	60.97
		SVM	93.39	65.41	73.41	76.46
PINS	MobileFaceNet	ME	90.97	76.65	47.32	81.41
		KNN	86.96	64.03	42.03	73.50
		SVM	94.81	85.44	65.75	87.97
	ArcFace	ME	88.44	70.58	28.10	81.80
		KNN	87.73	65.71	46.57	76.24
		SVM	95.82	88.34	70.17	**89.98**
	FaceNet	ME	91.37	81.26	69.42	82.14
		KNN	90.39	72.25	66.42	80.61
		SVM	93.83	85.98	79.22	88.06
YALE	MobileFaceNet	ME	93.33	79.16	94.16	95.83
		KNN	93.33	80.83	90.00	95.83
		SVM	98.33	92.50	96.66	97.50
	ArcFace	ME	100.00	81.66	51.66	97.50
		KNN	100.00	74.16	61.66	94.16
		SVM	100.00	90.00	82.50	**98.33**
	FaceNet	ME	98.33	85.83	89.16	93.33
		KNN	98.33	84.16	84.16	90.83
		SVM	98.33	93.33	90.83	97.50
MFR-2	MobileFaceNet	ME	93.33	56.66	30.00	63.33
		KNN	22.22	21.11	5.55	15.55
		SVM	91.11	74.44	45.55	**80.00**
	ArcFace	ME	91.00	51.00	31.11	58.88
		KNN	15.00	12.00	10.00	10.00
		SVM	93.33	65.55	38.88	71.11
	FaceNet	ME	95.55	68.88	60.00	60.00
		KNN	15.55	13.33	11.11	12.22
		SVM	95.55	70.00	65.55	71.11

recognition model and Means Embedding as its classifier. This shows that the proposed method of zeroing does impact the performance with positive results.

Overall the highest accuracy recorded is obtained using SVM as its classifier. The face recognition model that shows the highest performing results is ArcFace, with the exception of the MFR-2 test where the MobileFaceNet performs better.

Figure 7 shows the average of the accuracy recorded from Table 7 into a box plot grouped by each scenario, while Figure 8 shows the F1 score of the recorded weighted F1 score. The first scenario presents a baseline performance of each model given their

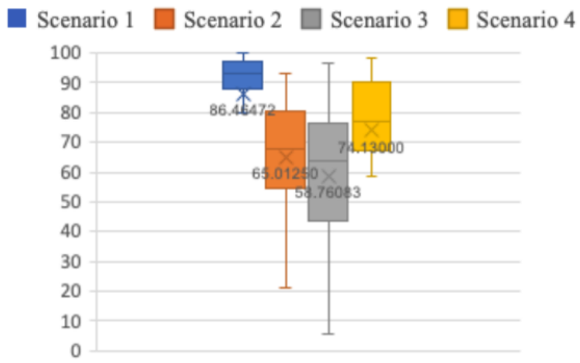

FIGURE 7. (color online) Accuracy over scenarios

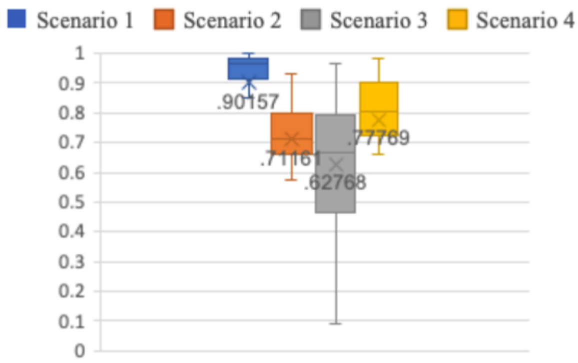

FIGURE 8. (color online) F1 score over scenarios

TABLE 7. Average accuracy for every scenario

Metric	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Accuracy	86.46%	65.01%	58.76%	74.13%
F1 score	0.90	0.71	0.63	0.78

intended usage, to generate face embeddings on images without any face mask. The evaluation of the first scenario is expected to perform well and outperforms every other scenario tested. This expectation is reflected in the recorded result in each figure between 7 to 12 with the blue color box plot with an average accuracy of 86.46% and an average weighted F1 score of 0.90.

The second scenario provides an example of the existing solution without considering masked faces when presented with masked face scenarios. Performance within this scenario is expected to decline compared to the first scenario. The evaluation in this study records an average accuracy of 65.01% which is a 21.45% decrease. This result is expected as has been found before by the previous study [5-8].

The third and the fourth scenarios performed the same scenario as the second scenario but present a preprocessing method on top of the dataset before performing face embedding extraction. In the third scenario, the cropping method [6,7] was used as an attempt to improve the performance. The results of this scenario show interesting results which shows that this method of improvement shows performance decline when used on not retrained models, with an average accuracy of 58.76% which is lower than the second scenario. However, with the high variance of the result shown by the box plot in Figures 7 and 8, this method might show improvement in some cases depending on the datasets.

Lastly, the fourth scenario was testing the proposed preprocessing method of zeroing. Evaluation of this scenario was evaluated exactly like the third but with a different type of preprocessing. The result in this scenario shows a positive improvement over the second scenario with an average accuracy of 74.13%. While this result is still lower than the baseline, this method still shows improvement over the existing solution without any kind of finetuning on the feature extraction model.

Further results are grouped scenarios for each of their recognition models and classifiers. Figures 9 and 10 show the experiment results grouped by each tested face recognition model. Results of the average show that the tested pre-trained model ArcFace performed better, with an average accuracy of 75.47% and an F1 score of 0.79. However, this performance difference is very closely followed by the other two models.

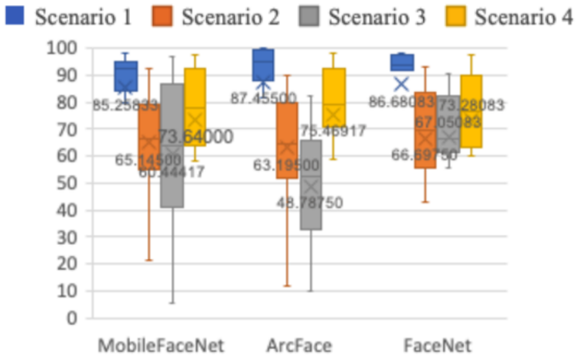

FIGURE 9. (color online) Accuracy over models

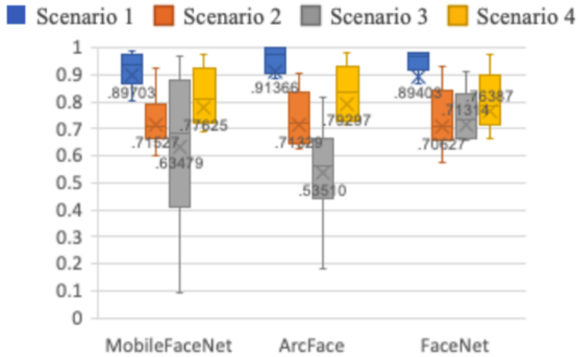

FIGURE 10. (color online) F1 score over models

TABLE 8. Average accuracy (%) over face recognition models

Models	Scenario 1	Scenario 2	Scenario 3	Scenario 4
MobileFaceNet	85.26	65.15	60.44	73.64
ArcFace	87.46	63.20	48.79	75.47
FaceNet	86.68	66.70	67.05	73.28

TABLE 9. Average F1 score over face recognition models

Models	Scenario 1	Scenario 2	Scenario 3	Scenario 4
MobileFaceNet	0.90	0.72	0.63	0.78
ArcFace	0.91	0.71	0.54	0.79
FaceNet	0.89	0.71	0.71	0.76

Figures 11 and 12 present the average of results grouped by the tested classifier. The result in this group also shows that the zeroing method that is tested in the fourth scenario managed to perform better than the second and third scenarios. Based on the evaluated result, SVM shows the best performance compared to the other tested classifier method. SVM manages to score an accuracy of 84.44% and an F1 of 0.86. These results compared to the other tested method are significantly better with more than a 7% difference in accuracy when compared with Mean Embeddings. KNN shows very poor performance with very low accuracy and high variance.

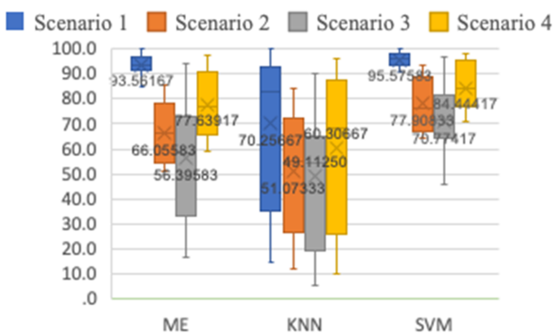

FIGURE 11. (color online) Accuracy over classifiers

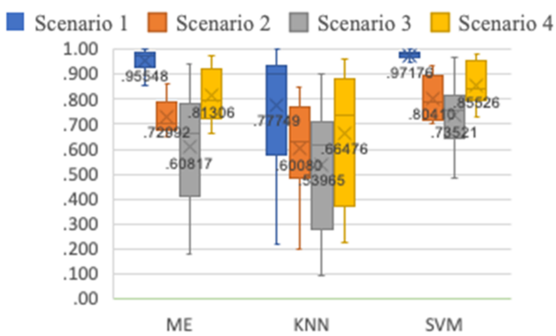

FIGURE 12. (color online) F1 score over classifiers

TABLE 10. Average accuracy (%) over classifiers

Classifier	Scenario 1	Scenario 2	Scenario 3	Scenario 4
ME	93.56	66.06	56.40	77.64
KNN	70.26	51.07	49.11	60.31
SVM	95.58	77.91	70.77	84.44

TABLE 11. Average F1 score over classifiers

Classifier	Scenario 1	Scenario 2	Scenario 3	Scenario 4
ME	0.96	0.73	0.61	0.81
KNN	0.78	0.60	0.54	0.66
SVM	0.97	0.80	0.74	0.86

4.2. **Processing time.** This study also records the processing time of each classifier when combined with different combination of pretrained face recognition models. This experiment was run on a laptop using with Ubuntu 20.04 using Intel i5-8300H CPU. Table 12 shows the average processing time captured during model feature extraction, as well as the classifier training and predicting time. This data was captured when evaluating the scenario using PINS dataset. Feature extraction was time recorded when the image is being predicted by the face recognition model. Classifier training time is the total time to train the classifier using 8,326 training images, and the prediction is the average time taken to predict a single feature by the classifier.

TABLE 12. Processing time results

Pretrained model	Time (ms) Feature extraction	Classifier	Time (ms) Predicting
MobileFaceNet	33.00	ME	0.0045
		KNN	1.2769
		SVM	2.2585
ArcFace	77.00	ME	0.0080
		KNN	4.2917
		SVM	7.1692
FaceNet	34.57	ME	0.0051
		KNN	1.1672
		SVM	1.4890

The results of the feature extraction process are as expected. MobileFaceNet is expected be the fastest model while ArcFace being the most complex model present in this experiment. The results of the classifier show that Mean Embeddings (ME) is the fastest classifier method at predicting the class of the face embeddings evaluated in this study, followed by KNN, and lastly the SVM.

4.3. **Discussion of results.** Based on the results of this study, we concluded that our proposed method of zeroing has managed to improve the accuracy of the pre-trained model without any form of retraining. While the performance is still below the baseline and does not out-perform models which have been retrained with augmented data of masked faces. It still manages to improve the accuracy when compared to masked face recognition without any preprocessing or using cropping which changes the height of the input image. When compared with previous works, the proposed method provides

Table 13. Comparison result with previous works

Dataset	Method	Accuracy
PINS	Cropping + MobileNetV2 (retrain) [7]	91.37%
	Cropping + VGG16 + SVM [7]	81.52%
	Zeroing + ArcFace + SVM **(Proposed)**	89.98%
YALE	Cropping + Train from Scratch [8]	81.42%
	Zeroing + ArcFace + SVM **(Proposed)**	98.33%

comparable performance compared to some of the previous results. This comparison is shown in Table 13.

The results of this experiment also show that the cropping method used in the previous study [7,8] cannot be used directly on top of the existing pre-trained model. A retraining process using a preprocessed dataset is required in order to improve the performance of the recognition process. Without this form of retraining, the performance is shown to have a very high variance.

Another result that can be obtained from this experiment is the recommended combination to be used according to the evaluated performance of combinations of pre-trained face recognition model and classification method. The best combination according to performance results found in this study is combining a pre-trained ArcFace model and SVM classifier with the One vs Rest strategy while using zeroing as the pre-processing method when considering masked face recognition without retraining the pre-trained model. However, when considering usage on portable devices, another model for face feature extraction is preferred as ArcFace results having more than twice the time required to process a single image while performing similarly to the other tested models. As for the classifier, the performance of SVM shows a significant performance difference from the rest and should still be preferred for usage on portable devices.

5. **Conclusion and Future Works.** With the recent global pandemic event, Masked Face Recognition (MFR) has become an active research topic. With the challenge presented by the use of masks during the pandemic, many studies have trained and evaluated face recognition models using masked face datasets to improve their performance. However, this method of improving the performance of the face recognition model might not be available for everyone given the resource needed to train or finetune a face recognition model is not small. Furthermore, the availability of these trained or tuned model is not easily available to the public. Thus, this study explores the performance of publicly available resources such as face recognition models and methods of preprocessing to increase the performance of these models against masked faces without the need for retraining or tuning.

This study found that removing the masked area but leaving the relative location of the facial landmark intact by zeroing the masked face rather than cropping it, can consistently improve the performance compared to without any preprocessing of the input image. The result of this study shows consistent improvement when comparing the identification results given masked faces as a test scenario. While the results might not out-perform models which have been retrained, it manages to improve the accuracy of these tested models given masked face scenarios. Future works can explore the optimum cropping method for this type of preprocessing. While implementation of this method can also be explored for usability in portable devices.

Acknowledgment. This work was supported by Ministry of Education, Culture, Research and Technology, Republic of Indonesia.

REFERENCES

[1] S. Riddell, S. Goldie, A. Hill, D. Eagles and T. W. Drew, The effect of temperature on persistence of SARS-CoV-2 on common surfaces, *Virol. J.*, vol.17, no.1, DOI: 10.1186/s12985-020-01418-7, 2020.

[2] J. Biryukov et al., Increasing temperature and relative humidity accelerates inactivation of SARS-CoV-2 on surfaces, *mSphere*, vol.5, no.4, DOI: 10.1128/msphere.00441-20, 2020.

[3] Asukanet Co., Ltd., *Awareness Survey on 'Contact with Things before and after the Corona Shock'*, https://www.asukanet.co.jp/contents/news/2020/20200525.html, Accessed on Jan. 27, 2022.

[4] R. German and K. S. Barber, *Current Biometric Adoption and Trends*, https://www.abiresearch.com/, 2017.

[5] A. Anwar and A. Raychowdhury, Masked face recognition for secure authentication, *arXiv Preprint*, arXiv: 2008.11104, 2020.

[6] Y. Martínez-Díaz, H. Méndez-Vázquez, L. S. Luevano, M. Nicolás-Díaz, L. Chang and M. González-Mendoza, Towards accurate and lightweight masked face recognition: An experimental evaluation, *IEEE Access*, vol.10, pp.7341-7353, DOI: 10.1109/ACCESS.2021.3135255, 2022.

[7] Y. M. Saib and S. Pudaruth, Is face recognition with masks possible?, *International Journal of Advanced Computer Science and Applications*, vol.12, no.7, 2021.

[8] Y. Li, K. Guo, Y. Lu and L. Liu, Cropping and attention based approach for masked face recognition, *Applied Intelligence*, vol.51, no.5, pp.3012-3025, DOI: 10.1007/s10489-020-02100-9, 2021.

[9] D. Sandbergs, *Training Using the VGGFace2 Dataset*, https://github.com/davidsandberg/facenet/wiki/Training-using-the-VGGFace2-dataset, Accessed on Nov. 03, 2022.

[10] Wirianto and T. Mauritsius, The development of face recognition model in Indonesia pandemic context based on DCNN and Arcface loss function, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1513-1530, 2021.

[11] D. Montero, M. Nieto, P. Leskovsky and N. Aginako, Boosting masked face recognition with multi-task ArcFace, *arXiv Preprint*, arXiv: 2104.09874, 2021.

[12] H. N. Vu, M. H. Nguyen and C. Pham, Masked face recognition with convolutional neural networks and local binary patterns, *Applied Intelligence*, DOI: 10.1007/s10489-021-02728-1, 2021.

[13] Burak, *Pins Face Recognition*, https://www.kaggle.com/datasets/hereisburak/pins-face-recognition, Accessed on Oct. 01, 2022.

[14] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol.19, no.7, pp.711-720, DOI: 10.1109/34.598228, 1997.

[15] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia and S. Zafeiriou, RetinaFace: Single-stage dense face localisation in the wild, *arXiv Preprint*, arXiv: 1905.00641, 2019.

[16] S. I. Serengil and A. Ozpinar, HyperExtended LightFace: A facial attribute analysis framework, *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pp.1-4, DOI: 10.1109/ICEET53442.2021.9659697, 2021.

[17] F. Schroff, D. Kalenichenko and J. Philbin, FaceNet: A unified embedding for face recognition and clustering, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp.815-823, DOI: 10.1109/CVPR.2015.7298682, 2015.

[18] J. Deng, J. Guo, N. Xue and S. Zafeiriou, ArcFace: Additive angular margin loss for deep face recognition, *arXiv Preprint*, arXiv: 1801.07698, 2018.

[19] S. Chen, Y. Liu, X. Gao and Z. Han, MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices, *arXiv Preprint*, arXiv: 1804.07573, 2018.

[20] S. I. Serengil and A. Ozpinar, LightFace: A hybrid deep face recognition framework, *Proceedings – 2020 Innovations in Intelligent Systems and Applications Conference (ASYU2020)*, Institute of Electrical and Electronics Engineers Inc., DOI: 10.1109/ASYU50717.2020.9259802, 2020.

[21] Leondgarse, Keras insightface, *GitHub Repository*, GitHub, DOI: 10.5281/zenodo.6506949, 2022.

[22] Zye, *Face Recognition with Coral EdgeTPU Support Based on MobileFaceNet*, https://github.com/zye1996/Mobilefacenet-TF2-coral_tpu, Accessed on Feb. 01, 2022.

[23] M. I. Hossain, Sama-E-Shan and H. Kabir, An efficient way to recognize faces using mean embeddings, *Proceedings of the 2021 1st International Conference on Advances in Electrical, Computing, Communications and Sustainable Technologies (ICAECT2021)*, Institute of Electrical and Electronics Engineers Inc., DOI: 10.1109/ICAECT49130.2021.9392401, 2021.

Author Biography

Roberto Johan Salim is currently pursuing Bachelor and Master degrees in Information Technology with Master of Computer Science, BINUS Graduate Program, Bina Nusantara University, Indonesia. His research interests include computer vision, the Internet of Things, and machine learning.

Nico Surantha (Member, IEEE) received the B.Eng. and M.Eng. degrees from the Institut Teknologi Bandung, Indonesia, in 2007 and 2009, respectively, and the Ph.D. degree from the Kyushu Institute of Technology, Japan, in 2013. He is a full-time Lecturer with the Department of Electrical, Electronic and Communication Engineering, Tokyo City University, Japan. He is also a Lecturer with the Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Indonesia. His research interests include ubiquitous computing, intelligent systems, the Internet of Things, health monitoring, and system on chip design.