ICIC Express Letters
Part B: Applications
Volume **15**, Number **1**, January **2024**
 ISSN 2185-2766
pp. 11–17

MASKED FACE RECOGNITION FOR A LIGHTWEIGHT SCHOOL ATTENDANCE MANAGEMENT SYSTEM DURING COVID-19 PANDEMIC

Boy Sugijakko[1], 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
{ boy.sugijakko; 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 April 2023; accepted July 2023

Abstract. *This paper evaluates the effectiveness of masked-face recognition methods for attendance systems in the context of the COVID-19 pandemic. Hybrid learning has emerged as a practical option for limiting the spread of the virus, and contactless attendance systems are necessary to reduce transmission risk. However, face recognition systems may be less accurate due to the partial coverage of face masks. Furthermore, these systems need to be lightweight and computationally efficient to be practical in large-scale institutions or schools. This study evaluates four pretrained face feature extractor models (FaceNet, ArcFace, VGGFace, and MobileFaceNet) with an SVM classifier in several scenarios to determine the most suitable face recognition model. The results indicate that FaceNet and ArcFace have comparable accuracy, with minimal performance decrease in the presence of face masks. The second scenario yields the least reduced performance, and the FaceNet model performs twice as fast as ArcFace on a Raspberry Pi. These findings provide valuable insights for developing effective attendance systems for hybrid learning during the COVID-19 pandemic.*
Keywords: Face recognition, Masked face recognition, Attendance system, Lightweight system, Raspberry Pi

1. **Introduction.** Attendance is a crucial factor in maintaining academic performance, and chronic absence has been linked to low academic achievement and a higher likelihood of dropouts [1]. The pandemic has exacerbated chronic absence rates, especially with the remote learning model, although there has been a shift towards in-person learning [2], and more schools are expected to reopen [3]. Manual attendance systems have been deemed inefficient due to the potential for errors caused by proxies and impersonation [4], which can negatively impact the learning process. Therefore, an attendance management system that allows for efficient and accurate attendance-taking without physical contact is necessary. Examples of attendance management systems proposed in the past include Kovelan et al.'s RFID-based system [5], Swain et al.'s fingerprint-based system [6], and Vinod et al.'s system that combines RFID and fingerprint scans on a Raspberry Pi [7]. However, these systems do not comply with current health protocols, so an alternative method for attendance taking is needed.

DOI: 10.24507/icicelb.15.01.11

Face recognitions are considered the most suitable method for attendance taking as they do not require physical contact, reducing the risk of transmission and minimizing errors caused by proxies and impersonation. Bah and Ming [8] presented a face recognition-based attendance system that uses the improved Local Binary Pattern Histogram (LBPH) method, Bhattacharya et al. [4] used a Convolutional Neural Network (CNN), and Arsenovic et al. [9] utilized the FaceNet CNN model. To quickly take attendance while lowering the danger of COVID-19 transmission, a face recognition technology that can accurately recognize people while wearing masks is required.

Although general occlusions such as sunglasses and partial captures have been studied in the past, the effect of facial occlusion generated by a face mask is critical considering the most recent COVID-19 outbreaks [9]. A study by Saib and Pudaruth [10] demonstrated the possibility of having a high degree of accuracy while recognizing faces using a face mask. To improve accuracy, Anwar and Raychowdhury [11] proposed retraining models with masked face images. Despite these advancements, an important issue has been overlooked in these studies: the effectiveness of the suggested methods on edge devices.

To address this gap, this study utilizes a Raspberry Pi 4 device to ensure efficient processing. We evaluate the performance of various existing face recognition models across different masked scenarios, incorporating both masked and unmasked train and test datasets. Additionally, we include a baseline scenario for performance comparison. By doing so, we aim to provide valuable insights into the following contributions:

1) Present the best scenario that an existing pretrained model performs in;
2) Present a model that is less affected by masked face recognition while performing well on a lightweight device.

This paper is structured by presenting the related study which contains the original works in which the model was proposed in Section 2, followed by detailed experiments design and implementation in Section 3. The final part of the paper would cover results and discussion in Section 4 before presenting a brief conclusion of the study in Section 5.

2. **Related Works.** The models utilized in the experiment, including FaceNet, VGG-Face, ArcFace, and MobileFaceNet, are introduced in this section. These models were selected due to their widespread use and applicability as well as the accessibility of numerous online resources. Face recognition-based attendance systems were described in earlier research by Bah and Ming [8], Bhattacharya et al. [4], and Arsenovic et al. [9] utilizing a variety of models.

ArcFace, developed by Deng et al. [12], is a face recognition model that focuses on generating highly distinct features for face identification. FaceNet, proposed by Schroff et al. [13] in 2015, utilized the triplet loss function and enabled distance comparison between faces. It has been widely used for face recognition tasks. MobileFaceNet, introduced by Chen et al. [14], is an efficient variant of FaceNet that utilizes the MobileNetV2 architecture as its backbone, aiming to reduce computational costs. VGGFace, created by Parkhi et al. [15], is a deep face recognition model that leverages a large dataset of 2.6 million pictures and 2,622 identities to generate face descriptors. This model has been influential in the field of face recognition.

Several previous studies have explored the potential to enhance the performance of existing face recognition methods in recognizing masked faces [16]. For example, in a study by Mundial et al. [17], an accuracy of 97% was achieved on a Real-world Masked Face Dataset using a combination of CNN and SVM techniques. Another study by Negi et al. [18] demonstrated accuracies of 97.42% and 98.97% by employing CNN models built from scratch and VGG16 model with transfer learning, respectively, on a Simulated Masked Face Dataset. In line with these efforts, the present study aims to evaluate the performance of existing face recognition models on a standardized mobile device, such as

Raspberry Pi, when confronted with masked faces. The performance comparison will be conducted in terms of accuracy and computational time for each model.

3. **Method.** The aim of this research is to determine the most accurate and time-efficient model for face recognition on masked faces. The experiment consists of five stages: dataset collection, dataset processing, face embedding extraction, classifier training, and evaluation. The experiment is performed on both a personal computer and a Raspberry Pi 4.

In the first stage, two datasets with distinct characteristics are collected to represent different conditions. The second stage involves using a tool named MaskTheFace by Anwar and Raychowdhury [11] to mask faces in images and generate a masked face dataset, which is then split for training and testing. In the third stage, face embeddings are extracted from the images using several pre-trained models. In the fourth stage, classifiers for each model are trained using support vector machine and the face embeddings obtained in the previous stage. Finally, in the fifth stage, the trained classifiers are tested and evaluated.

3.1. **Dataset collection.** In this experiment, two datasets are used: the Yale Face dataset and the PINS face recognition dataset. The Yale Face dataset is used to represent a constrained environment for face recognition, while the PINS dataset represents a more complex and unconstrained environment. Both datasets are then transformed using MaskTheFace, a tool developed by Anwar and Raychowdhury [11], to create new masked datasets that will serve as evaluation instruments.

3.2. **Dataset augmentation and split test train.** The unmasked dataset, the cropped dataset, and the dataset with the area below the nose removed were all created for the study. Then, the masked dataset was produced by applying masks to the unmasked dataset using the MaskTheFace tool [11]. The masks were randomly selected from four different types, namely surgical, N95, KN95, and cloth. Samples of the datasets are shown in Figure 1, while the distribution of the datasets is presented in Table 1.

When constructing dataset for each generated dataset, some data are discarded when face landmark for cropping or MaskTheFace tools fails to output the desired results.

Figure 1. Sample images from non-masked, masked, and cropped PINS dataset and Yale Face dataset

Table 1. Dataset train and test distribution

Dataset	Subject	Generated dataset	Train image	Test image
PINS	105	Non-masked dataset	8838	2070
		Masked dataset	7297	1744
		Cropped dataset	6577	1594
Yale Face	15	Non-masked dataset	121	45
		Masked dataset	105	45
		Cropped dataset	113	42

To evaluate the performance of face recognition models on real-life application, several scenarios were constructed with these generated datasets:

1) Trained using non-masked dataset, tested using non-masked dataset;
2) Trained using masked dataset, tested using masked dataset;
3) Trained using non-masked dataset, tested using masked dataset;
4) Trained using cropped dataset, tested using cropped dataset.

Scenario 1 serves as a baseline for comparison because it is the scenario that represents the optimal use case of face recognition models. Scenario 2 represents a situation where a face recognition system is trained using images of masked faces to recognize images masked faces, while scenario 3 represents a situation where an existing face recognition system, which was trained using images of un-masked faces, is presented with masked faces to recognize. Scenario 4 aims to ignore the mask and only use upper part of the face which is not occluded to do face recognition.

3.3. **Feature extraction.** The DeepFace Python library [19] provides a range of face recognition models that are publicly available. In a prior study by Serengil and Ozpinar [19], FaceNet, ArcFace, and VGGFace achieved the highest accuracy on the LFW dataset. Thus, these models were selected for evaluation in our research. We also included MobileFaceNet as it is known to be a more lightweight version of FaceNet. We implemented the FaceNet, ArcFace, and VGGFace models using the DeepFace library [19], while the MobileFaceNet model was obtained from a repository by Zye [20], which is a Tensorflow 2 implementation of the model proposed by Chen et al. [14].

We obtained face embeddings for each of the models using the represent function from the DeepFace library [19]. This function resizes the input images according to the input size of each model, normalizes them, and uses the model to predict face embeddings. We followed the same process to obtain face embeddings for MobileFaceNet.

3.4. **Training classification model.** In this study, the SVM classifier was implemented using the scikit-learn library. The input face embeddings were normalized using the *sklearn.preprocessing.normalize* function. The normalized embeddings were then used to train the classifier model, created with *sklearn.svm.SVC* and a linear kernel.

3.5. **Model evaluation.** A Raspberry Pi 4 and a PC (Intel i5-8300H CPU, Nvidia GTX 1050 Ti mobile GPU) were utilized in the experiment. The average computation time and accuracy percentage were evaluated. Correctly identified faces were used to measure accuracy, while total computation time per face/image was used to measure average computation time. While average computation times varied between scenarios, accuracy calculations were equivalent across platforms.

4. Result and Discussion.

4.1. **Accuracy.** The Yale Face dataset was used as a baseline for comparison due to its controlled lighting and pose conditions. The accuracy achieved on this dataset was higher than that of the PINS dataset. The controlled environment and smaller size of the Yale Face dataset likely contributed to better model generalization. This highlights the importance of dataset characteristics and size, as they strongly influence the performance of face recognition models.

For scenario 2, the classifier performed somewhat worse than in the other scenarios after being trained and tested on masked datasets. With only a 3% performance difference, ArcFace and FaceNet were the best-performing models in this scenario. For scenario 3, it showed a significant decline in performance when the existing models were presented with masked face images. The performance decrease in accuracy was as much as 31% from the baseline. This indicates that some models are not directly usable in different scenarios. However, both FaceNet and ArcFace showed less impact while still experiencing

Table 2. Accuracy results

Dataset	Model	Accuracy (%)			
		Scenario 1	**Scenario 2**	**Scenario 3**	**Scenario 4**
PINS	ArcFace	98.0193	**93.176**	79.873	33.3816
	FaceNet	**98.3091**	90.940	**80.905**	**84.7342**
	MobileFaceNet	93.0917	83.371	65.194	17.8743
	VGGFace	92.0772	79.587	61.697	72.5120
Yale Face	ArcFace	100	100	97.7	60
	FaceNet	**100**	**100**	**100**	**97.7**
	MobileFaceNet	100	97.7	91.1	53.3
	VGGFace	100	97.7	95.5	97.7

a performance decrease. These two models are shown to be flexible enough to be used as is in the current condition of common masked face subjects.

Results from scenario 4 showed that mong the pretrained models, FaceNet and VGGFace maintained their usable accuracy, while ArcFace and MobileFaceNet output unusable performance for face recognition. Upon further research on the implementation, it was found that the different datasets on which the models were trained might be the reason for this performance difference. FaceNet and VGGFace were trained using VGGFace and VGGFace2 [21] datasets, which are very large academic datasets containing 2,622 identities with 2.6M images and 9,131 identities with 3.31M images [19], respectively. In contrast, ArcFace and MobileFaceNet were trained using CASIA WebFace datasets, which is a smaller dataset containing 494,414 images of 10,575 identities.

Except for the fourth case, FaceNet and ArcFace performed comparably across all scenarios. When compared to the other circumstances, scenario 2 showed the least performance loss. The model's performance with partial face inputs depended on the dataset used to train the feature extractor, according to an intriguing conclusion from the fourth scenario. When given incomplete face regions, models trained on larger datasets showed stronger robustness.

4.2. **Processing time.** Another metric obtained from the experiment is processing time, which is presented in Table 3. MobileFaceNet outperformed all the other tested models in terms of the time required to do face recognition. This is an expected result since MobileFaceNet utilizes MobileNetV2 as the model backbone architecture [12]. On the other hand, VGGFace performed the worst as it is a "very deep" CNN model constructed with 16 layers of convolutional layers [13]. Meanwhile, ArcFace was implemented using ResNet34 and FaceNet with Inception v1, with both having a similar number of parameters compared to the other two, with FaceNet containing fewer parameters than ArcFace, thus performing as expected.

Table 3. Average processing time

Platform	Model	Avg time (s)
Raspberry Pi	ArcFace	0.80298
	FaceNet	0.41194
	MobileFaceNet	0.27015
	VGGFace	1.97614
Personal Computer	ArcFace	0.09719
	FaceNet	0.08636
	MobileFaceNet	0.05257
	VGGFace	0.37091

4.3. **Summary of result and discussion.** It is crucial to highlight that the results of this study may differ from those of previous studies. A prior work by Wirianto and Mauritsius [22], for example, attained a high accuracy of 97.13% using the ArcFace pre-trained model. It is worth noting, however, that their study concentrated on recognizing faces with surgical masks while also considering other variables such as glasses and varied positions without masks. In contrast, the current study assesses the models' effectiveness in recognizing faces with various types of masks, such as surgical, N95, KN95, and cotton masks. The results of this investigation may not be as accurate as those of previous study, but they do provide useful insights into the models' performance, particularly on masked faces.

The experiment results indicate that ArcFace performs better than the other models in scenario 2. However, FaceNet demonstrates more robust performance across the tested scenarios, particularly in scenarios 3 and 4, where other models experience a significant decrease in accuracy. Additionally, FaceNet is a more suitable model for use on a Raspberry Pi due to its superior performance time compared to ArcFace, while still maintaining a comparable accuracy.

The study reveals that feature extraction models pre-trained on larger datasets, such as the VGGFace dataset, tend to perform better when presented with only a partial face region, as demonstrated in scenario 4 by the FaceNet and VGGFace models. This suggests that models trained on a larger and more diverse dataset can generalize well to recognize faces even with limited visual information.

Overall, the results of this study provide insights into the performance of different face recognition models on masked faces, considering various mask types. The findings contribute to understanding the strengths and limitations of the models in real-world scenarios involving face recognition under mask-wearing conditions.

5. **Conclusion.** This study compares existing pretrained face feature extraction algorithms in the context of masked faces in devised scenarios, including previously neglected processing time measures. The benefit of this study is that it identifies a lightweight (low processing time) face recognition system with good accuracy in the context of masked faces. This study gives practical insights for the creation of face recognition systems that may be efficiently deployed on resource-constrained devices by taking both accuracy and processing time into account.

The evaluation is performed on several generated masked datasets representing different scenarios for face recognition with masked faces. Results show a decline in performance of existing models when faced with masked subjects, with the least difference in performance found in scenarios where the classifier was trained on masked faces. ArcFace performs better than other models in terms of accuracy when presented with masked subjects, as shown in scenario 2, but requires twice the processing time of FaceNet to achieve similar results. These findings suggest that FaceNet is more suitable for use as a face recognition model on Raspberry Pi due to its shorter processing time and comparable accuracy.

However, it is important to note that the datasets used to train these pretrained models may have an impact on their performance, as shown in scenario 4. Further studies could be conducted by training models on the same dataset for a more general comparison. Additionally, an actual face recognition system applicable for in-person, hybrid, and remote learning models could be developed based on these findings.

REFERENCES

[1] Utah Education Policy Center, *Research Brief: Chronic Absenteeism*, http://uepc.ed.utah.edu, 2012.
[2] H. N. Chang, K. Gee, B. Hennessy, D. Alexandro and A. Gopalakrishnan, *Chronic Absence Patterns and Prediction during COVID-19: Insights from Connecticut*, www.attendanceworks.org, 2021.

[3] UNICEF, *No Excuses. Keep Schools Open. Children Can't Wait*, https://www.unicef.org/press-releases/no-excuses-keep-schools-open-children-cant-wait, Accessed on January 27, 2022.

[4] S. Bhattacharya, G. S. Nainala, P. Das and A. Routray, Smart Attendance Monitoring System (SAMS): A face recognition based attendance system for classroom environment, *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, pp.358-360, DOI: 10.1109/ICALT.2018.00090, 2018.

[5] P. Kovelan, N. Thisenthira and T. Kartheeswaran, Automated attendance monitoring system using IoT, *2019 International Conference on Advancements in Computing (ICAC)*, pp.376-379, DOI: 10.1109/ICAC49085.2019.9103412, 2019.

[6] B. Swain, J. Halder, S. Sahany, P. P. Nayak and S. Bhuyan, Automated wireless biometric fingerprint based student attendance system, *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)*, pp.1-7, DOI: 10.1109/ODICON50556.2021.9428983, 2021.

[7] V. M. Vinod, G. Murugesan, V. Mekala, S. Thokaiandal, M. Vishnudevi and S. M. Siddharth, A low-cost portable smart card based attendance system, *IOP Conference Series: Materials Science and Engineering*, vol.1012, no.1, 12046, DOI: 10.1088/1757-899x/1012/1/012046, 2021.

[8] S. M. Bah and F. Ming, An improved face recognition algorithm and its application in attendance management system, *Array*, vol.5, 100014, DOI: 10.1016/j.array.2019.100014, 2020.

[9] M. Arsenovic, S. Sladojevic, A. Anderla and D. Stefanovic, FaceTime – Deep learning based face recognition attendance system, *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, pp.53-58, DOI: 10.1109/SISY.2017.8080587, 2017.

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

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

[12] J. Deng, J. Guo, N. Xue and S. Zafeiriou, *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*, https://github.com/, 2018.

[13] 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.

[14] 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.

[15] O. M. Parkhi, A. Vedaldi and A. Zisserman, Deep face recognition, *British Machine Vision Conference*, 2015.

[16] N. Damer, J. H. Grebe, C. Chen, F. Boutros, F. Kirchbuchner and A. Kuijper, The effect of wearing a mask on face recognition performance: An exploratory study, *arXiv Preprint*, arXiv: 2007.13521, 2020.

[17] I. Q. Mundial, M. S. Ul Hassan, M. I. Tiwana, W. S. Qureshi and E. Alanazi, Towards facial recognition problem in COVID-19 pandemic, *2020 4th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, pp.210-214, DOI: 10.1109/ELTICOM50775.2020.9230504, 2020.

[18] A. Negi, K. Kumar, P. Chauhan and R. S. Rajput, Deep neural architecture for face mask detection on simulated masked face dataset against COVID-19 pandemic, *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp.595-600, DOI: 10.1109/ICCCIS51004.2021.9397196, 2021.

[19] S. I. Serengil and A. Ozpinar, LightFace: A hybrid deep face recognition framework, *Proc. of 2020 Innovations in Intelligent Systems and Applications Conference (ASYU2020)*, DOI: 10.1109/ASYU50717.2020.9259802, 2020.

[20] Zye, *Face Recognition with Coral EdgeTPU Support Based on MobileFacenet*, https://github.com/zye1996/Mobilefacenet-TF2-coral_tpu, Accessed on February 1, 2022.

[21] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, VGGFace2: A dataset for recognising faces across pose and age, *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG2018)*, pp.67-74, 2018.

[22] 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.