International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Masked Face Detection and Recognition System Using HOG Algorithm

Maryam Sarmad M.Ali[1], Dr. Fattah Alizade[2]

Submitted: 04/02/2024 **Revised**: 12/03/2024 **Accepted**: 18/03/2024

Abstract

The emergence of COVID-19 pandemic at the end of 2019 has introduced new problem to the face detection and recognition systems as faces are covered with masks that in return lead to reduction in the accuracy of face identification. For this reason, new adaptation to the previously available methods are becoming a necessary work in this area. In this paper, at first we are going to create a masked dataset to be used in our work that will mainly be divided into two parts, first to propose an algorithm for mask detection using Viola Jones algorithm, and second to suggest a face recognition algorithm that would use two methods namely (using pre-trained convolutional Neural Network CNN architectures Resnet-50 and Mobilenet, and building a customized CNN). The evaluation of this method was done on the dataset that we created at the beginning of our work based on the use of FEI dataset. The mask detection algorithm used has provided and accuracy of 79% whereas the face recognition methods has shown accuracy ranging between 75% and 99%.

1. Introduction

The new adaptation for the life after COVID will affect many current available methods and technologies and may limit their performance and accuracy especially for face recognition systems that require capturing the whole face to be able to detect and verify the identity of that person. When wearing masks in all public places became mandatory during the COVID-19 time and being able to control individuals entering any building or closed area became necessary and machine vision technologies can make this task easier. This will allow organizations, universities, or governmental offices to apply an automatic mask detection system to check their staff before being allowed to enter their buildings that would bring some other considerations of having different types and colors of mask that people might be wearing. The remaining of the paper is organized as follows. Section 2, we provide an overview of related work. Subsequently, in Section 3, we describe the creation of the masked dataset, followed by our proposed method for masked face detection and recognition in Section 4. And finally the evaluation of our proposed method and the results achieved with the conclusion in Section 5 and 6 respectively.

2. Related Work

To be able to detect faces in an image, the first thing is that we need is to detect the facial points or components of the face, which are parts or areas of the face that are mainly located in the eyes, nose, mouth, and chin. These points can help identify or find the face in an image and even distinguish one person from another. For detecting these facial points there have been many methods and algorithms proposed by many researchers, but they are generally limited in their performance and success by some conditions like in-wild and real-world conditions (ex: different poses, illuminations, or occlusion). These points or components of the face can be detected by either locating the facial component as an object or using facial landmarks which are points on the face that make up specific areas of the face (eyes, nose, mouth, and even the jawline). Figure 2.1 shows the representation for both facial landmarks and facial components.

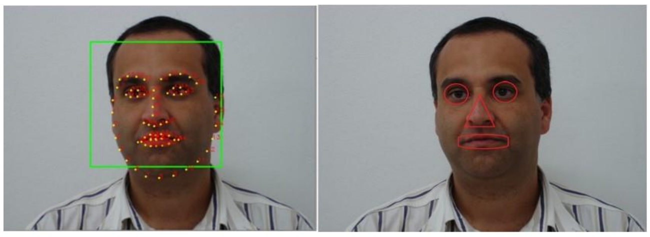


Figure 2.1

In the facial component detection using object detection, the most widely used method was proposed by (Viola and Jones, 2001). This method was based on the idea of using a window to scan the image to extract the features of the face in that image. It combines the idea of Haar features, integral image, AdaBoost method, and the use of cascade classifiers. While the work of (Kazemi and Sullivan, 2014) was based on the use of facial landmarks for face detection using an ensemble of regression trees to estimate the location of landmarks. On the other hand, in the area of partial face recognition using the Convolutional Neural Network, (Elmahmudi, 2018), have suggested the use of Convolutional Neural Network

[1]*Assistant lecturer at LFU (Lebanese French University), maryam.sarmad@lfu.edu.krd*
[2]*(Assistant professor) at UKH (University of Kurdistan/Hawler)*

(CNN) for partial face recognition. The proposed method uses a pre-trained model VGG-Face for the training, and two classifiers which are the Cosine Similarity (CS) and Linear Support Vector Machine (SVM) to test the recognition rate. They have tested their work using the publically available dataset, FEI.

3. Dataset Preparation

While conducting our research, none of the publically available datasets had facial images that have the features required for our research, which has frontal or near frontal images, and at the same time, these faces must be covered with a medical mask to test the proposed algorithm for mask detection and later for partial face recognition. For this reason, we have used the FEI dataset and applied the work of (Anwar and Raychowdhury, 2020) to add synthetic masks to our dataset and generate a new masked dataset. Figure 3.1 shows a sample image from FEI dataset before and after applying the masking algorithm.

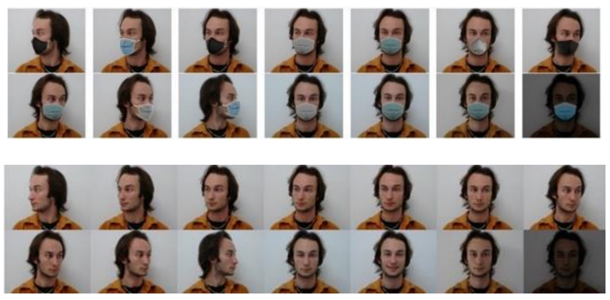


Figure 3.1

4. Proposed Method

Our proposed method is composed of the following step as show in figure 4.1

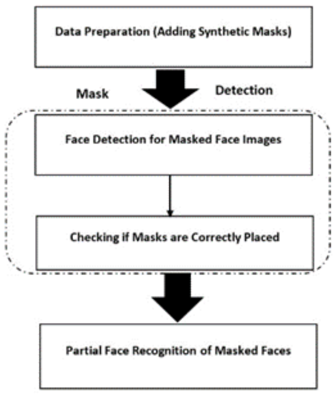
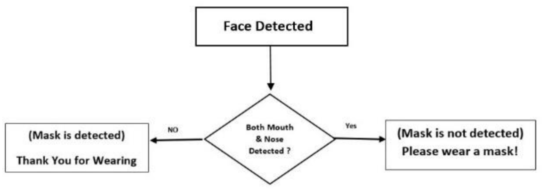

Data Preparation (Adding Synthetic Masks)
Mask
Detection
Face Detection for Masked Face Images
Checking if Masks are Correctly Placed
Partial Face Recognition of Masked Faces

Figure 4.1

4.1 Mask Detection

The first step in the mask detection phase is face detection, which is followed by our proposed method for mask detection. The Histograms of Oriented Gradients (HOG) algorithm was used for face detection by (Dalal and Triggs, 2005) that suggested the use of HOG with linear Support Vector Machine (SVM) for classification. Figure 4.2 shows a sample result of face detection. Next, the detected face is checked for having masks covering or not. Our proposed algorithm is based on the work by (Viola and Jones, 2001) for object detection to check whether it can find nose and mouth in the face or not. As shown in figure 4.3, the proposed method works as follows: Checks the face to see whether it can find both nose and mouth, if it finds either a nose or mouth it would show a message “Please wear a Mask”, otherwise, if none of them are detected, it assumes that they are covered with a mask and cannot detect them and shows a message that says, “Thank You for Wearing a Mask”


Face Detected
Both Mouth & Nose Detected ?
NO
Yes
(Mask is detected) Thank You for Wearing
(Mask is not detected) Please wear a mask!

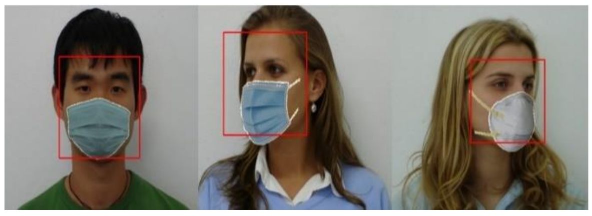


Figure 4.3 Figure 4.2

Figure 4.4 below shows how the proposed method works and how it finds the nose and mouth location in an image.

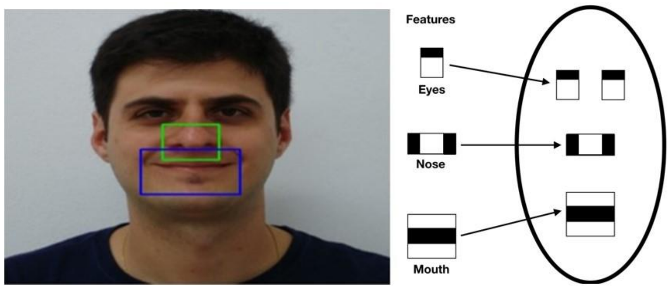


Figure 4.4

4.2 Partial Face Recognition

The second phase of the proposed method is the partial face recognition for the masked faces. This part was implemented through the use of the convolutional neural network CNN. In order to build and train a CNN, there are different techniques such as, 1: building and training a customized Convolutional Neural Network CNN with a specific number of layers and with the use of data prepared for that task, 2: transfer learning technique in which a pre-trained model is used, which are deep learning models that have been built and trained for a specific task or to solve a specific problem. Several available pre-trained models can be used for transfer learning like Resnet-50 (He et al, 2015), VGG-16

(Simonyan and Zisserman, 2015), InceptionV3 (Szegedy et al, 2015), MobileNet (Howard et al, 2017), and others. The models that will be used in this research are MobileNet, Resnet-50, in addition to building a customized CNN without the use of transfer learning techniques. The reason for choosing Resnet for this research is that it uses residual unit with bottleneck architecture that reduces the overall computational cost, on the other hand, choosing MobileNet is due to its small size and the high speed it provides that could adapt to the mobile requirements if the application to be used on smart phones. The following sub-sections will present in detail the steps used for partial face recognition starting with pre-processing phase, followed by the implementation of each model used.

4.2.1 Pre-processing Phase

The pre-processing requirements depend highly on the algorithm and the architecture of the learning process and the type of convolutional neural network used. For This particular research, the pre-processing steps are:

- **Image Resizing**

The images that are used from the FEI dataset are of size 640 x 480 that will be resized to 224x224 as it is one of the requirements for the models used (Resnet and MobileNet) is to have an image of that size.

- **Image Cropping**

The images will be cropped around the upper part of the face to discard the part of the face that is covered by a mask which may not be useful during the learning process. This process will yield a new dataset that consists of images of the upper part of the face (eyes, forehead, etc.) which is named **"Cropped Dataset"**. A copy of the original dataset (i.e., without cropping) named **"Full Face Dataset"** will be kept to be used in the training and testing of the models as another scenario. Figure 4.5 shows a sample of the Cropped and Full Face dataset.

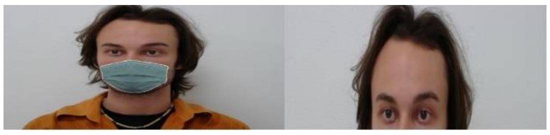


Figure 4.5

- **Dataset Partitioning**

To be able to run a CNN model, it is required to have three different sub-datasets, namely (training, validation, and testing) datasets. Since FEI has all the images as one dataset, dividing them into three separate parts is required as a preprocessing phase before starting to train the networks. The percentage used for splitting the datasets has followed best practices suggested in the literature and is shown in table 4.1 below.

Training	Validation	Testing
70%	15%	15%

Table 4.1

4.2.2 Learning Phase

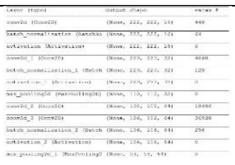
After the preprocessing step, the training and learning process starts using two approaches: Transfer Learning, and Building a Customized Convolutional Neural Network CNN and as below.

- **Transfer Learning**

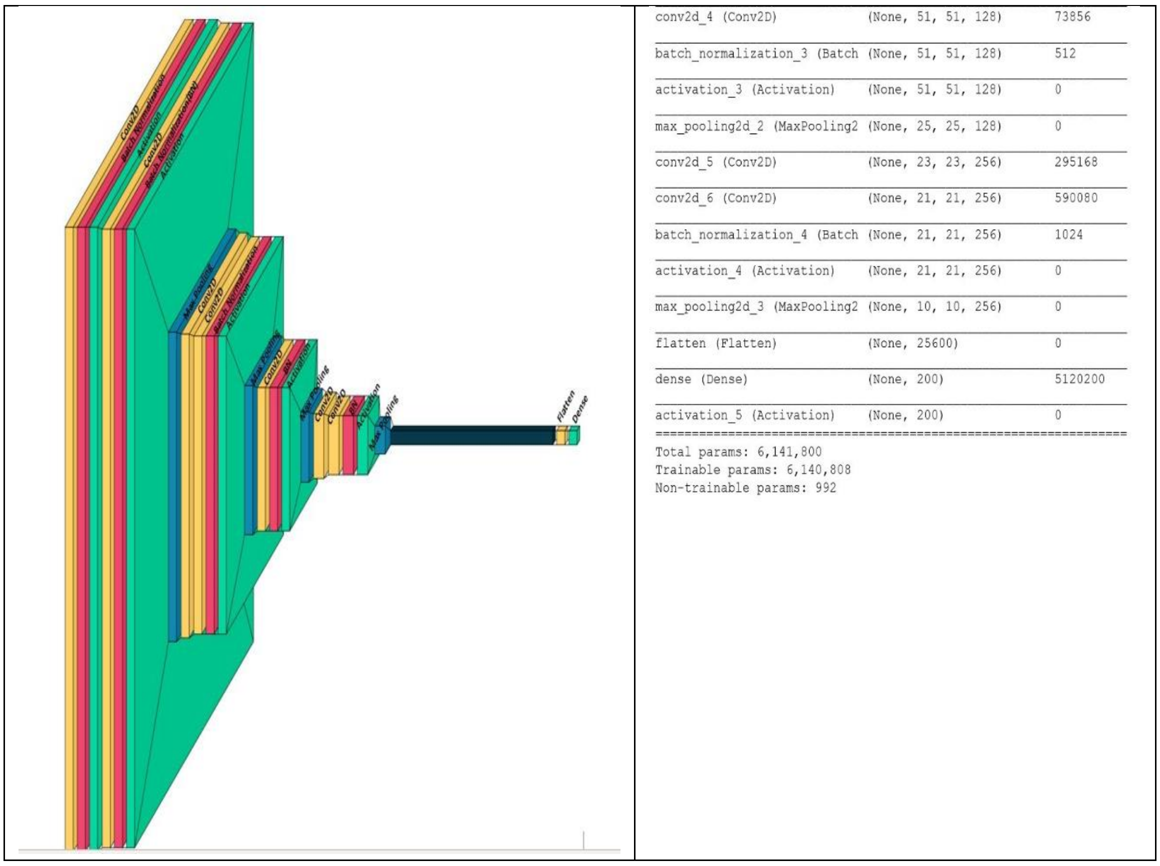
In transfer learning, we have used the pre-trained models (Resnet-50, MobileNet). Both of the models were used for the partial face recognition problem by importing all the layers of that model except for the top layer and the output layer so that the model will work on the data we provide and not on what it was originally trained for. The top layer or input layer will be given once the **Full face dataset** and another time the **Cropped Face dataset** (using both training and validation datasets). In order for the model to provide predictions based on the new data and according to the number of classes the dataset has, the output layer of the model will be changed to comply with the data specific for this problem, which is in our case 200 different classes or faces. A Dense layer will be added with a softmax activation function. Table 4.2 shows the configuration settings used for both Resnet-50 and MobileNet models.

- **Building a Customized CNN**

The second method for the face recognition is to build and train a customized neural network following the work of (Patrik et al, 2017). The network architecture is shown in figure 4.6



```
Layer (type)                  Output Shape             Param #
conv2d (Conv2D)               (None, 222, 222, 16)     448
batch_normalization (BatchNo  (None, 222, 222, 16)     64
activation (Activation)       (None, 222, 222, 16)     0
conv2d_1 (Conv2D)             (None, 220, 220, 32)     4640
batch_normalization_1 (Batch  (None, 220, 220, 32)     128
activation_1 (Activation)     (None, 220, 220, 32)     0
max_pooling2d (MaxPooling2D)  (None, 110, 110, 32)     0
conv2d_2 (Conv2D)             (None, 108, 108, 64)     18496
conv2d_3 (Conv2D)             (None, 106, 106, 64)     36928
batch_normalization_2 (Batch  (None, 106, 106, 64)     256
activation_2 (Activation)     (None, 106, 106, 64)     0
max_pooling2d_1 (MaxPooling2  (None, 53, 53, 64)       0
```



```
conv2d_4 (Conv2D)            (None, 51, 51, 128)       73856
_________________________________________________________________
batch_normalization_3 (Batch (None, 51, 51, 128)       512
_________________________________________________________________
activation_3 (Activation)    (None, 51, 51, 128)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 25, 25, 128)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 23, 23, 256)       295168
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 21, 21, 256)       590080
_________________________________________________________________
batch_normalization_4 (Batch (None, 21, 21, 256)       1024
_________________________________________________________________
activation_4 (Activation)    (None, 21, 21, 256)       0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 10, 10, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 25600)             0
_________________________________________________________________
dense (Dense)                (None, 200)               5120200
_________________________________________________________________
activation_5 (Activation)    (None, 200)               0
=================================================================
Total params: 6,141,800
Trainable params: 6,140,808
Non-trainable params: 992
```

Figure 5.6

Using this network for learning different features from the training dataset, that was first put into a data augmentation function to increase the number of samples in the training set by performing some transformation to the original images (flipping, rotating, and shearing) that would reduce the possibility of over and under-fitting problem. The validation dataset is also used during the training process to measure the performance of the model during the training using unseen data. Figure 5.7 shows a sample data augmentation performed.

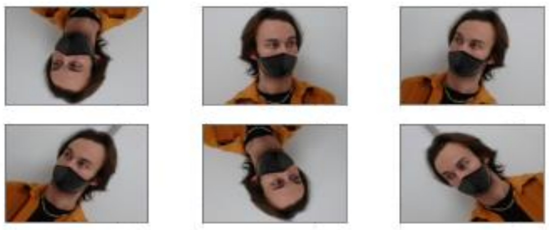


Figure 4.7

In this customized network, a Batch Normalization (BN) layer is added after each input layer, which is useful in reducing the problem of covariate shift, where during training the distribution of the data changes and that leads to slower training and will in return reduce the effect of the overfitting problem and works well in generalizing data.

- **Datasets**

The datasets used for both training and testing where both of the **Full face dataset** and the **Cropped Face dataset** which were implemented as two different scenarios.

5. Evaluation and Results

The training was done on NVidia GeForce (MX130) GPU, with a machine of Core i5 processor and 8 GB RAM. For training large CNN, GPU is used for higher performance. To implement the research, Jupyter notebook was used with Python programming language version 3.6.7. A Tensorflow version 2.4.1 with Keras version 2.4.3 was used. The following section will show the results achieved for each part of our proposed method.

5.1 Mask Detection

For mask detection phase, the results achieved using the Viola Jones algorithm has shown 79% in detecting whether the person in the image is wearing a mask or not. Figure 5.1 shows a sample result of a successful mask detection.

Figure 5.1

5.2 Face Recognition

For evaluating the results achieved in this part of the research, a test dataset will two different scenarios are used (Full Face, Cropped) and as shown in table 5.1 below

Dataset	No. of classes	Total No. of images in the test dataset	No. of test images per class
Full-face	200	400	2
Cropped	200	400	2

Table 5.1

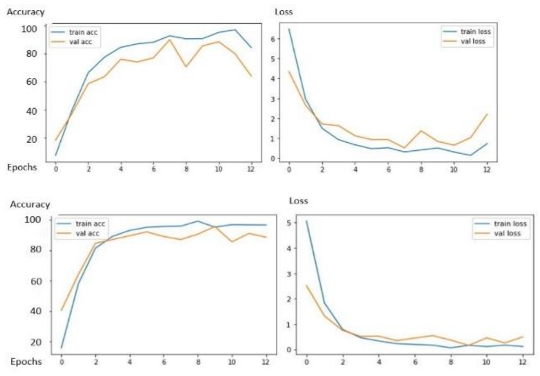
The following section will show the results achieved for each scenario with each dataset

5.2.1 Training the Models

- **First Scenario: Using full-face dataset**

This section shows how the Resnet-50, MobileNet, and customized CNN were implemented on this dataset.

The Resnet-50 has shown an accuracy of 75% when implemented on this dataset, MobileNet has shown 86%, while our customized CNN has provided 93% accuracy in face recognition, and as shown in figure 5.2



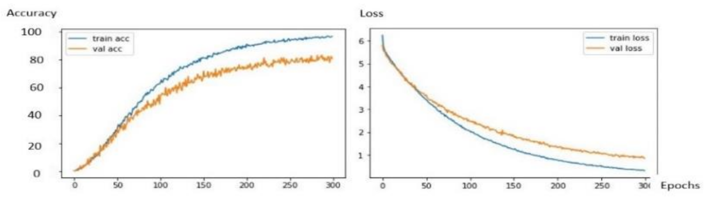


Figure 5.2

- **Second Scenario: Using Cropped Dataset**

In this part, the cropped dataset will be used for training all of the three models: Resnet-50, MobileNet, and the customized CNN, and as in the previous scenarios.

The Resnet-50 has shown an increase in the accuracy to reach 98%, the same applies for MobileNet which has provided 99% accuracy. While our customized CNN had the same accuracy provided for the previous scenario with 93% accuracy, as shown in figure 5.3

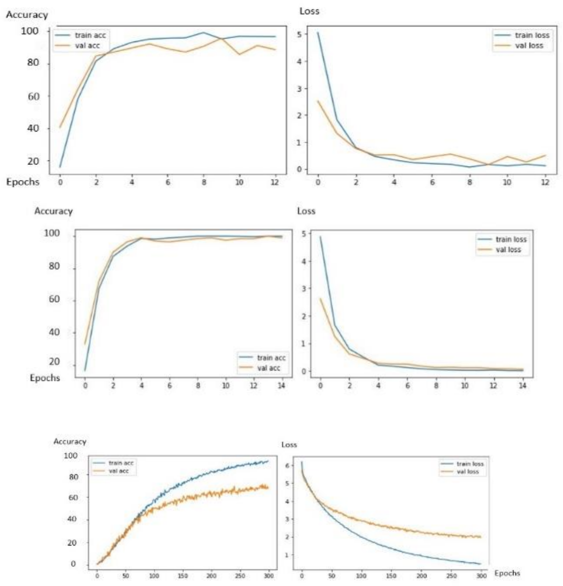


Figure 5.3

5.2.2 Testing the Trained Models

Now that our models are trained using our masked dataset. It is time to test these models on new unseen data, using the test data from both the full-face dataset and cropped dataset by predicting the classes of the given images. The prediction result are shown in figure 5.4.

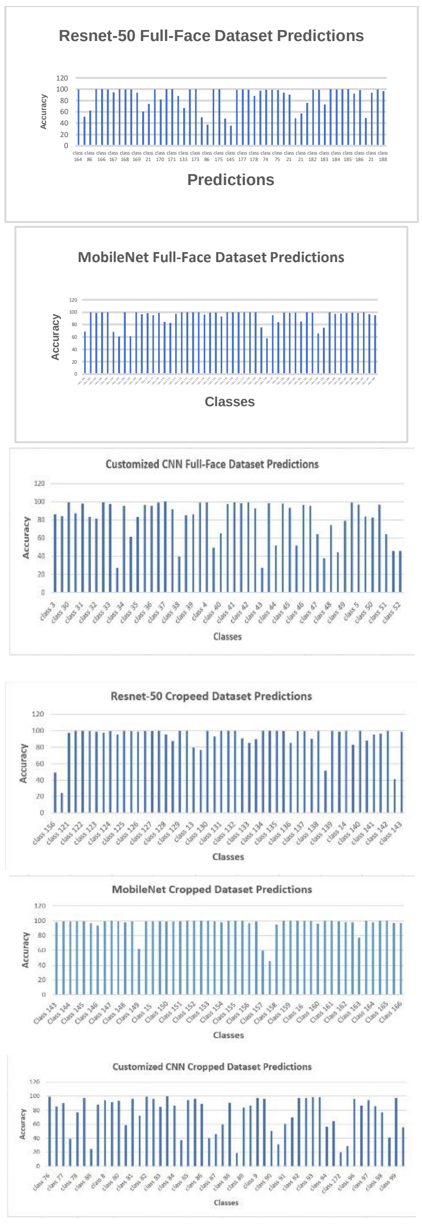


Figure 5.4

6. Conclusion and Future work

This work has mainly focused on the face recognition in masked faces using two different CNN techniques. The results has shown that when the network is trained via the cropped dataset, the recognition accuracy increases because CNN will focus only on the upper part of the face and discards the parts that are covered with masks which might lead to mispredictions. On the other hand, the use of transfer learning technique has shown higher accuracy and faster implementation over the use of a customized CNN as the pre-trained models does not require to re-train the whole network again which will save much more time. Our future work would be to apply the CNN techniques to improve the accuracy of our mask detection method.

References:

[1] Kazemi, Vahid & Sullivan, Josephine. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. 10.13140/2.1.1212.2243.

[2] Viola, Paul & Jones, Michael. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conf Comput Vis Pattern Recognit. 1. I-511. 10.1109/CVPR.2001.990517.

[3] A. Elmahmudi and H. Ugail, "Experiments on Deep Face Recognition Using Partial Faces," 2018 International Conference on Cyberworlds (CW), Singapore, 2018, pp. 357-362, doi: 10.1109/CW.2018.00071.

[4] Anwar, A., & Raychowdhury, A. (2020). Masked face recognition for secure authentication. *arXiv preprint arXiv:2008.11104*.

[5] Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893). Ieee.

[6] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[7] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[9] Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[10] Kamencay, Patrik & Benco, Miroslav & Mizdos, Tomas & Radil, Roman. (2017). A New Method for Face Recognition Using Convolutional Neural Network. Advances in Electrical and Electronic Engineering. 15. 10.15598/aeee.v15i4.2389.

[11] C. E. Thomaz and G. A. Giraldi. A new ranking method for Principal Components Analysis and its application to face image analysis, Image and Vision Computing, vol. 28, no. 6, pp. 902-913, June 2010.