



## Occluded Facial Recognition for Surveillance Using Deep Learning

---

Hameed Moqbel and Murali Parameswaran

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 17, 2022

# OCCLUDED FACIAL RECOGNITION FOR SURVIELLANCE USING DEEP LEARNING

Hameed Moqbel

*Department of Computer Science Engineering  
Presidency University, Bangalore, India*

[almoqryhameed@gmail.com](mailto:almoqryhameed@gmail.com)

Murali Parameswaran

*Department of Computer Science Engineering  
Presidency University, Bangalore, India*

[murali.p@presidencyuniversity.in](mailto:murali.p@presidencyuniversity.in)

**Abstract** - Nowadays, due to the advancement in technology, facial recognition is becoming one of the methods to identify a person. One of the challenges arises due to occlusion or partial covering of face, especially with a facial mask or a scarf. In this project, we use deep neural networks to solve the problem of recognizing such an occluded face. For this work, we have used three publicly available facial datasets, namely Labelled Face Wild dataset, COMASK20 and Specs on Faces (with images having low illumination), cumulatively consisting more than 5000 facial images. We evaluated four existing facial detection classifiers namely OpenCV, SSD, MTCNN and RetinaFace. We found that MTCNN to be most relevant for our work. We proposed a new Convolutional Neural Networks (CNN) as part of this work. We got accuracy of 99.38% for LFW, 99.62% for COMASK20 and 98.33% for SOF dataset.

*Index Terms* – Labelled Face Wild dataset, Specs on Face Convolutional neural Network

## I. INTRODUCTION

Biometric recognition software is becoming more important in security, administration and business systems, the biometric systems can be finger print, iris recognition, retinal scanning, voice recognition and facial recognition. Facial recognition has sparked a lot of attention since it concentrates on detection, identification and verification. It is now used in public places embedded with surveillance cameras, airport security and law enforcement agencies. Face recognition has been an important topic in scientific community for the past five years; nonetheless, it is still an unresolved problem that failed to recognize faces under occlusion. Occlusion can be due to multiple reasons like facial accessories such as eyeglasses, sunglasses, scarves, mask, hat, hair, and factors like extreme illumination.

During Covid 19 pandemic, people used to wear masks wherever they go to prevent themselves and others from infection by this virus. Wearing mask becomes like face accessories and this increase the difficulty of face recognition. We need to develop occluded face recognition software that can help us to recognize the occluded faces of criminals, shoplifters and wanted people.

Occluded face recognition system focused on detection and recognition faces in which part of the faces are occluded, whether occlusion can be by mask or by any other accessories, therefore some of face features will be difficult to extract like

mouth, nose and chin in case of mask or eyes and eyebrows in case of sunglasses. Occluded face recognition depends on analysing and extracting face features but face detection of occluded faces is taken into consideration as the first step. Face detection frame works are divided into two categories, first one depends on CNN which called two stage detectors such as MTCNN, RetinaFace, RCNN and fast RCNN. The second category does not depend on CNN which called one stage detectors such as the Single-Shot multibox Detector (SSD), Open CV, Haarcascade Face detector and YOLO. The two stage detectors outperform better than the one stage detectors but it takes longer time, in the other hand, the one stage detectors take less time consumption but achieve less accurate detection result.

In this work, we developed a new CNN architecture to improve occluded face recognition. We attempt four different techniques of face detection, where two techniques are taken from the first category and two techniques from the second category. We compared those techniques with different parameters such as consuming time of detecting faces in data, training time, accuracy, precision and F1-score. The techniques of face detection are applied on different three publically datasets. LFW (Labelled Face Wild dataset), COMASK20-master and Specs on Faces (SoF). Finally, we applied the best model on the three datasets and compared the result.

The rest of this report is organized as follows. In section 2 related work shows other researchers work on occluded face recognition. Section 3 shows the proposed work that we attempted in this paper. Section 4 includes network architecture that defines the architecture of CNN. Section 5 includes the result of comparison between different models on different datasets. In section 6 conclusion and future work are presented. Finally references of all papers are presented in section 7.

## II. RELATED WORK

In this section, we will discuss the CNN architectures that have been proposed in the past, so we will show the architecture of different CNN with the result of its models.

In Fig 1, the architecture proposed by Syafeeza et al, in [5] is shown, in their work, they used two CNN architectures where each one contains of 2 convolutional layers, 2 subsampling layers and 2 fully connected layers. In Fig. 1(a), the

architecture designed for frontal images with occlusion, where the architecture in Fig. 1(b). designed for illumination variation, pose variation and facial expression. The CNN architecture is governed by three parameters which are number of layers, number of feature map in each layer and their connectivity. The CNN architecture achieved the highest accuracy in terms of z-score normalization and Gaussian weight initialization methods are applied. They got accuracy of 99.50% on AR dataset and 85.16% on FERET dataset.

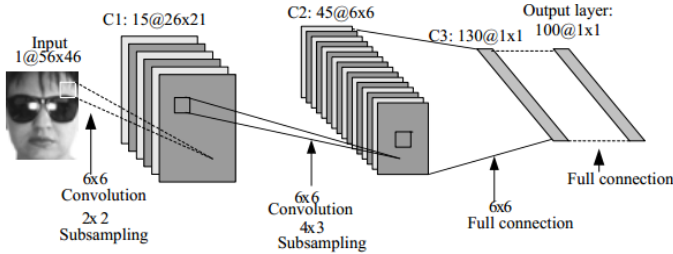


Fig. 1(a). CNN architecture made by Syafeeza et al, in [5]

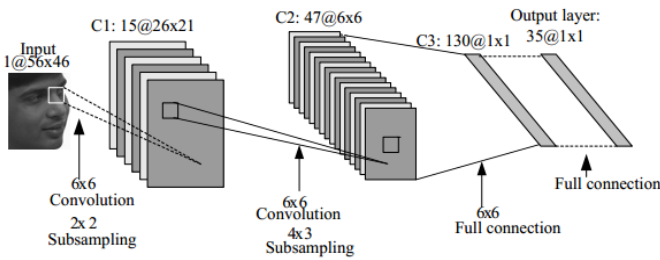


Fig. 1(b). CNN architecture made by Syafeeza et al, in [5]

In Fig 2, the architecture proposed by Young et al [2] is shown. In their CNN architecture, they used 8 convolutional layers with corresponding maxout and max pooling operators and 3 fully connected layers. They used two CNN architecture channels for occluded and non-occluded faces. Features of the occluded faces and non-occluded faces can be extracted from the shared layer and apply comparison method using cosine method. In Fig. 2, a sample image used by them is shown. The CNN model applied on three publically datasets and they compared result in terms of accuracy and they got accuracy of (CACD 99.12%, YTF 97.30%, LFW 99.73%).

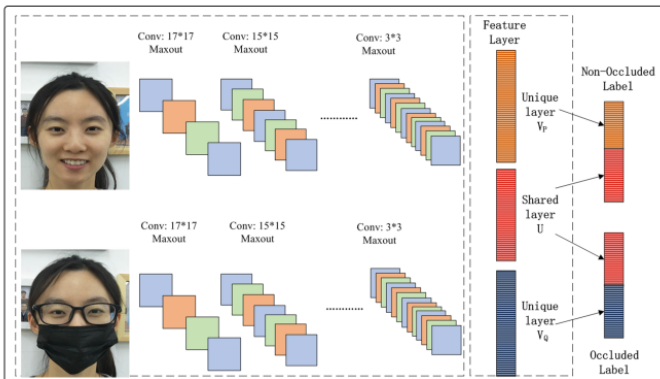


Fig. 2. CNN architecture, in [2]

In Fig. 3, the CNN architecture proposed by Wu et al in [1], they used 2 convolutional layers, 2 subsampling layers and 2 fully connected layers as it is shown in figure 2-3. In the convolutional neural network, the input layer is connected with the convolutional layer and the middle layer is connected with convolutional layer and down sampling layer. The fully connected structure helps to find the output of the whole network. The result of this work is 98.6% recognition rate of masked faces.

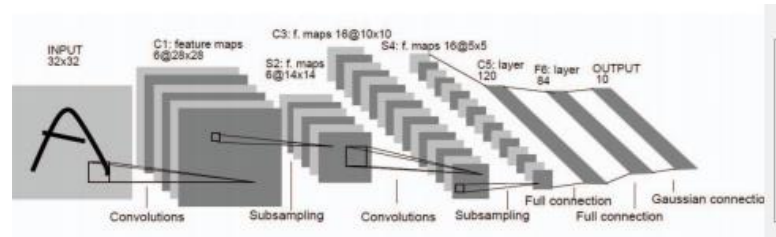


Fig. 3 CNN architecture, in [1]

### III. PROPOSED METHOD

The project proposes a pipeline to build and develop a deep learning model for occluded facial recognition systems. In this paper, we will use different face detection classifiers with a new CNN architecture and apply it on three datasets (LFW, COMASK20, SoF). Then we will do comparison between the face detection classifiers and choose the optimal one to apply it on the datasets. We are going to discuss and compare the result in terms of accuracy. Fig. 4. shows the pipeline of this project.

#### A. Dataset

In this project, we used three public datasets Labelled Face Wild dataset (LFW), COMASK20-master and Specs on Faces (SoF) dataset. Each one of these datasets has different occlusions.

- *Labelled Face Wild dataset (LFW)*

LFW dataset is one of the most public dataset that used in face recognition projects, so many papers are done on this dataset. This dataset contains of 13,233 images of 5749 subjects. Viola Jones face detector is used to detect faces with specific size and position. Each subject has one file with two or more distinct photos. The dataset is collected from web and maintained by researchers at the University of Massachusetts, in [6]. Samples of the dataset are shown in Fig. 5.



Fig. 5 sample of LFW dataset

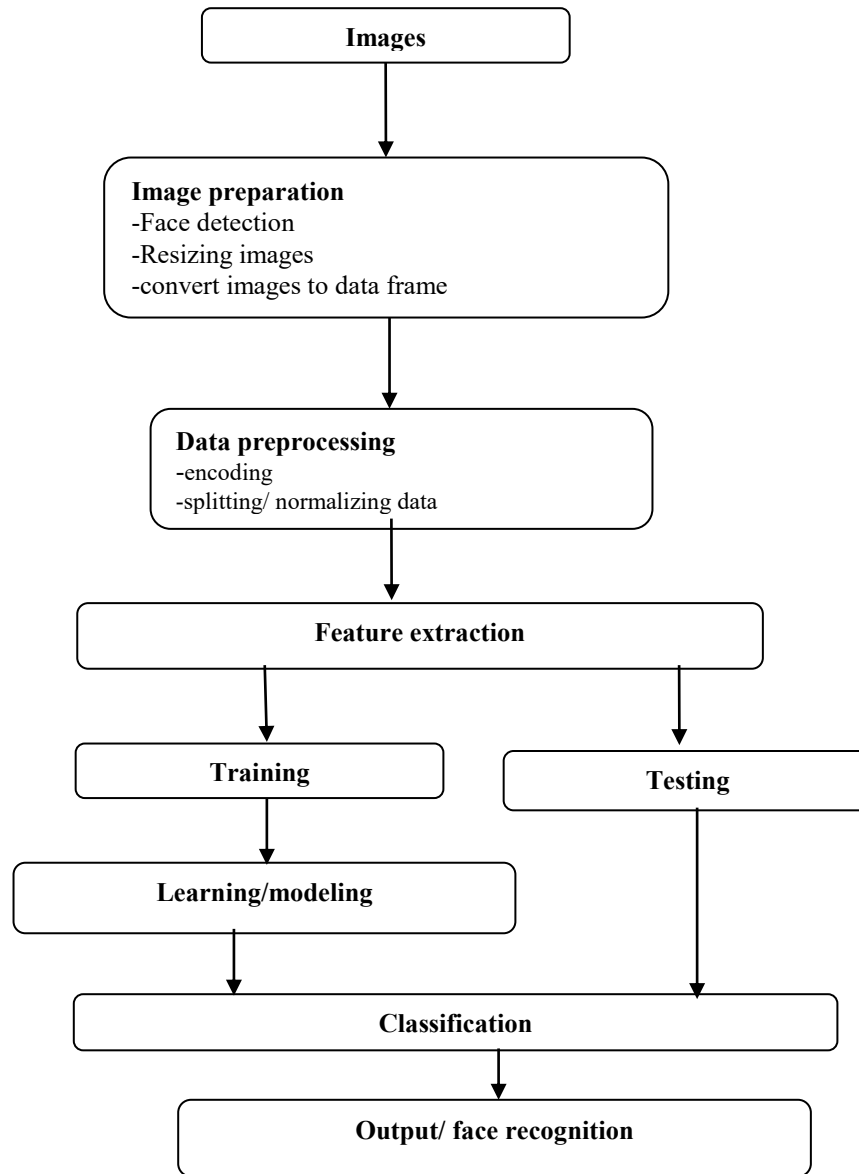


Fig. 4. proposed work pipeline

- *COMASK20-master*

COMASK 20 dataset is created and collected by (Vu, Hoai and Nguyen, Mai and Pham, Cuong), in [4]. The dataset contains 2754 facial images labeled for 300 subjects with different identities. For each subject there is one file which contain images with mask and without mask. this dataset has occlusion as masked faces and illumination factor. Samples of dataset are shown in Fig. 6.

- *Specs on Faces (SoF) dataset:*

SoF dataset is created and collected by Afifi, Mahmoud and Abdelhamed, Abdelrahman, in [7]. This dataset contains of

42,592 (2,662\* 16) images for 112 persons (66 males and 46 females). The occlusion in this dataset is face accessories and different illumination conditions. Most of the subjects are wearing glasses which is the common among them. This dataset is used in face detection, recognition and classification. Samples of dataset are shown in Fig. 7.

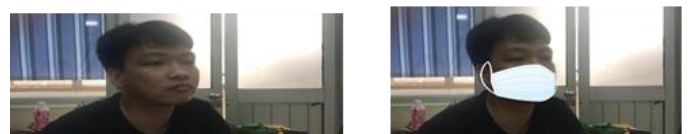


Fig. 6. sample of COMASK datasets



Fig. 7. sample of SoF datasets

### B. Face detection classifiers

In this approach we used pre-trained face detection classifiers that are included with DeepFace open source library. Deepface is a face recognition library for python, which includes all AI models of face recognition and face detection. All models in this library are pre-trained models which can be called by importing the library and passing the exact image path as the input and collect the result. In this section, we attempted 4 face detection and we did comparison between them in terms of accuracy of detecting the face, uploading time of data, training time accuracy. In this comparison, we used COMASK 20 dataset and we apply these classifiers on it to see which one is the best face detection classifier.

- *OpenCV*

OpenCV is a face detector which based on machine learning and it uses Haar cascade algorithm, so it is fast as it is not based on CNN. It works properly only for frontal images. The performance of detecting the eye is average which causes alignment issues. Result of OpenCV face detection on COMASK 20 is shown on Fig. 8.



Fig. 8. sample of images detected by OpenCV classifier

From the above figure, we can see that there is false positive classification and the classifier failed to detect the face in some images.

- *SSD*

SSD is a single shot detector which depends on CNN. This detector detects the object in the image in the first shot. Unlike the other models which pass the image more than once to

detect an object. As it can detect the object from the first pass, it doesn't need much time to detect objects, so it is fast, at the same time it has a good accuracy in its detection. SSD detector depends on OpenCV's eye detection module, so alignment issues will be there. Result of applying SSD classifier on COMASK20 dataset will be shown in Fig. 9. From the figure we can notice that SSD failed to detect faces in some images especially if the subjects have occluded faces.

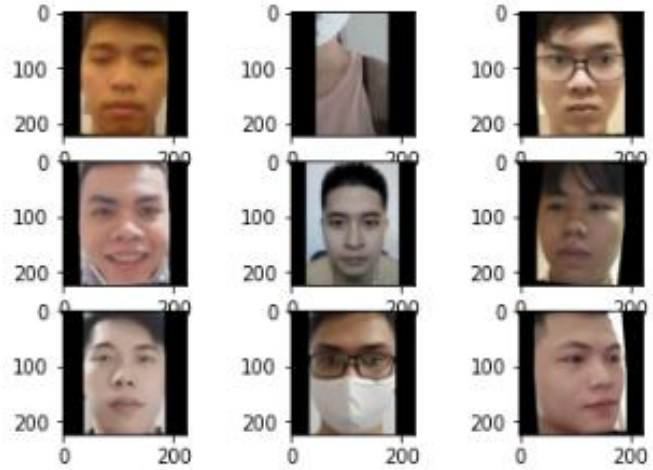


Fig. 9. sample of images detected by SSD classifier

- *MTCNN*

Multi-Task Cascaded Convolutional Neural Networks is a detector that based on deep learning (CNN). It is used to detect faces and facial landmarks, so both detection and alignment scores are high, however it is slower than SSD and OpenCV. MTCNN is one of the most popular face detectors used today. The result of applying this classifier on the COMASK20 dataset is shown in Fig. 10. From figure we can see that all faces are detected well.

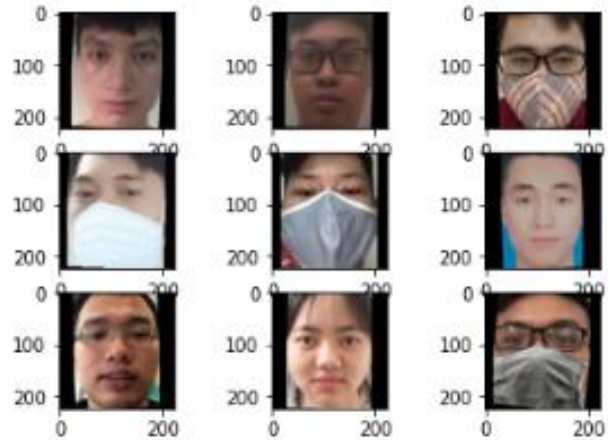


Fig. 10. sample of images detected by MTCNN classifier



- *RetinaFace*

RetinaFace is a deep learning-based model used to detect faces and facial landmarks. It is used to detect 2D face alignment and 3D face reconstruction. It requires high computation time, so it is the slowest face detector compared to the others. Result of applying this detector on COMASK20 dataset is shown in Fig. 11.

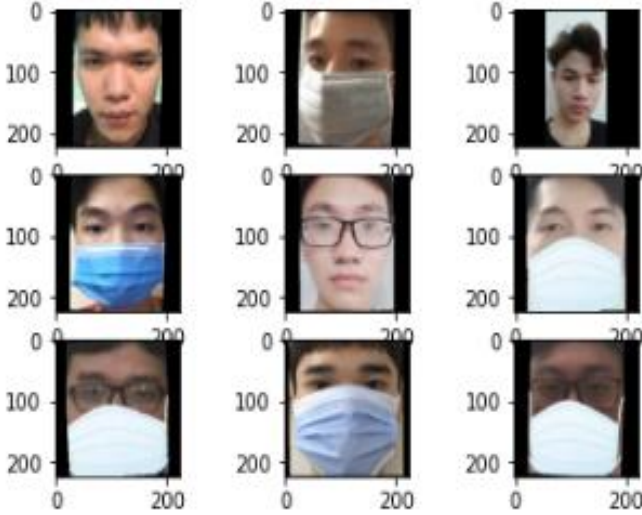


Fig. 11. sample of images detected by RetinaFace classifier

### C. Comparison between face detection classifiers

We apply the four face detection classifiers and we compared between them in terms of many parameters as it is shown in TABLE 1 (a, b).

From Table 1(a), we can see that MTCNN and RetinaFace classifiers have a true positive classification. All faces are detected perfectly and we noticed that OpenCV and SSD classifiers have a false positive classification. The two classifiers failed to detect some faces especially occluded faces.

From TABLE 1(b), we can observe that MTCNN detector takes long time of uploading data and detecting faces. RetinaFace detector has the highest training time with highest evaluation accuracy, also minimum loss and minimum training accuracy. OpenCV achieved the highest train accuracy and the minimum training loss. SSD achieved the minimum execution time of uploading training data and the minimum evaluation accuracy with maximum evaluation loss. After comparison, which face detector should we choose?

we can choose the proper face detection classifier by depending on application requirements and our dataset. as we apply all detectors on COMASK 20 dataset and we noticed that OpenCV and SSD detectors failed to detect faces with occlusion, so we have to choose MTCNN and RetinaFace but as we noticed that RetinaFace has the highest training time and the lowest evaluation accuracy, so MTCNN is the best detector among others.

TABLE 1(a) comparison between face detection classifiers, in terms of detecting faces

Dataset	OpenCV	SSD	MTCNN	RetinaFace
COMASK20				

## IV. NETWORK ARCHITECTURES

Convolutional Neural Network is the most important type of Deep Neural Network that can extract features in images and do classification. CNN has two main parts; the first part is convolutional layers which are responsible for the process of feature extraction, the second part is fully connected layers and these layers are responsible for classification, in addition to these layers, it has three important parameters, those parameters are dropout layer, batch normalization layer and activation function. Convolutional layer is the most important part in CNN architecture which carries out the network's computational load. The objective of convolutional computation is to extract the high-level features such as edges, nose, mouth, chin and eyes. The next layer is pooling layer, this layer usually follows convolutional layer, and this layer is responsible for reducing the size of convolved features in order to decrease the computation power. After extracting high level features of images, fully connected dense layers come next to do the classification, so it is responsible for classification based on the features that are extracted in the previous layers. The input image is flattened and fed to the fully connected layer, this layer consists of weights and biases along with the neurons which used to connect the neurons between two different layers. When dense layers are fully connected, it can cause overfitting which means the model is working well on the training data but in test data very low performance so to solve this problem we can use dropout layer.

TABLE 1(b) comparison between face detection classifiers, in terms of time and accuracy

Evaluation parameters	OpenCV	SSD	MTCNN	RetinaFace
Execution time of uploading train data	3197.227s	1224.322s	4795.554s	1681.067s
Execution time of uploading test data	103.8175	58.174s	266.136s	126.223s
Length of train data	2824 images	2824 images	2824 images	2824 images
Length of test data	264 images	264 images	264 images	264 images
Training time	2184.063S	2186.487s	2183.509s	2304.44s
Train accuracy/loss	0.9564 / 0.144	0.9398 / 0.1768	0.9419 / 0.1683	0.9380 / 0.1955
Val accuracy/loss	0.9924 / 0.0344	0.9621 / 0.1215	0.9962 / 0.0270	0.9962 / 0.0209

In this layer few neurons are dropped randomly from the neural network during the training. The activation function is one of the most crucial parameters in the CNN architecture. It is responsible for making decision; it needs to decide which information should transfer to the next layer and which one should not be at the end of the network.

In this paper, we proposed CNN architecture with 4 convolutional layers, 2 maxpooling layers, 5 dropout layers, 1 flatten layer and 3 fully connected layers. The architecture is shown in Fig. 12.

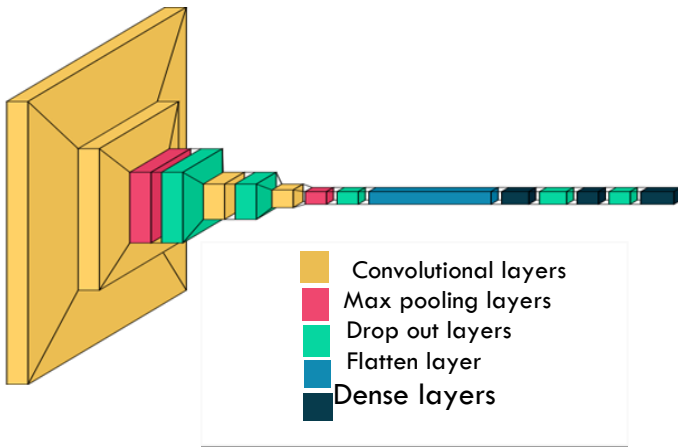


Fig. 12. CNN architecture of the proposed approach

## V. EXPERIMENTAL RESULTS

In this experiment we apply our models on three datasets by using different face detection classifiers; those are OpenCV, SSD, MTCNN and RetinaFace. We applied all approaches on the COMASK20 dataset which its occlusion is a mask which cover important features in face recognition such as mouth, nose and chain.

Here we will show the 4 approaches results on COMASK20 dataset. The same approach will apply on the dataset but we will change the face detector classifier and we will compare the result in terms of accuracy.

### A. Face detection classifiers results

- *OpenCV face detection*

In Fig. 13(a), we have observed that the model is performing well on both the training and testing data, but it seems that the model is working better on the test data, we can see the validation loss of the test data is smaller than the loss of training data and this is due to size of the test data is small compared to the train data. Sample of the predicted images are shown in Fig. 13(b).

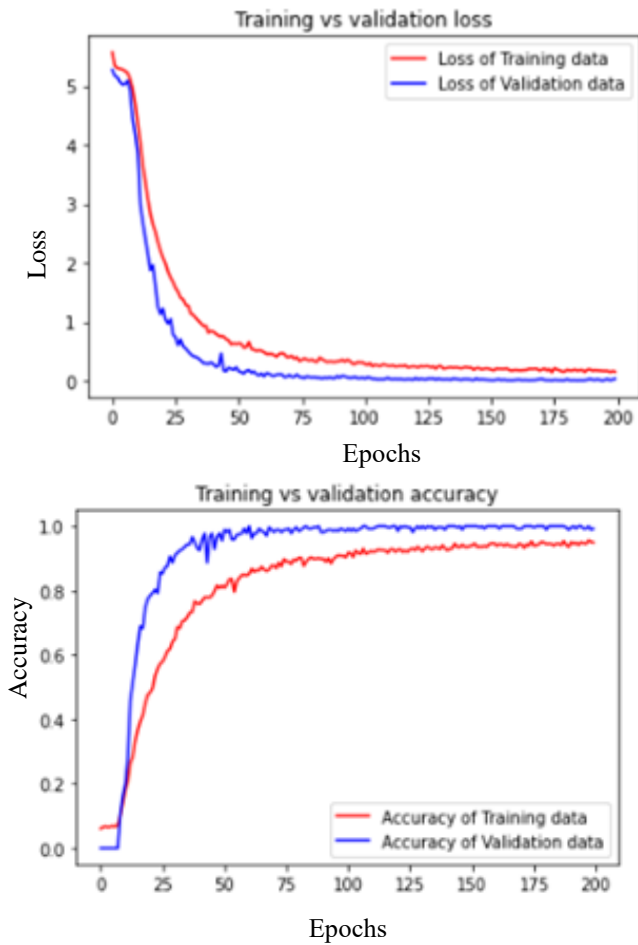


Fig. 13(a). accuracy and loss of the model that used OpenCV detector

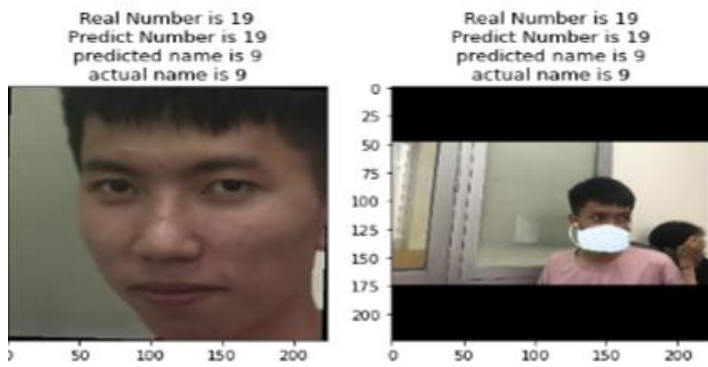


Fig. 13(b). sample of predicted image in OpenCV model

- *SSD face detection*

In Fig. 14(a), we have observed that the model is performing well on both the training and testing data, but it seems that the model is working better on the test data, we can see the validation loss (blue curve) is closer to zero than the loss of training data (red curve) so the validation loss of the test data is smaller than the loss of training data and this is due to size of the test data which is too small compared to the train data.

In Fig. 14(b). Sample of the predicted images are shown, and we can observe that the real number and the predicted number is identical, so it is matched

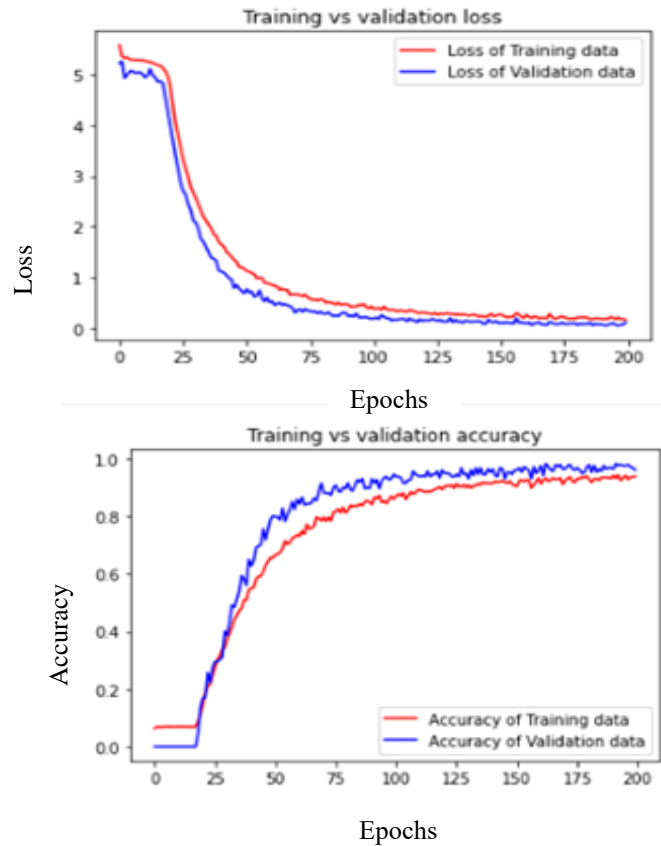


Fig. 14(a). accuracy and loss of the model that used SSD detector

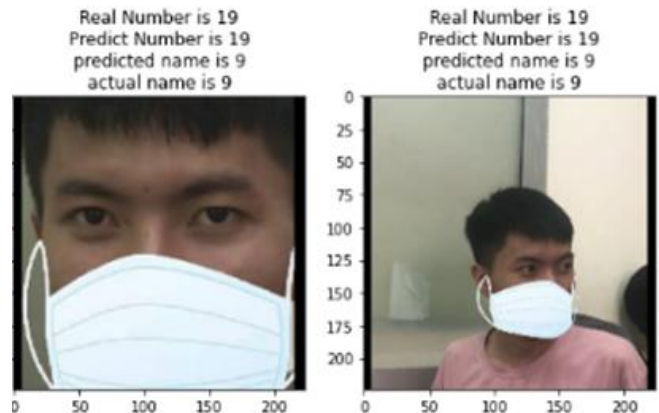


Fig. 14(b) sample of predicted image in SSD model

- *Retina Face detection*

In Fig. 15(a), we have observed that the model is performing well on both the training and testing data, but it seems that the model is working better on the test data, we can see the validation loss (blue curve) is closer to zero than the loss of training data (red curve) so the validation loss of the test data is smaller than the loss of training data and this is due to size



of the test data which is too small compared to the train data. In Fig. 15(b). Sample of the predicted images are shown, and we can observe that the real number and the predicted number is identical, so it is matched

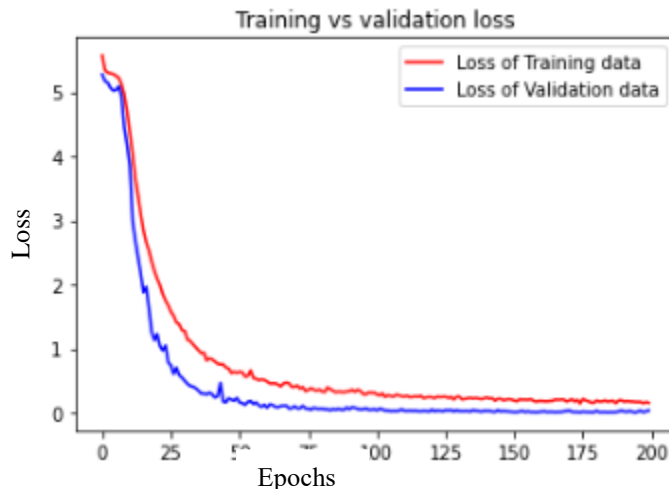
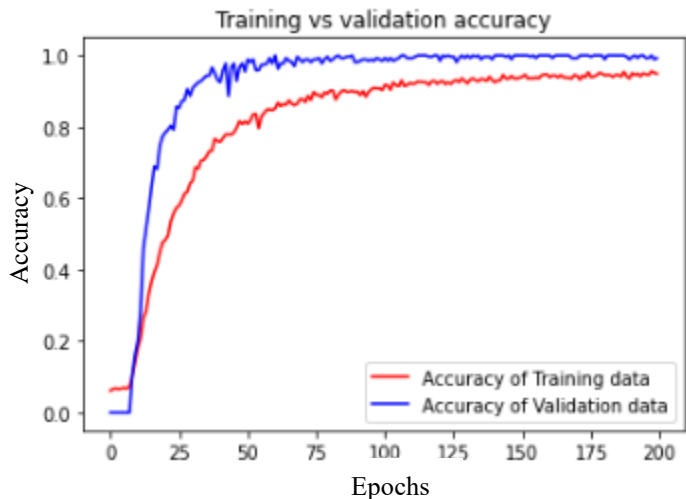


Fig. 15(a). accuracy and loss of the model that used RetinaFace detector

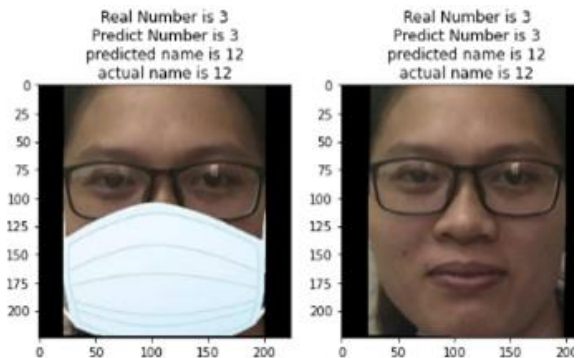


Fig. 15(b). sample of predicted image in RetinaFace model

- *MTCNN face detection*

In Fig. 16(a). we have observed that the model is performing well on both the training and testing data, but it seems that the

model is working better on the test data, we can see the validation loss (blue curve) is closer to zero than the loss of training data (red curve) so the validation loss of the test data is smaller than the loss of training data and this is due to size of the test data which is too small compared to the train data. In Fig. 16(b). Sample of the predicted images are shown, and we can observe that the real number and the predicted number is identical, so it is matched

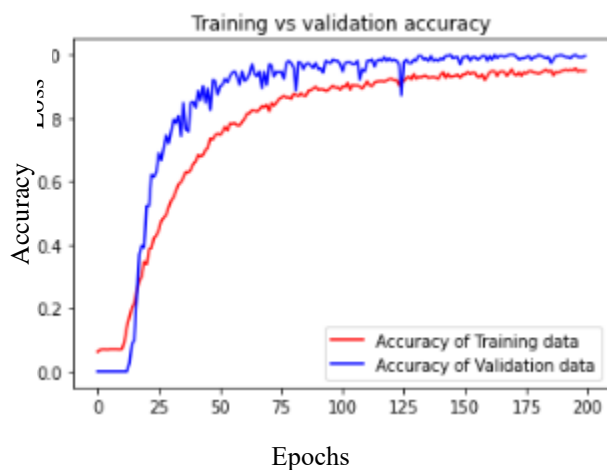
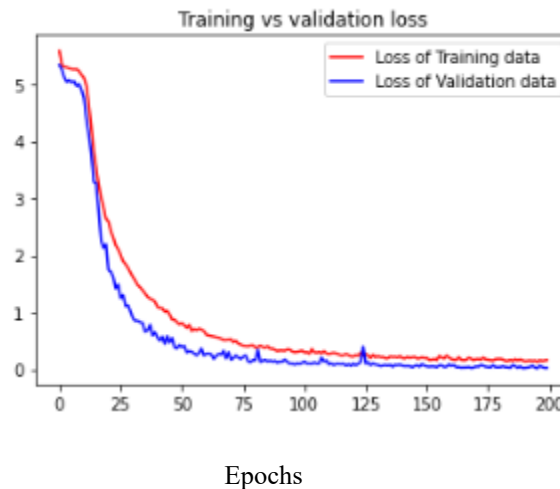


Fig. 16(a) accuracy and loss of the model that used MTCNN detector

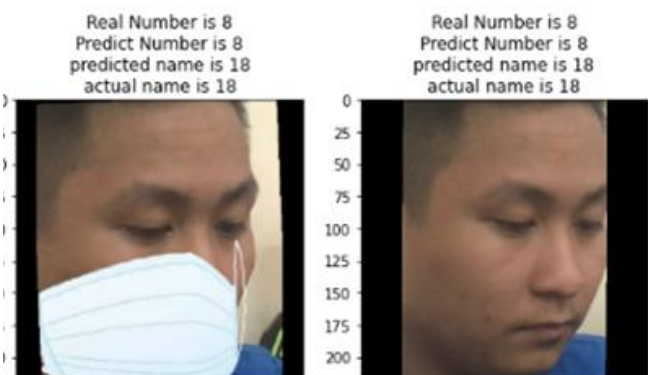






Fig. 16(b). sample of predicted image in MTCNN model

From the above graphs of each face detection classifier, we can notice that two classifiers have the highest accuracy (MTCNN, Retina Face), these two classifiers based on CNN, therefore they have high power consumption. See the table of comparison between the four classifiers in terms of prediction test image. TABLE 2.

TABLE 2 result of comparison between models of the second approach

Tested image	Face detection classifier	Detected ID	Actual ID	Match/not match	Accuracy
	Open CV	13	13	Matched	0.9924
	Retina Face	12	12	Matched	0.9962
	SSD	9	9	Matched	0.9621
	MTCNN	18	18	Matched	0.9962

### B. Final result

In TABLE 3, we showed the result of the proposed model on the three datasets and we can observe that our approach outperforms better than the first approach due to the face detection classifier that is used in the first approach failed to detect occluded faces while the classifiers in the second approach most the classifiers proved their ability of detecting occluded faces

TABLE 3 results on the three datasets

Datasets	accuracy	Over fitting/fitting /under fitting
COMASK 20	0.9962	fitting
LFW	0.9938	fitting
SoF	0.9883	fitting

## VI. CONCLUSION AND FUTURE WORK

In this work, we proposed four approaches to solve occluded facial recognition problem. From our detailed literature review, we observed that CNN approach has gained popularity for the facial recognition problem in the recent past [1,2,4,5]. We also found that the existing facial classifiers have varying success across multiple articles [1,2]. we used four face

detection classifiers and evaluated them by using COMASK20 as a dataset.

We developed a CNN architecture for the facial recognition system. In this, we used a total of 15 layers, four of them being convolution layers, one flattens layer, five dropout layers, two maxpooling layers and three fully connected layers. We evaluated this approach by using three datasets—LFW, COMASK 20 and SOF. The observed accuracy for LFW dataset was 99.38%, for COMASK 20 dataset 99.62% and for SOF dataset 98.33%.

In the future, we will further enhance our work by considering more diverse datasets consisting of subjects belonging to multiple races and multiple ethnicities. Currently we have only considered limited occlusion, primarily consisting of scarfs, eyewear, masks, etc. We would like to explore facial recognition problem in the event of increased occlusion rate. We can further consider collecting a new dataset consisting of Muslim women subjects who use veil to cover their faces.

## REFERENCES

- [1] WU, Gui; TAO, Jun; XU, Xun. Occluded face recognition based on the deep learning. In: 2019 Chinese Control And Decision Conference (CCDC). IEEE, 2019. p. 793-797
- [2] ] YANG, Lei, et al. Deep representation for partially occluded face verification. EURASIP Journal on Image and Video Processing, 2018, 2018.1: 1-10.
- [3] BASHBAGHI, Saman, et al. Deep learning architectures for face recognition in video surveillance. In: Deep Learning in Object Detection and Recognition. Springer, Singapore, 2019. p. 133-154.
- [4] VU, Hoai Nam; NGUYEN, Mai Huong; PHAM, Cuong. Masked face recognition with convolutional neural networks and local binary patterns. Applied Intelligence, 2021, 1-16.
- [5] SYAFEEZA, A. R., et al. Convolutional neural network for face recognition with pose and illumination variation. International Journal of Engineering & Technology, 2014, 6.1: 0975-4024.
- [6] HUANG, Gary B., et al. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition. 2008.
- [7] AFIF, Mahmoud; ABDELHAMED, Abdelrahman. AFIF4: deep gender classification based on adaboost-based fusion of isolated facial features and foggy faces. Journal of Visual Communication and Image Representation, 2019, 62: 77-86.
- [8] MAO, Li; SHENG, Fusheng; ZHANG, Tao. Face occlusion recognition with deep learning in security framework for the IoT. IEEE Access, 2019, 7: 174531-174540.
- [9] PARKHI, Omkar M.; VEDALDI, Andrea; ZISSERMAN, Andrew. Deep face recognition. 2015.
- [10] LAL, Madan, et al. Study of face recognition techniques: A survey. International Journal of Advanced Computer Science and Applications, 2018, 9.6: 42-49.
- [11] TARRÉS, Francesc; RAMA, Antonio; TORRES, L. A novel method for face recognition under partial occlusion or facial expression variations. In: Proc. 47th Int'l Symp. ELMAR. 2005. p. 163-166.
- [12] <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>
- [13] <https://towardsdatascience.com/image-similarity-using-triplet-loss-3744c0f67973>
- [14] VIOLA, Paul; JONES, Michael. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Ieee, 2001. p. I-I.
- [15] CHEN, Bor-Chun; CHEN, Chu-Song; HSU, Winston H. Cross-age reference coding for age-invariant face recognition and retrieval.

- In: European conference on computer vision. Springer, Cham, 2014. p. 768-783.
- [16] Lior Wolf, Tal Hassner and Itay Maoz, Face Recognition in Unconstrained Videos with Matched Background Similarity. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011.
- [17] REBHAN, Sven; SHARIF, Waqas; EGGERT, Julian. Incremental learning in the non-negative matrix factorization. In: International Conference on Neural Information Processing. Springer, Berlin, Heidelberg, 2008. p. 960-969.
- [18] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," Image and Vision Computing J, Vol. 16, No. 5, pp. 295-306, 1998.
- [19] <https://mblor.com/includes/mlai/index.html>
- [20] <https://inblog.in/CONVOLUTIONAL-NEURAL-NETWORK-N5aCeQGS5U>
- [21] MARTINEZ, A.; BENAVENTE, R. The AR face database, CVC. Copyright of Informatica (03505596), 1998.