Develop Face Mask detection algorithm for COVID-19

Carlo Provenzani 06442416 **Mohammad Pournazeri** 06487506 **Megala Anandakumar** 06409804

1.0 Abstract

Few algorithms for face detection, localization and classification are proposed in this study. The idea is to find people who wear masks and those who don't. Two different classifiers, MLP and CNN, are initially used to only classify the cropped images of faces into two classes of "With Mask" and "Without Mask". Then, a baseline model is proposed to detect and localize face(s) in an arbitrary image. The model is developed based on SVM classifier with input data in the form of histogram of oriented gradient (HOG). A sliding window approach is then used to localize the face in the image. A more advanced model is then developed for face detection, localization and classification (mask/no mask). This model is designed based on R-CNN approach. At the end of this study, an existing trained face detection and localization model (Haar Cascade Classifier) is combined with the designed CNN mask/no-mask classifier to detect, localize and classify the face.

2.0 Introduction

During the pandemic, one of the main challenges of the business is to improve their processes to be more efficient in providing a safe and healthy environment for their employees and customers. One of the most stringent protocols for public places is to assure everyone cover their faces with masks. In a crowded environment like shopping malls and supermarkets it is extremely difficult and time consuming for the businesses to find the customers or employees violating the health and safety protocols. Thus, having a camera with face detection algorithm which can detect and highlight the individuals who are not wearing mask could be a big relief.

In this study, three different issues are addressed:

i. Classification for single face to detect if the face is covered with mask or not
ii. Detection, localization and classification (mask and no mask) of a single face located in an image
iii. Detection, localization and classification (mask and no mask) of multiple faces located in an image

Detecting faces with masks is a challenging task due to followings

- lack of large dataset of faces with mask that contains face bounding box coordinates labels
- type of facial occlusions such as masks, scarf, cloths, beard, mustache or even hand

3.0 Related Works

Previous study in the area of machine vision, facial recognition and mask detection involves the application of various image classification techniques. The issue of facial mask recognition is a quite new subject. However, there have already been many attempts to create an automatic tool. In 2017 Shiming Ge et. al. [1] focused their attention on recognition of faces with occlusion (absence of facial cues from masked face regions). There are also commercial tools (proposed as pre trained model) recently made available for face mask detection applications for COVID-19. Ejaz et al (2019) [2] attempted Principal Component Analysis (PCA) to recognize person with mask and person without mask. They found that the accuracy of PCA is affected by wearing mask and recognition accuracy dropped to less than 70% when the face is masked. Muhammad et al. (2019) developed an interactive MRGAN model. They have developed Generative Adversarial Network and the methods depends on getting the microphone area from the user and regenerative that area. Kulkarni et al. (2020) [3] developed a computer program that could look at images and detect whether people are wearing masks. Yan (2020) developed pretrained models and reference applications that with transfer learning and pruning can be used for multiple applications, including face mask detection. Wang and Fei (2020) developed YOLOv3 algorithm for face detection. YOLOv3 uses Darknet-53 as the backbone.

4.0 Model Dataset

In this project, the MAFA dataset created in [1] and [4] is used for masked faces. The dataset includes around 27000 of images. The data is accompanied with a label file which includes the facial landmarks such as location of bounding box for each face in the image and three types of occlusions (simple, complex and human body). For the faces without mask, two sets of datasets are used: UTKFace [5] and FDDB [6]. The UTKFace [5] is a large dataset with a wide age span from 0 to 116 years old. This dataset includes around 23000 of images of people with different age, gender, race. The main problem with this dataset is that all the pictures are cropped, aligned and centered. This results in lack of algorithm training for face localization. The dataset from FDDB includes 2,845 images of single or multiple persons at different coordinates and with different face extents and orientations. The dataset is accompanied with a label file which includes the facial landmarks such as location of bounding box for each face in the image. In order to use the image datasets for training, the image data and their labels from different data sources should be cleaned up and grouped together with proper labels.

5.0 Proposed Methods

5.1 Single Face Mask/No-mask Classification Model

Total size of the data considered for the single facial recognition models (MLP and CNN) is 49,410. The given data is balanced one where there are 24,705 images with mask and 24,705 images without mask. The data with no mask is downloaded from [5] , The sizes of the images are not same, so the images are resized to 200 x 200. The 70% (34,587 images) of the random samples are used for training the models and the remaining 30% (14,823 images) of the data is used as a test set.

5.1.1 MLP Model

Multilayer Perceptron (MLP) Neural Network is a type of fully connected feedforward network. MLP architecture consists of at least three layers: an input layer, a hidden layer and an output layer. Input layer represents the input features that goes into the model. The nodes in the hidden layer uses non-linear activation function that maps the input features into output unit. MLP uses backpropagation algorithm while training the network.

The architecture of the model consists of one input layer, one hidden and one output layer. The ReLu activation function is used in the hidden layer, and the sigmoid function is used in in the output layer. The loss function considered in the model is binary_crossentropy, and Stochastic gradient descent (SGD) optimizer is used with batch size of 32.

5.1.2 CNN Model

CNN comprises of convolutional and pooling layers which are stacked up on each other creating a complex model. Figure 2.1 illustrates the CNN algorithm. When an image is fed into the model it flows through the convolutional and pooling layers and the last connected layers outputs the label of the class. One of the limitations with CNN's is that the model performance drops drastically with smaller size of dataset. CNN utilizes local spatial coherence which helps to get required details for low computational cost.

We used Python 3 and Keres to implement the CNN model. The architecture of the model consists of one input layer, three convolutional layers and one output layer. The *relu* activation function is used in the convolutional layers, and the *sigmoid* function is used in in the final dense layer. The loss function considered in the model is *binary_crossentropy*, and Stochastic gradient descent (SGD) optimizer is used with batch size of 32. Total number of epochs in the setting is 10.

5.2 Baseline Face detection and Localization Model

In this section, a baseline model is proposed for face detection and localization in an arbitrary image. The proposed baseline model consists of following steps:

<u>*Model Training Stage:*</u>

i. The total number of 16782 training images (including 5594 with face and 11188 without face) are all resized to size (62x62 pixels) and converted into grey format (no channel)
ii. Histogram of Oriented Gradients (HOG) vector of every training images is calculated [7]
iii. The HOG vectors with their labels (1:face / 0:no face) are fed into a Support Vector Model with linear kernel. For the linear Support Vector Machine, the SVM package from scickit-learn is used [8].
iv. The linear SVM model is trained and saved

<u>*Face Detection and Localization Stage:*</u>

i. Rescale the test image to be at least 6 times bigger than 62x62
ii. Select an sliding window with size 62x62
iii. Start from top left corner of the image and slide the window by 2 pixels at every iterations
iv. At every location of the sliding window, take a snapshot of the image confined by the sliding window
v. Calculate HOG of the snapshot and predict the image class and its prediction probability using the trained model
vi. If the class is 1, save the location of the sliding window center point ($x^{(i)}, y^{(i)}$)
vii. Repeat steps (i) to (vi) until the whole image is covered by the sliding window
viii. Use k-means unsupervised algorithm to calculate $\boldsymbol{k}$ different cluster means ($\mu_1, \dots, \mu_k$) for all centroids ($x^{(i)}, y^{(i)}$)
ix. For every combination of two cluster mean, calculate the Intersection over Union (IoU) assuming that the cluster mean is a center point of a bounding box with width and height of X and Y
x. If two bounding boxes have IoU greater than 0.5, delete the eliminate the bounding box (or cluster mean) which has the lower average prediction probability
xi. Iterate step(x) until there is no combination with IoU >0.5 is found

k_means algorithm for face bounding box clustering:

i) Assume k cluster centroids (μ_j) for maximum k faces in the image

ii) For every identified bounding box center point $(x^{(i)}, y^{(i)})$ identify the j at which the distance between the center point $(x^{(i)}, y^{(i)})$ and cluster centroid (μ_j) is the minimized: $j(i) = \underset{j}{\operatorname{argmin}} \left\| \left(x^{(i)}, y^{(i)}\right) - \mu_j \right\|$

iii) For every j, update μ_j coordinate as follow: $\mu_j = \sum_{i=1}^{n} \{j(i) = j\}\left(x^{(i)}, y^{(i)}\right) / \sum_{i=1}^{n} \{j(i) = j\}$

iv) Iterate step (ii) and (iii) until converges

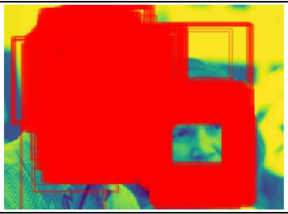
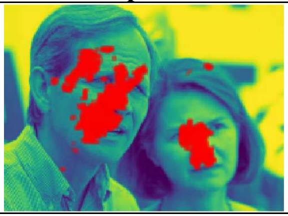
Original Image	Predicted Bounding Boxes	Bounding Boxed center points
		

Figure 1: k-means algorithm to predict clusters centroid

5.3 Region based Convolutional Neural Network Model (R-CNN)

In this section, a more advance model is proposed for face detection (or classification) and localization in an arbitrary image. R-CNN initially uses Selective Search to extract a small quantity of region proposals from an image, which can then be processed by deep learning to classify those regions of the image.

Selective search identifies patterns in an image like size similarity, color similarity, and texture similarity. It first takes an image as input and generates initial sub-segmentations so that we have multiple regions from this image, it then combines the similar regions to form a larger ones.

Model Training Stage:

i) Perform Selective Search on the training image to calculate all the possible bounding boxes

ii) For every proposed bounding box, calculate the IoU between this bounding box and true bounding box of the image obtained from image label

iii) If IoU > 0.6 and if the image is classified as face with mask, then the image section surrounded by the bounding box proposed by Selective Search is resized and added to the new masked face training list

iv) If IoU > 0.6 and the image is classified as face without mask, data is then added to the new masked face training list

v) If IoU < 0.3, then the image section surrounded by the bounding box proposed by Selective Search is resized and added to the new negative training data list

Model Prediction Stage:

i) Apply Selective Search on the test image

ii) Take the snapshot of the image section that is confined by the bounding box

iii) Resize the selected image section to be aligned with CNN model input

iv) Pass the input data through the trained CNN model and predict the image section class

v) If the predicted class is face with mask, then append the bounding box coordinate to the candidate list (same for faces with mask)

vi) Iterate steps((ii) to (v) until all the proposed bounding boxes from Selective Search are processed

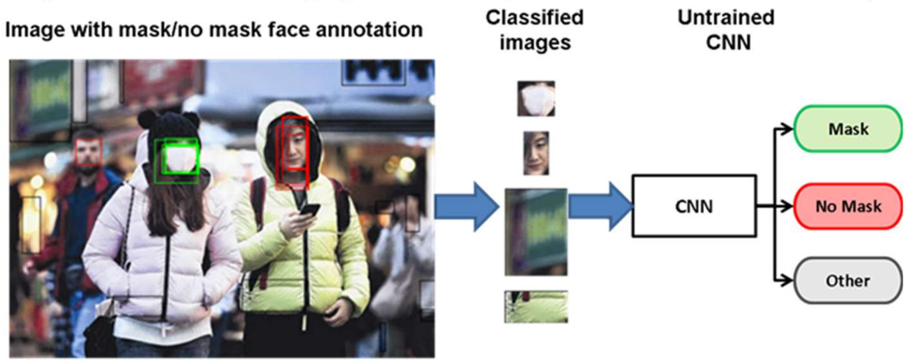


Figure 2: R-CNN Model Face Localization Procedure

Selective Search:
Selective search is an algorithm that propose image regions that may potentially contain an object of interest [9]. This algorithm is built on top of the image segmentation which works based on grouping neighbour pixels with similar colors.

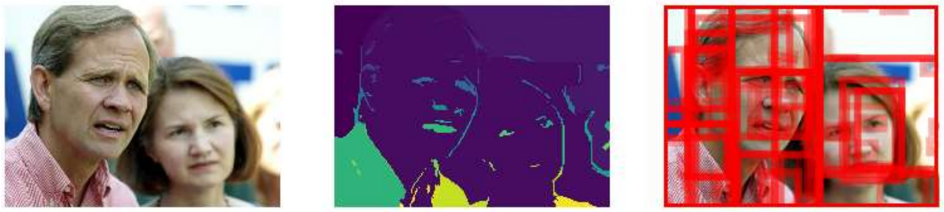


Figure 3: Image segmentation (Middle) and Selective Search (Right)

CNN Model Architecture: MobileNetV2 architecture [10] has been chosen for CNN mask detection because it uses depth-wise separable convolutions, as oppose to the standard convolution, and allows building lightweight deep neural networks that can have low latency for mobile and embedded devices.
The training has been accomplished using 852 images containing multiple faces (notated with position and dimensions) and 3218 individual face images (1962 without mask and 1256 with mask). After the Selective Search process 39,077 images were catalogued (2730 faces with mask, 2267 faces without mask and 34080 images with no face). 85% of this dataset (random) was used to run MobilnetV2 for learning (the process lasted about 2 hours using a commercial laptop PC). The remaining 15% was then used as validation data.

5.4 Facemask detection Using HAAR Cascade classifier

As an alternative to Selective Search, to correctly detect and identify faces, HAAR Cascade Classifier (OpenCV) has been used. It is an algorithm that needs a lot of positive images of faces and negative images without faces to train the classifier. It works by extracting and considering adjacent rectangular regions at a specific location in a detection window (HAAR features), summing up the pixel intensities in each region and calculating the difference between these sums. Once region proposals have been determined by HARR, a classifier can be run for each proposed region to classify if the proposed region is a face with mask or a face without mask.

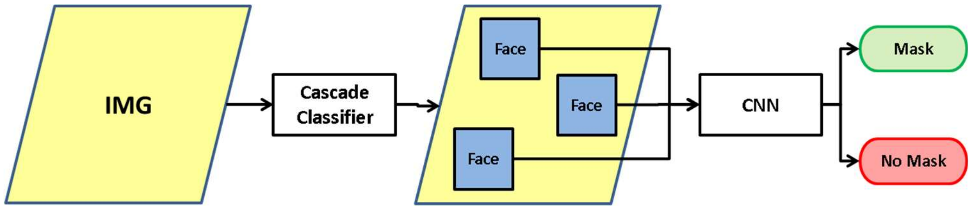


Figure 4: Proposed Face with/without Mask Detection and Localization algorithm using Haar Cascade Classifier

6.0 Experiments Results and Discussion

Figure 5 and Figure 6 show the MLP and CNN model accuracy with respect to training and test data. As shown in the confusion matrices, both models have pretty high classification (mask/ no mask) accuracy (> 93 %) for the test dataset.

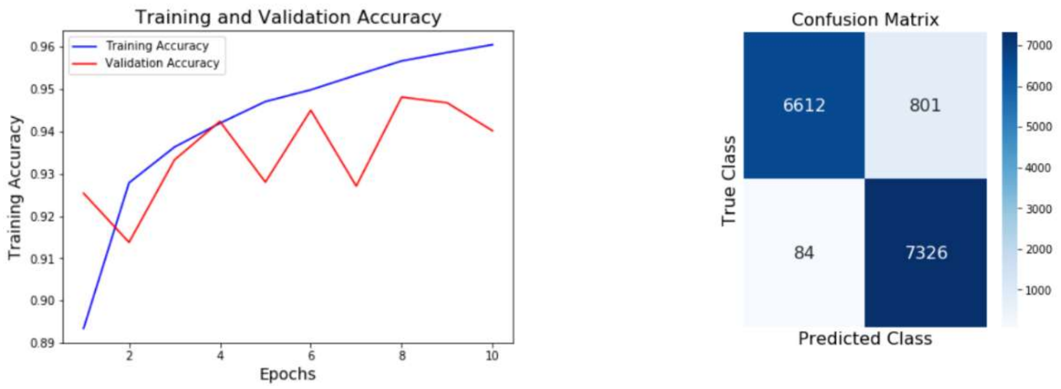


Figure 5: Proposed MLP model accuracy and performance with respect to training and test data

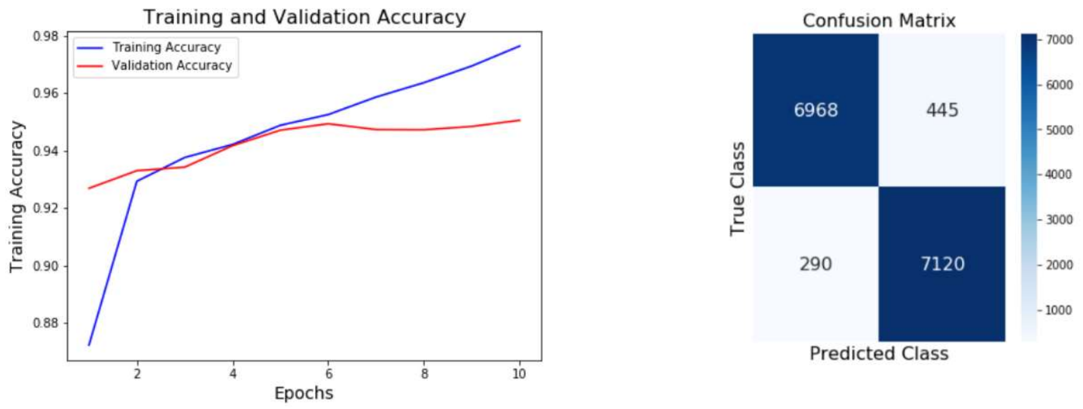


Figure 6: Proposed CNN model accuracy and performance with respect to training and test data

The proposed baseline model (SVM+Sliding Window+*k*-means) was trained and tested on several arbitrary images to predict face bounding box. Figure 7 shows the performance of the model prediction for different images. As shown, in some of these images, the predicted bounding box is sometime missing or at wrong location; however, in average the results are very promising.

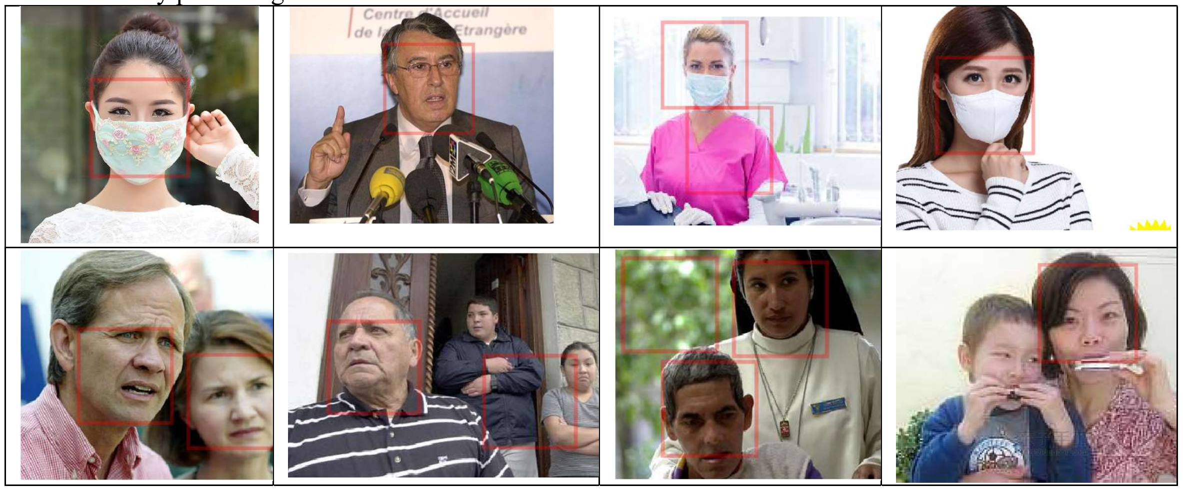


Figure 7: Face detection and localization using the proposed baseline model with train and test performances of around 95% and 88 %.

Figure 8 shows the convergence performance of the CNN model designed for R-CNN algorithm using MobileNet architecture. The trained model has accuracy of over 99% at both training and validation dataset.

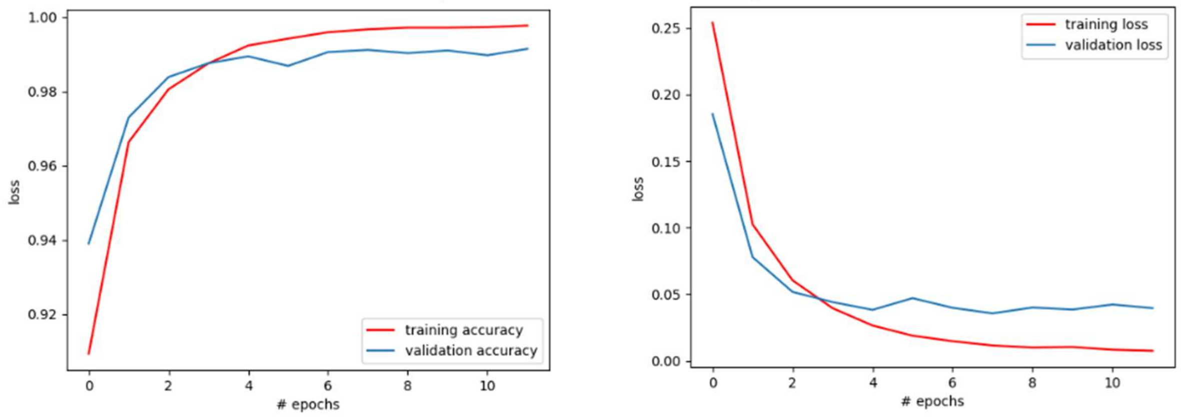


Figure 8: Convergence performance of CNN model for R-CNN algorithm for face detection and classification

Figure 9 shows the performance of the designed R-CNN model for detection, classification and localization. The results show less false positive and false negative predictions compared to the baseline model. Also, the model not only detect and localize the face in the image, but also classify the face as with mask or without mask. Figure 10 shows a results for same set of images from Haar cascade classifier. The results show very high accuracy compared to SVM and R-CNN models . The reason is that the face detection model (Haar Cascade classifier) was already trained with much larger amount of training data.

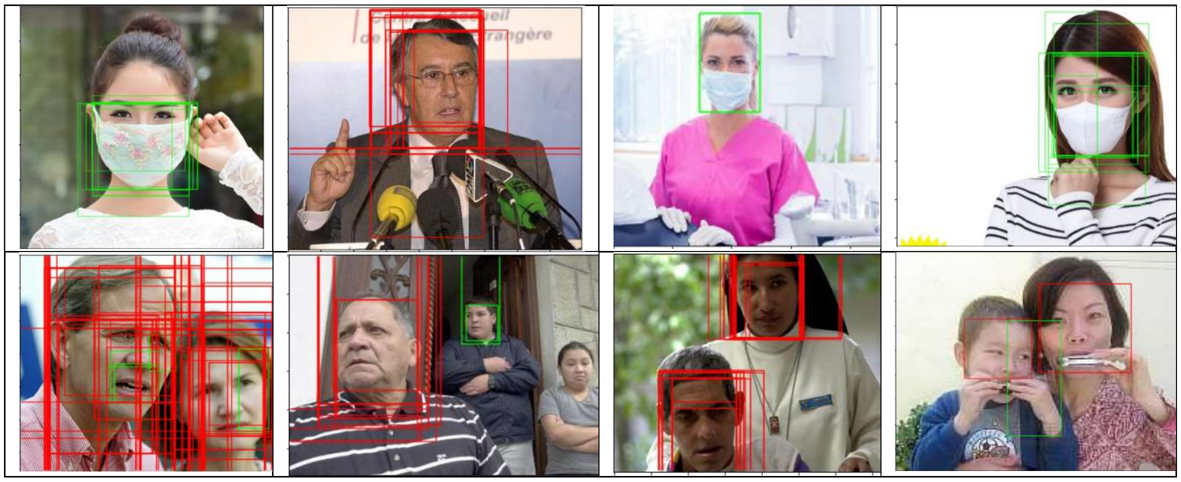


Figure 9: Face detection, classification and localization using the proposed R-CNN (**green**: mask / **red**: no mask)

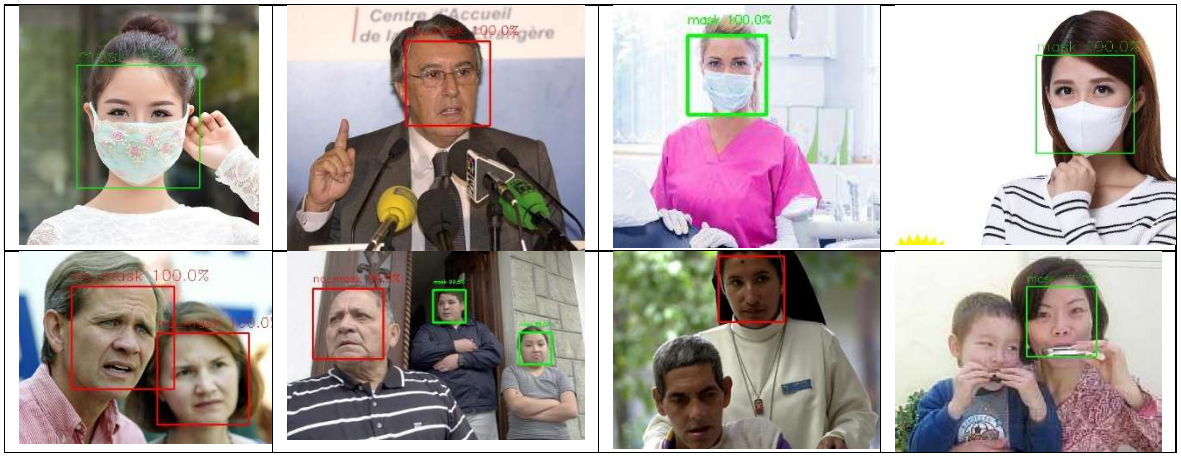


Figure 10: Face detection, classification and localization using the HAAR Cascade classifier with CNN classifier

7.0 Summary, Conclusion and Future Work

Three models of SVM with sliding window, R-CNN and Haar Cascade Classifier were developed and tested for face detection, localization and mask/no mask classification. The R-CNN model shows more accurate prediction with respect to the baseline model due to the more complex CNN classifier and also selective search instead of sliding window. The Haar cascade classifier showed superb face detection and localization. This also helped with mask/no mask classification due to more accurate location of the bounding box which results in better input to CNN classifier. In order for this research work to be valuable, it need to be extended to real-time video images.

References

1. Shiming Ge J. L. Q. Y. Z. L, "Detecting Masked Faces n the Wild with LLE-CNNs," in Computer Vision Foundation. (https://openaccess.thecvf.com/content_cvpr_2017/papers/Ge_Detecting_Masked_Faces_CVPR_2017_paper.pdf
2. Ligang Zhang D. W. T. V. C., "Facial Expression Analysis under Partial Occlusion: A Survey," in ACM Computing Surveys, 2018.

3. "Kaggle," [Online]. Available: https://www.kaggle.com/rahulmangalampalli/mafa-data/version/1.
4. Z. Z. Yang Song, "UTKFACE - Large Scale Face Dataset," [Online]. Available: https://susanqq.github.io/UTKFace/.
5. Chowdhury A.R. "FDDB: Face Detection Data Set and Benchmark," University of Masachusetts Amherst , [Online]. Available: http://vis-www.cs.umass.edu/fddb/.
6. Yan W, "Face-mask recognition has arrived—for better or worse, September", 11, 2020. (https://www.nationalgeographic.com/science/2020/09/face-mask-recognition-has-arrived-for-coronavirus-better-or-worse-cvd/).
7. Kulkarni A, Vishwanath A and Shah C "Implementing a Real-time, AI-Based, Face Mask Detector Application for COVID-19". August 17, 2020 . (https://developer.nvidia.com/blog/implementing-a-real-time-ai-based-face-mask-detector-application-for-covid-19/).
8. Adnane Cabani A, Hammoudi K, Benhabiles H, Melkemi M, "Maskedface-Net – A Dataset of Correctly/Incorrectly Masked Face Images In the Context of COVID-19", August 18, 2020.
9. M.K.J. Khan, N. Ud Din, S. Bae, J. Yi, Interactive removal of microphone object in facial images, Electronics 8 (10) (2019) , Art. no. 10, doi: 10.3390/ electronics8101115.
10. Shaik et al [16] used deep learning real-time face emotion classification and recognition. They used VGG-16 to classify seven facial expressions. The proposed model was trained on the KDEF dataset and achieved 88% accuracy.
11. S. A. Hussain, A.S.A.A. Balushi, A real time face emotion classification and recognition using deep learning model, J. Phys.: Conf. Ser. 1432 (2020) 012087, doi: 10.1088/1742-6596/1432/1/012087.
12. M.S. Ejaz, M.R. Islam, M. Sifatullah, A. Sarker, Implementation of principal component analysis on masked and non-masked face recognition, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–5, https://doi.org/10.1109/ ICASERT.2019.8934543.
13. C. Li, R. Wang, J. Li, L. Fei, Face detection based on YOLOv3, in:: Recent Trends in Intelligent Computing, Communication and Devices, Singapore, 2020, pp. 277–284, doi: 10.1007/978-981-13-9406-5_34.
14. N. Ud Din, K. Javed, S. Bae, J. Yi, A novel GAN-based network for unmasking of masked face, IEEE Access 8 (2020) 44276–44287, https://doi.org/10.1109/ ACCESS.2020.2977386.
15. A. Nieto-Rodríguez, M. Mucientes, V.M. Brea, System for medical mask detection in the operating room through facial attributes, Pattern Recogn. Image Anal. Cham (2015) 138–145, https://doi.org/10.1007/978-3-319-19390-8_16.
16. https://cezannec.github.io/Convolutional_Neural_Networks