Face Recognition from Masked Images and an Attendance System Implementation using Convolutional Neural Network and LBP Histogram with HAAR classifiers

Jibin P Reji[1], Gerardine Immaculate Mary[2],
[1]School of Electrical Engineering, [2]School of Electronics Engineering, Vellore Institute of Technology,Vellore, Tamilnadu, India
jibinpreji@gmail.com, gerardine@vit.ac.in

ABSTRACT
With the outbreak of the COVID 19, the pandemic has prodded a vital improvement in the field of touch-less biometric scanning that can also examine a person is wearing a facial mask or not. The development of touch-less bio-metric technologies has gained importance due to COVID-19 hygiene requirements. Mask is a key element to avoid corona disease, but it affects the face recognition algorithms which loses the facial features when the face is covered by the mask. This paper proposes a method to recognize the face of a mask-wearing person using local binary pattern histogram (LBPH) algorithm from digital images. Also, develops a system that can detect instances where the face mask is on or not with the help of computer vision and convolutional neural network (CNN) architecture. Furthermore, this paper demonstrates a method of developing an attendance system using facial recognition. The proposed combination of face recognition and mask detection models is computationally efficient and easy to deploy on any embedded system (Raspberry Pi, Nano, Google Coral, Jetson, etc.) thus this paper showcases the proposed model implementation on Raspberry Pi 4 using an external web camera on live video and the result obtained is found to be satisfactory as a touch-less systems for validating the person's biometric using face recognition and also checks whether the person wears the mask or not. These two factors of touch-less biometric along with confirmation of wearing face mask are mandatory requirements during pandemic periods like COVID-19.

Keywords: Attendance system, Convolutional neural network (CNN), Face Recognition, Local binary pattern histogram (LBPH), Mask Detection, HAAR classifiers, Raspberry Pi

1. INTRODUCTION

Video surveillance frameworks are introduced for screening the people to enter a building like malls and theatres. Today, with the development of science and technology, the era of artificial intelligence and machine learning has been proposed and set off changes in varying backgrounds. Face recognition is a non-contact bio-metric identification strategy which has an application prospect in the fields of public security. Furthermore, checking an individual's identity with a mask has become an exploration in the area of interest. There is an urgent need for touch-less bio-metric innovations because of the crisis of COVID19. Commonly used methods like ID cards and fingerprints have been neglected due to hygiene considerations. These conventional methods are easy to cause time delay, waste of material and human resources. So, this paper depicts simple and cost-effective equipment for mask detection and face recognition with the help of deep learning algorithms and deploying the model in an embedded processor (Raspberry Pi). In addition to the available databases, the facial recognition model is implemented to detect and identify faces in real-time through a web camera for the user-defined database. Attendance system through face recognition saves time as well as removes the chances of proxy attendance because of the face identification. This system is much more important in educational and other institutions to manage workforce productivity. The basic requirements for this implementation are Raspberry Pi, Open CV, and Dlib using python. In section 2, the related work is discussed and the uniqueness of the proposed work is highlighted. In section 3, the overview of the proposed system is presented. In section 4, the datasets used in this paper is shared. In section 5, the proposed methodology is depicted, section 6 describes the non contact attendance system and section 7 explains the implementation of the proposed methodology.

2. RELATED WORK

In the field of computer vision, face recognition is a major challenge. There are several face recognition algorithms like LBPH, eigenface and fisher face, etc. In the paper [1], which proposes training eigenfaces and recognizing the face. Using the algorithm of principal component analysis, face recognition is implemented in [2]. And several methods such as Fisherface and elastic graph matching (EGM) were researched in [3][4][5]. LBPH, Eigenface, and Fisher face calculation for face recognition on python library OpenCV is compared and presented[6]. Eigen faces and Fisher faces are affected both by light and one cannot ensure perfect light conditions in real life. An improved LBPH faces recognizer overcomes this problem. Local Binary Patterns (LBP) were introduced for the classification of texts in [8], but the proposed method was previously implemented for face recognition by T. Ahonen et al [7].

On the other hand, the object detection algorithm [9][13] is used for mask detection, To detect the facial mask, one has to initially detect the human face [10]. The maximum likelihood of face detection by tracking the facial features is proposed in [11]. False detection (no face region) is avoided by implementing a bootstrap algorithm to get positive results [12]. Face detector can be used as a Single Shot Multi-box Detector and framework as the MobilenetV2 architecture for the classifier, which is used for implementing face mask detector in embedded devices such as Raspberry pi NVIDIA Jetson Nano [14]. Deep learning architecture like the convolutional neural network is used for an automated system to detect masked people in smart cities by monitoring public places through CCTV cameras [15].The face of the person is recognized by using LBPH face recognizer. The system determines the persons from the trained images are present or not. The tested attendance names are stored in an excel sheet which is automatically updated in the system [16].

[21] proposes a model which records the attendance of an employee using RFID and facial recognition along with a temperature check, it also captures the facial expression of the employee to detect his emotion. This study of face mask detection is made possible using Deep Learning, Convolutional neural network algorithm, and MobileNetV2, a python open-source image-processing and classification model [22]. This research aims to create a face mask identification system, that has used a transfer learning method with the MobileNetV2 model to classify people who wear face masks properly, wear face masks improperly, and are without masks [23].This research proposes a face mask detection system using a machine learning algorithm known as Support Vector Machine (SVM) [24].

In all the related work discussed in this section, the combined model of face recognition and mask/object detection as well as implementation of face recognition from masked images is not tried. This paper stands out from the related work in recognizing face from masked images which is the unique contribution of this work in an appropriate timeframe like the pandemic.

3. SYSTEM OVERVIEW

3.1. Face Detection

A technique that shows the position of human faces from digital images in a rectangular box. Face recognition is distinct from face detection. Face detection determines the presence of human faces in an image, while facial recognition requires identifying the person's biometric from an image. Face detection is mainly used to locate and extract a face from an image using the face detection algorithm. Mainly used for face verification and face or non-face decision

Face detection is done by using HAAR classifiers. A classifier is an algorithm that decides whether a given image is positive (face region) or negative (non-face region). A classifier needs to be trained on a huge number of images with and without faces. OpenCV has two pre-trained classifiers, which can be used in a program promptly. The pre-trained classifiers are the HAAR Classifier for face detection and Local Binary Pattern (LBP) classifier for face recognition.

By using the DetectMultiscale module Classifier function will return a rectangle with coordinates around the detected face. For this function, two important parameters have to be tuned according to the required results.

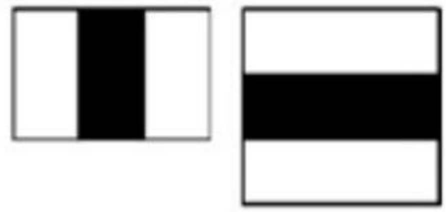
1.Scalefactor: In a group shot, there might be some faces that are closer to the camera than others. Naturally, those faces appear more prominent than the ones behind them. Scale factor compensates this reason.

2.MinNeighbors: This parameter specifies the number of neighbors in a rectangle that should be considered for a face.

3.2. HAAR Classifier

HAAR cascade classifier is created by training the huge amount of images like frontal face, number plates, eyes, profile face and it is taken into the framework. During testing, the classifier starts by extracting HAAR features from each image. HAAR features are the type of convolution kernels which recognize whether an appropriate element is available on an image or not. Figure 1 shows examples of HAAR Features.

(a)



(b)

(c)

Figure 1: Examples of HAAR Features: Square shaped kernels [17]
(a)Edge Features (b) Line Features (c) Four rectangle features

The HAAR feature is primarily a single target acquired by the difference between the sum of the white region pixels and black region pixels. The visualization of the process is shown in Figure 2.

:

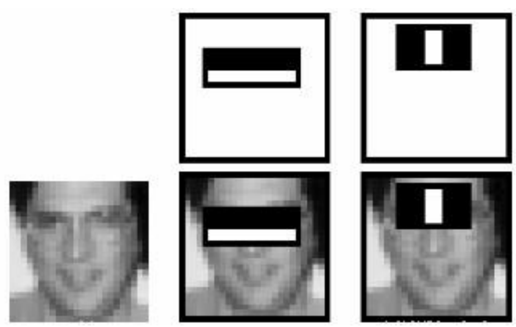


Figure 2: HAAR Features applied through the relevant parts of the face [17]

Here, the first characteristic is based on the fact that the region of the eye is darker than the area of the adjacent cheeks and nose. The second factor focuses on the fact that the eyes are darker relative to the bridge of the nose.Thus, when the feature window moves over the eyes, it will calculate a single value.Then this value is compared to a certain threshold and it means that if it passes, there is an edge or some positive features.

.

3.3. Cascade of Classifiers

Using a classifier cascade is another way, Viola Jones[9] made sure that the algorithm performs quickly. Essentially, the classifier of the cascade where each stage consists of a strong classifier, that is instead of applying a large number of features in a window, the feature are ordered and applied one by one, refer Figure 3 . If the feature fails at first stage, then discard the process. If it does not fail, the second group of features applied. Finally, the window with all stages of the face is taken.This is beneficial as it avoids the need for all features to be connected to a window at once. Instead, the features are grouped into different sub-windows at each stage and the classification is categorized.

3.4. Convolutional Neural Networks

Convolutional neural networks, or CNNs, are a special form of a neural network built with a grid-like structure to process data. Some of the first deep learning models to be demonstrated to generalize well to useful tasks. Modern interest in CNNs came when Alex Krizhevsky won the 2012 ImageNet object recognition challenge using AlexNet [19].

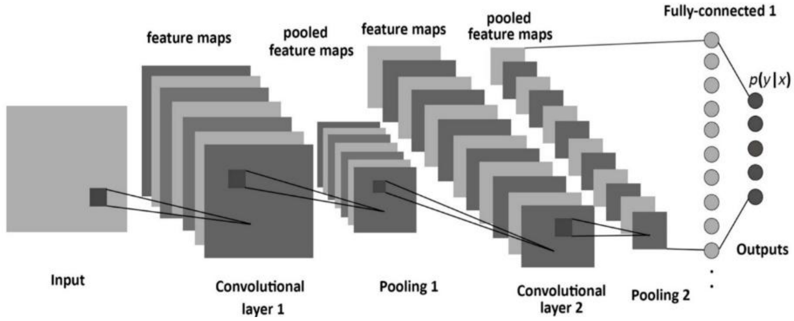


Figure 3: Convolutional Neural Network Architecture [18]

CNNs were implemented to enable neural networks to focus on images on the local characteristics or patches. It is like a detector that scans images looking for eye-like and nose-like features to detect a human. For example, features can be extracted at different local regions to predict whether a car is present in an image or not.

3.5. Local Binary Pattern Histogram (LBPH)

A very effective texture operator will label the image pixels by thresholding each pixel's area and consider the final result as a binary number. LBPH will generate an image which shows the features better. The four parameters of the LBPH algorithm are radius, neighbors, x grid, and y grid. Radius is used to create a circular binary prototype. The amount of the sample binary pattern is called neighbors Horizontally-oriented number of cells and vertically oriented number of cells is called x grid and y grid respectively. To identify with face images, the algorithm requires a dataset of individuals. For each image, it also needs to set an ID (it may be a number or the name of a person), so the algorithm will use this information to recognize an input image and give back an output. The images of the same individual must have the same ID.

4. DATASET

4.1. Mask Detection Dataset

To check automatically, whether a person is wearing a mask or not wherein the classroom or mall, or theatres, a large dataset of images is needed for the processing of images as compared to the face recognition dataset. The mask detection dataset consists of two kinds: images of a person wearing a face mask and images of a person who doesn't wear a face mask. For the construction of this model, the face mask dataset provided by Prajna Bhandary consists of approximately 1,376 images containing 690 images of individuals with face masks and 686 images containing individuals without face masks, as shown in Figure 4.

(a) (b)

Figure 4. Dataset for mask detection
(a)With Mask, (b)Without Mask

4.2. Face Recognition Dataset

The face recognition dataset doesn't need a huge amount of images of the person. But the data-set becomes bigger if it needs to recognize many people. For training, it needs approximately 20-30 images of the person saved and labeled in the corresponding folder, as shown in Figure 5.

(a) (b)

Figure 5. Dataset for face recognition
(a)Person 1(b) Person 2

5. PROPOSED METHODOLOGY

5.1. Mask Detection

This technique is done through convolutional neural network architecture. For creating the CNN algorithm in OpenCV, the lists that are generated (data and target) are converted into NumPy arrays for easier data pre-processing. The data array is reshaped so that it can be supplied to Neural Network Architecture as an input. These are then stored as a .npy format. First, it will load the data from the files that was created in the previous stage. Then, a neural network using Convolutional and MaxPooling layers is created. Finally, the output is flattened and fed into a completely connected dense layer of 50 neurons and finally into 2-neuron layers as it will give the probability if an individual is wearing a mask or not. Data and targets are then split into training and data testing, keeping 10 percent of the data for testing and 90 percent as training data.

A checkpoint that saves the model is built, which will have the least loss of validation. The training information is then built into the model so that predictions can be built in the future, the two phases and individual steps shown in Figure 6 are elaborated in this section.

Phase 1: Training face with mask detector – the steps are

i. Create face dataset (with mask and without mask)
ii. Train the classifier
iii. Serialize the dataset.

Phase 2: Apply the face mask detector

i. Load the dataset.
ii. Extract the region of interest.
iii. Apply face mask classifier.

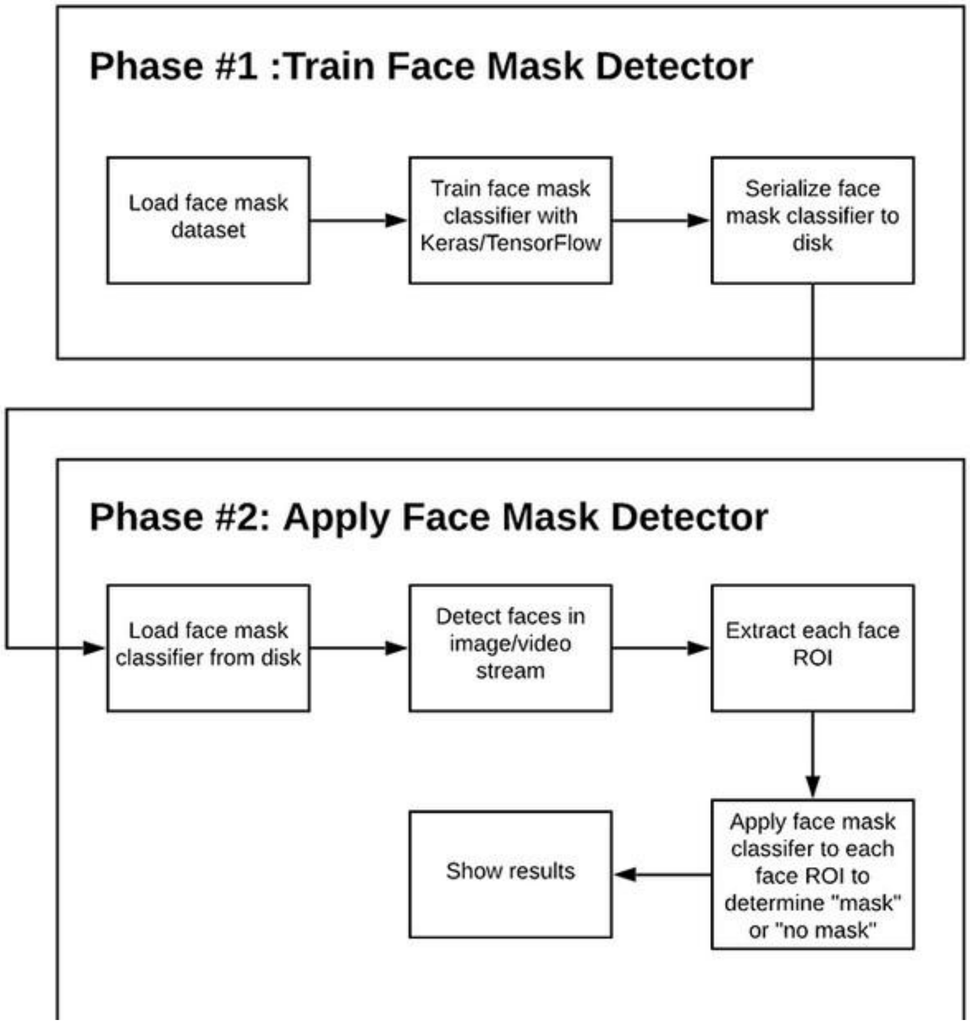


Figure 6: Phases and individual steps for using Python, OpenCV, and TensorFlow/Keras to create a face mask detector using computer vision and deep learning. [20]

5.2. Face Recognition

After training 20-30 images of persons,using LBPH recognizer, one can start to recognize the trained person or not. The algorithm's output is the imageID with the histogram closer to it. The algorithm should also return the calculated distance, which can be used as a 'confidence' calculation. One can then use a threshold and 'confidence' to estimate automatically whether the algorithm has correctly recognized the picture or not. The features of the algorithm and the algorithm are listed in this section. The flowchart of the algorithm is shown in Figure 7.

Features of LBPH recognizer:

- Easiest face recognition algorithm.
- Represent local features in the images.
- Possible to get great results (mainly in a controlled environment).
- Provided by the OpenCV library

Algorithm of LBPH recognizer:

i. Input face images of the person
ii. Divide the face images into blocks
iii. Calculate histogram for each block
iv. Face processed
v. Compare the histogram to get face recognition results

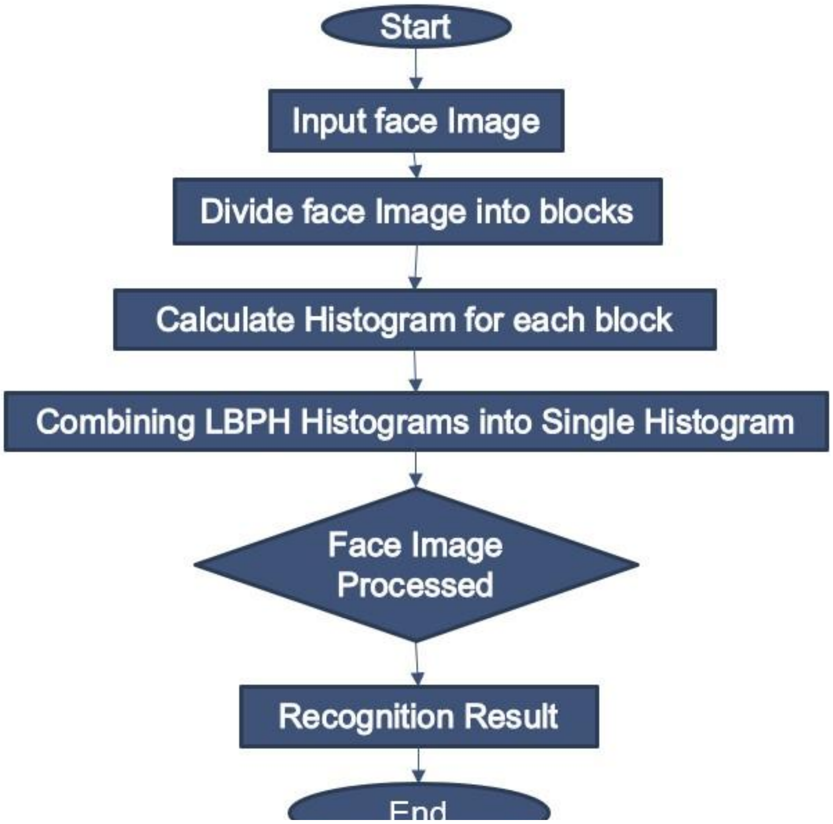


Figure 7: Flowchart of LBPH recognizer

5.3. Face Recognition and Mask Detection combined Model

For the face recognition with mask, the proposed combined model is used. The training steps and testing steps to deploy the proposed combined model is discussed in this section.

Training steps:

i. The location of the face is detected using the HAAR Cascade algorithm and the region of interest (ROI) is cropped using the cascade classifier.
ii. The region of interest (ROI) is converted into greyscale and the label is declared.
iii. The label and faces are updated by training from the required folder.The obtained details are given to the classifier function.
iv. LBPH algorithm is invoked and the face is recognized with a mask even if some facial clues are absent.

Testing steps:

i. For the face recognition with mask, one can combine above two methods: Convolution Neural Network and LBPH algorithm
ii. For that, one needs to train and save the mask detection-CNN model as .sav file and face recognition model (LBPH recognizer) as a .yml file.
iii. By calling this model one creates a new system that can filter out the details of masked images and ultimately detects the face of the person.
iv. Using the pickle library one needs to load the .sav file for mask detection.
v. Using the OpenCV library one loads the HARR cascade classifier for face detection.

vi. Now with the same library, one can load the model of face recognition (LBPH recognizer).
vii. Using the library function for camera in OpenCV, one invokes the web camera.
viii. The bounding box for face with mask is Green and without mask is Red, which is determined by color features.
ix. Now one can go to the required directory for labeling the face.
x. When the camera is turned ON it first checks for the face, if face is found it will resize the ROI and convert the image to a greyscale.

6. ATTENDANCE SYSTEM

The time and management system are incorporated to the proposed system as shown in Figure 8, to monitor the employee/student tracking. Data collection for accurate absence and leave calculation at real-time clock helps to increase the engagement and workforce productivity in various institutions.

This attendance system not only validates the person's biometric without touch but also checks whether the person wears the mask or not. The recorded attendance name is extracted to an excel sheet with the standard real-time. The traditional attendance management system recognizes the person's face only, but the proposed attendance system recognizes the person with a mask, and this is essential during situations like COVID 19 pandemic.

The proposed system is implemented by using the face recognizer and mask detector methodology discussed in section 4. Mask detector detects the mask using the mask classifier to check the person has put on or not. Mask detection results will be shown in the excel sheet even if a non-trained (unknown) person has come in front of the camera. Moreover, it also recognizes the person biometric in a touch-less manner by using LBPH algorithm.

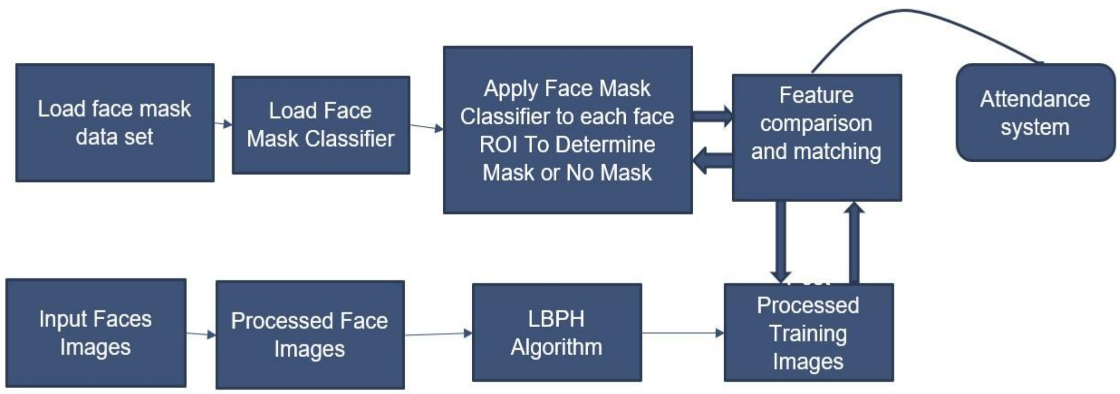


Figure 8: Block diagram of attendance system using computer vision and deep learning

7. PROPOSED MODEL IN RASPBERRY PI-4

This proposed model is implemented on Raspberry Pi powered on a web camera for the portability of the system and to check the surroundings. The system contains Raspberry Pi 4 (RAM 4GB) and a smart webcam (8 megapixel), a display monitor, micro USB adapter, HDMI cable or Ethernet cable, SDcard, keyboard and mouse, featuring face recognition, mask detection, attendance system, home automation and ability to interact with web cam's live recording or from a recorded video, as shown in Figure 9.

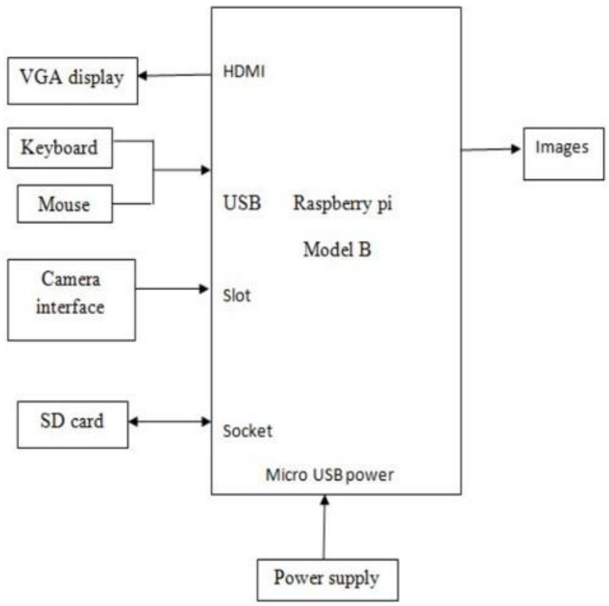


Figure 9 Block Diagram of the proposed system

The Raspberry Pi 4 based facial recognition and mask detection system requires the installation of the Raspbian operating system in 64 GB micro SD and switch on the power supply. Then the step involves installing python, OpenCV, dlib, TensorFlow libraries. After login with HDMI cable, complete the Pi installation setup wizard. The steps are; activate the camera module of the Raspberry Pi, then connect the camera module to the USB port and start the live recording and run the code for feature comparison, as shown in Figure 10.

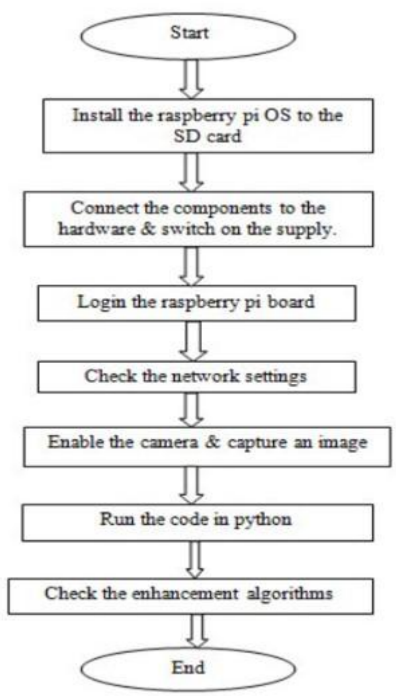


Figure 10: Flowchart of the Implementation

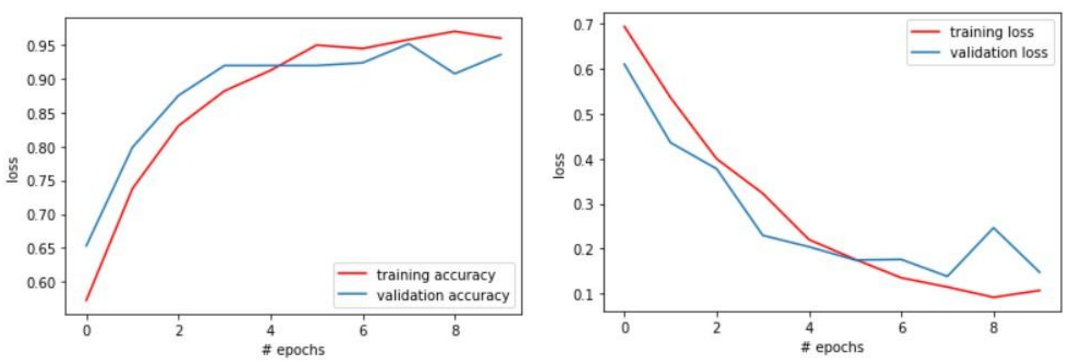
8. EXPERIMENTAL RESULTS

Mask detection model is perfectly able to detect whether a person is wearing a mask or not and is labelled as "MASK" and "NO MASK" on the bounding box as shown in Figure 11. CNN architecture is used on the mask/no mask dataset and it is a computationally good model for deploying in many embedded processors.

(a) (b)

Figure 11 Mask Detection using convolutional neural network
(a)Person with mask (b) Person without mask

Figure 12 shows the estimated graph of accuracy and loss. The training accuracy is around 96% and the validation accuracy is approximately up to 93%. Epoch vs accuracy graph shows how the optimized accuracy overcome the cost of error. Epoch vs loss graph shows the loss in training and validation, gradually decreasing after each epoch. The optimized filter values help the model to get the region of interest (face) to detect the presence of the mask accurately by avoiding overfitting.



(a) (b)

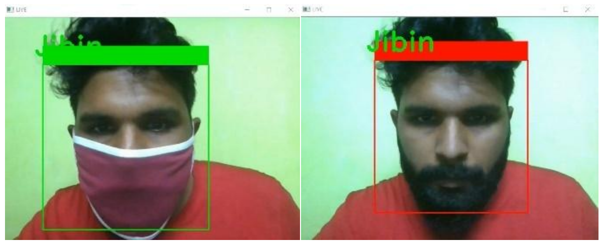
Figure 12 Accuracy and loss evaluation of dataset
(a)epochs vs accuracy (b)epochs vs loss

After feature comparison and matching with the processed training images, the result is shown in Figure 13. The algorithm worked accurately even though the person used a mask when some of the facial features (nose, mouth) are covered by the mask. Hence, it gives more probability for face recognition with masked images.

(a) (b)

Figure 13: Face Recognition Using LPBH algorithm
(a)Face recognition of the person (b) Face recognition with mask

Hence by combining the mask detection and face recognition models, a new model as mask detection and face recognition combined is obtained as a system. This proposed model recognizes the person and labels the name on the rectangular box and checks whether the person has mast or not. If the person has mask, then the box color turn green else it shows red as shown in the Figure 14.



(a) (b)

Figure 14 Combined model: Face recognition and mask detection
(a)with mask (Person's biometric in a green bounding box)
(b)without mask(Person's biometric in a red bounding box)

For the attendance management system 5 datasets of 5 different persons is used. The person's name is automatically extracted in an excel sheet. It also shows if the person wears the mask or not in real-time as depicted in Figure 15. It checks the person's biometric with the mask in a touchless manner, if a person does not wear the mask, it is clearly shown in the sheet.

	A	B	C
1	name	mask	time
2			
3	Jibin	mask	09:45:15
4	Roni	mask	09:45:51
5	Roby	no mask	09:47:01
6	Subin	mask	09:56:23
7	Navin	mask	09:59:03

Figure 15: Attendance management system excel sheet

The mask detection model, face recognition model, the combined face recognition and mask detection model are computationally efficient and easy to deploy in any embedded system (Raspberry Pi, Nano, Google Coral, Jetson, etc.).So, the proposed model is implemented in Raspberry Pi 4 and the results obtained on the screen are as shown in Figure 16.

(a) (b)

Figure 16: Proposed model implementation in Raspberry Pi
(a)biometric label with mask (b)biometric label without mask

9. CONCLUSION

The world is confronting a colossal well-being emergency on account of pandemic COVID-19. The administrations of different nations are battling to control the transmission of the Covid. Hygiene is most important to avoid the spreading of coronavirus. Thus touchless biometric is more vital for any security system. In this paper, efficient facial recognition and mask detection is proposed. Face recognition is implemented by local binary pattern histogram and with help of Convolutional Neural Network mask detection is implemented. These algorithms are developed using libraries in the python platform. An attendance management system is also developed to detect a face with mask on or not. Besides, the implemented model applied in the Rasbian operating system using Raspberry Pi 4 is also presented. Future work includes studying more advanced methods and recognizing faces in Raspberry Pi cam and train more people to check how well the algorithms work accurately.

REFERENCES

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 586-591, 1991

[2] R. Kaur and E. Himanshi, "Face recognition using Principal Component Analysis," 2015 IEEE International Advance Computing Conference (IACC), Banglore, India, 2015, pp. 585-589, doi: 10.1109/IADCC.2015.7154774.

[3] Hyung-Ji Lee, Wan-Su Lee and Jae-Ho Chung, "Face recognition using Fisherface algorithm and elastic graph matching," Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205), Thessaloniki, Greece, 2001, pp. 998-1001 vol.1, doi: 10.1109/ICIP.2001.959216.

[4] N. Jamil, S. Lqbal and N. Iqbal, "Face recognition using neural networks," Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century., Lahore, Pakistan, 2001, pp. 277-281, doi: 10.1109/INMIC.2001.995351.

[5] G. Xiang, Z. Qiuyu, W. Hui and C. Yan, "Face recognition based on LBPH and regression of Local Binary features," *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, Shanghai, 2016, pp. 414-417, doi: 10.1109/ICALIP.2016.7846668

[6] A. M. Jagtap, V. Kangale, K. Unune and P. Gosavi, "A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV," *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, India, 2019, pp. 219-224, doi: 10.1109/ISS1.2019.8907965

[7] T. Ahonen, A. Hadid and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.

[8] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002, doi: 10.1109/TPAMI.2002.1017623.

[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518, Kauai, Hawaii, USA, December 2001.

[10] G. Yang and T.S. Huang, "Human Face Detection in a Complex Background", Pattern Recognition, Vol. 27, No 1, pp. 53–63, 1994

[11] B. Moghaddam and A. Pentland, "Maximum Likelihood Detection of Faces and Hands", Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995.

[12] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection", CVPR, 1996.

[13]]L. Liu et al., "Deep Learning for Generic Object Detection: A Survey", Int. J. Comput. Vis., vol. 128, no. 2, pp. 261-318, Sep. 2018.

[14] Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustainable cities and society, 66, 102692.

[15] M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud and J. -H. Kim, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216386.

[16] A. Arjun Raj, M. Shoheb, K. Arvind and K. S. Chethan, "Face Recognition Based Smart Attendance System," 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 2020, pp. 354-357, doi: 10.1109/ICIEM48762.2020.9160184.

[17] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[18] https://www.mdpi.com/1099-4300/19/6/242

[19] https://en.wikipedia.org/wiki/AlexNet

[20] https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/

[21] K. Sanath, M. K, M. Rajan, V. Balamurugan and M. E. Harikumar, "RFID and Face Recognition based Smart Attendance System," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 492-499, doi: 10.1109/ICCMC51019.2021.9418481.

[22] Dagar, D., Dagar, Y., & Sharma, R. (2022). Face-mask recognition and detection using deep learning. Paper presented at the *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, COM-IT-CON 2022,* 692-697. doi:10.1109/COM-IT-CON54601.2022.9850742

[23] Habeeb, Z. Q., & Al-Zaydi, I. (2023). Incorrect facemask-wearing detection using image processing and deep learning. *Bulletin of Electrical Engineering and Informatics, 12*(4), 2212-2219. doi:10.11591/beei.v12i4.4786

[24] Vipul, V., Manoj, N., Nair, L. S., & Rani, S. S. (2023). An improved machine learning approach to detect real time face mask. Paper presented at the *Proceedings - 5th International Conference on Smart Systems and Inventive Technology, ICSSIT 2023,* 889-893. doi:10.1109/ICSSIT55814.2023.10060976

[25] Al-Selwi, H. F., Hassan, N., Ghani, H. B. A., Hamzah, N. A. B. A., & Aziz, A. B. A. (2023). Face mask detection and counting using you only look once algorithm with jetson nano and NVDIA giga texel shader extreme. *IAES International Journal of Artificial Intelligence, 12*(3), 1169-1177. doi:10.11591/ijai.v12.i3.pp1169-1177

[26] Umer, M., Sadiq, S., Alhebshi, R. M., Alsubai, S., Al Hejaili, A., Eshmawi, A. A., . . . Ashraf, I. (2023). Face mask detection using deep convolutional neural network and multi-stage image processing. *Image and Vision Computing, 133* doi:10.1016/j.imavis.2023.104657

[27] Sheikh, B., & Zafar, A. (2023). RRFMDS: Rapid real-time face mask detection system for Effective COVID-19 monitoring. *SN Computer Science, 4*(3) doi:10.1007/s42979-023-01738-9

[28] Pham, T. -., Nguyen, V. -., & Huh, J. -. (2023). Integration of improved YOLOv5 for face mask detector and auto-labeling to generate dataset for fighting against COVID-19. *Journal of Supercomputing, 79*(8), 8966-8992. doi:10.1007/s11227-022-04979-2

[29] Pawar, S., Jain, M., Akole, P., Mahagaonkar, S., & Kapse, R. (2022). Real-time CNN-based face mask detection system. *Artificial intelligence in information and communication technologies, healthcare and education: A roadmap ahead* (pp. 43-57) doi:10.1201/9781003342755-6

[30] Jayaswal, R., & Dixit, M. (2022). A face masks detection system: An approach to fight with COVID-19 scenario. *Concurrency and Computation: Practice and Experience, 34*(28) doi:10.1002/cpe.7394

[31] Cao, Z., Shao, M., Xu, L., Mu, S., & Qu, H. (2020). Maskhunter: Real-time object detection of face masks during the COVID-19 pandemic. *IET Image Processing, 14*(16) doi:10.1049/iet-ipr.2020.1119