

Real Time Face Mask Detection and Recognition using Python

Roshan M Thomas¹

Dept. of Comp Science & Engineering Mangalam College of Engineering, Kottayam, India.

Tintu Samson³

Dept. of Comp Science & Engineering Mangalam College of Engineering, Kottayam, India.

Motty Sabu²

Dept. of Comp Science & Engineering Mangalam College of Engineering, Kottayam, India.

Shihana Mol B⁴

Dept. of Comp Science & Engineering Mangalam College of Engineering, Kottayam, India.

Tinu Thomas⁵

Dept. of Comp Science & Engineering Mangalam College of Engineering Kottayam, India.

Abstract- After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. According to the World Health Organization, the corona virus COVID-19 pandemic is causing a global health epidemic, and the most successful safety measure is wearing a face mask in public places. Convolutional Neural Networks (CNNs) have developed themselves as a dominant class of image recognition models. The aim of this research is to examine and test machine learning capabilities for detecting and recognize face masks worn by people in any given video or picture or in real time. This project develops a real-time, GUI-based automatic Face detection and recognition system. It can be used as an entry management device by registering an organization's employees or students with their faces, and then recognizing individuals when they approach or leave the premises by recording their photographs with faces. The proposed methodology makes use of Principal Component Analysis (PCA) and HAAR Cascade Algorithm. Based on the performance and accuracy of our model, the result of the binary classifier will be indicated showing a green rectangle superimposed around the section of the face indicating that the person at the camera is wearing a mask, or a red rectangle indicating that the person on camera is not wearing a mask along with face identification of the person.

Keywords – Face Recognition and Detection, Convolutional Neural Network, GUI, Principal Component Analysis, HAAR Cascade Algorithm

I. INTRODUCTION

Face Recognition is a technique that matches stored models of each human face in a group of people to identify a person based on certain features of that person's face. Face recognition is a natural method of recognizing and authenticating people. Face recognition is an integral part of people's everyday contact and lives. The security and authentication of an individual is critical in every industry or institution. As a result, there is a great deal of interest in automated face recognition using computers or devices for identity verification around the clock and even remotely in today's world. Face recognition has emerged as one of the most difficult

and intriguing problems in pattern recognition and image processing. With the aid of such a technology, one can easily detect a person's face by using a dataset of identical matching appearance. The most effective approach for detecting a person's face is to use Python and a Convolutional Neural Network in deep learning. This method is useful in a variety of fields, including the military, defense, schools, colleges, and universities, airlines, banks, online web apps, gaming, and so on. Face masks are now widely used as part of standard virus-prevention measures, especially during the Covid-19 virus outbreak. Many individuals or organizations must be able to distinguish whether or not people are wearing face masks in a given location or time. This data's requirements should be very real-time and automated. The challenging issue which can be mentioned in face detection is inherent diversity in faces such as shape, texture, color, got a beard/moustache and/or glasses and even masks. From the experiments it is clear that the proposed CNN and Python algorithm is very efficient and accurate in determining the facial recognition and detection of individuals.

I. RELATEDWORKS

Local Binary Pattern (LBP) method is used to filter the candidate region. LBP reflects the details of the face characteristics, focusing on the description of texture features. Therefore, after using global feature to identify the face, the local feature recognition LBP method is used to filter the candidate region. LBP reflects the details of the face characteristics, focusing on the description of texture features. This paper combines skin colour detection with LBP. If the number of pixels of the skin colour points exceeds the set threshold, the face image is initially determined. Otherwise it is a non-face image. Then the LBP algorithm is used to detect the candidate window. If the match is successful, it is a face image. Otherwise it is non-human face. Otherwise it is non-human face. LBP is not capable of detecting faces with masks or glasses in faces. [1]

A robust approach to face & facial features detection must be able to handle the variation issues such as changes in imaging conditions, face appearances and image contents. Here we present a method in which utilizes colour, local symmetry and geometry information of human face based on various models. The algorithm first detects most likely face regions or ROIs (Region-Of-Interest) from the image using face colour model and face outline model, produces a face colour similarity map. Then it performs local symmetry detection within these ROIs to obtain a local symmetry similarity map. These two maps are fused to obtain potential facial feature points. Finally, similarity matching is performed to identify faces between the fusion map and face geometry model under affine transformation. The output results are the detected faces with confidence values [2].

Face detection and eyes extraction has an important role in many applications such as face recognition, facial expression analysis, security login etc. Detection of human face and facial structures like eyes, nose are the complex procedure for the computer. This paper proposes an algorithm for face detection and eyes extraction from frontal face images using Sobel edge detection and morphological operations. The proposed approach is divided into three phases; pre-processing, identification of face region, and extraction of eyes. Resizing of images and gray scale image conversion is achieved in pre-processing. Face region identification is accomplished by Sobel edge detection and morphological operations. In the last phase, eyes are extracted from the face region with the help of morphological operations. [3].

YOLO has a fast detection speed and is suitable for target detection in real-time environment. Compared with other similar target detection systems, it has better detection accuracy and faster detection time. This paper is based on YOLO network and applied to face detection. In this paper, YOLO target detection system is applied to face detection. Experimental results show that the face detection method based on YOLO has stronger robustness and faster detection speed. Still in a complex environment can guarantee the high detection accuracy. At the same time, the detection speed can meet real-time detection requirements [4].

Partially Occluded Face Detection (POFD) problem is addressed by using a combination of feature-based and part based face detection methods with the help of face part dictionary. In this approach, the devised algorithm aims to automatically detect face components individually and it starts from mostly un-occluded face component called Nose. Nose is very hard to cover up without drawing suspicion. Keeping nose component as a reference, algorithm search the surrounding area for other main facial features, if any. Once face parts qualify facial geometry, they are normalized (scale and rotational) and tag with annotation about each facial

features so that partial face recognition algorithm can be adapted accordingly with the test image. [5].

We are focused on the face detection process and the role of interest regions of the human face. In order to locate exactly the facial area, we propose the use of horizontal and vertical IPC (Integral Projection Curves). The role of important patches of face: nose and eyes is investigated in this work. An efficient method based on PCA (Principal component analysis) followed by EFM (Enhanced Fisher Model) is used to build the characteristic features, these latter are sent to the classification step using two methods, Distance Measurements and SVM (Support Vector Machine). Finally, the effect of fusion of two modalities (2D and 3D) is studied and examined. [6].

Existing standards which were developed for recognizing the face with masks on it do not work well due to the unique structure of the human faces. Face recognition is one of the latest technologies being studied area in biometric as it has wide area of applications. But Face detection is one of the challenging problems in Image processing. The basic aim of face detection is determining if there is any face in an image & then locates position of a face in an image. Evidently face detection is the first step towards creating an automated system which may involve other face processing. The neural network is created & trained with training set of faces & non-faces. All results are implemented in MATLAB 2013 environment. [7].

II. PROPOSED METHODOLOGY

We use Convolutional Neural Network and Deep Learning for Real Time Detection and Recognition of Human Faces, which is simple face detection and recognition system is proposed in this paper which has the capability to recognize human faces in single as well as multiple face images in a database in real time with masks on or off the face. Pre-processing of the proposed frame work includes noise removal and hole filling in colour images. After pre-processing, face detection is performed by using CNNs architecture. Architecture layers of CNN are created using Keras Library in Python. Detected faces are augmented to make computation fast. By using Principal Analysis Component (PCA) features are extracted from the augmented image. For feature selection, we use Sobel Edge Detector.

A. *The Input Image*

Real-time input images are used in this proposed system. Face of person in input images must be fully or partially covered as they have masks on it. The system requires a reasonable number of pixels and an acceptable amount of brightness for processing. Based on experimental evidence, it is supposed to perform well indoors as well as outdoors i.e. passport offices, hospitals, hotels, police stations and schools etc.

B. The Pre-processing Stage

Input image dataset must be loaded as Python data structures for pre-processing to overturn the noise disturbances, enhance some relevant features, and for further analysis of the trained model. Input image needs to be pre-processed before face detection and matching techniques are applied. Thus pre-processing comprises noise removal, eye and mask detection, and hole filling techniques. Noise removal and hole filling help eliminate false detection of face/ faces. After the pre-processing, the face image is cropped and re-localised. Histogram Normalisation is done to improve the quality of the pre-processed image.

C. The Face Detection Stage

We perform face detection using HAAR Cascade algorithm. This system consists of the value of all black pixels in greyscale images was accumulated. They then deducted from the total number of white boxes. Finally, the outcome is compared to the given threshold, and if the criterion is met, the function considers it a hit. In general, for each computation in Haar-feature, each single pixel in the feature areas can need to be obtained, and this step can be avoided by using integral images in which the value of each pixel is equal to the number of grey values above and left in the image.

$$\text{Feature} = \sum_{i \in \{1..N\}} w_i \cdot \text{RecSum}(x, y, w, h),$$

where $\text{RecSum}(x, y, w, h)$ is the summation of intensity in any given upright or rotated rectangle enclosed in a detection window and x, y, w, h is for coordinates, dimensions, and rotation of that rectangle, respectively. Haar Wavelets represented as box classifier which is used to extract face features by using integral image

D. The Feature-Extraction Stage

Feature Extraction improves model accuracy by extracting features from pre-processed face images and translating them to a lower dimension without sacrificing image characteristics. This stage allows for the classification of human faces.

E. The Classification Stage

Principal Component Analysis (PCA) is used to classify faces after an image recognition model has been trained to identify face images. Identifying variations in human faces is not always apparent, but PCA comes into the picture and proves to be the ideal procedure for dealing with the problem of face recognition. PCA does not operate classifying face images based on geometrical attributes, but rather checks which all factors would influence the faces in an image. PCA was widely used in the field of pattern recognition for classification problems. PCA demonstrates its strength in terms of data reduction and perception.

F. Training Stage

The method is based on the notion that it learns from pre-processed face images and utilizes CNN model to construct a framework to classify images based on which group it belongs to. This qualified model is saved and used in the prediction section later. In CNN model, the stages of feature extraction are done by PCA and feature selection done by Sobel Edge Detector and thus it improves classification efficiency and accuracy of the training model.

G. Prediction Stage

In this stage, the saved model automatically detects the face mask image captured by the webcam or camera. The saved model and the pre-processed images are loaded for predicting the person behind the mask. CNN offers high accuracy over face detection, classification and recognition produces precise and exact results. CNN model follows a sequential model along with Keras Library in Python for prediction of human faces.

IV. MODULES

The proposed system contains the following modules:

- A. Pre-processing Images
- B. Capture image ()
- C. Upload image ()
- D. Classifier(image)
- E. Prediction(image)

A. Pre-processing Images

The input image is captured from a webcam or camera in real-time world. The frames (images) from the dataset are loaded. Face images are cropped and resized after they have been loaded. Later, noise distortions in the images are suppressed. Normalization is then done to normalize the images from 0-255 to 0-1 range.

B. Capture image ()

In this Module we are able to capture real time images. We do this by the help of Flutter and applying in to the Classifier Model.

Input: Nothing

C. Upload image ()

Here we can browse the image and upload for finding the Plant disease. We need to fetch the image. And this image passes to Classifier Module.

Input: Nothing

Output: Image

D. Classifier(image)

Following data Preprocessing of the images, will apply to the Classifier. Here it will find out the feature of the images. Mainly in this module feature extraction occurs. Image similarity features will be stored in to the model which gets created.

Input: Image
Output: Model

E. Prediction(image)

In this Module prediction of person take place. Here the browsed image will be placed in to the model and output will be shown as based on which label its get matched the most.

Input: Image
Output: Predicted Label

V. SYSTEM ARCHITECTURE

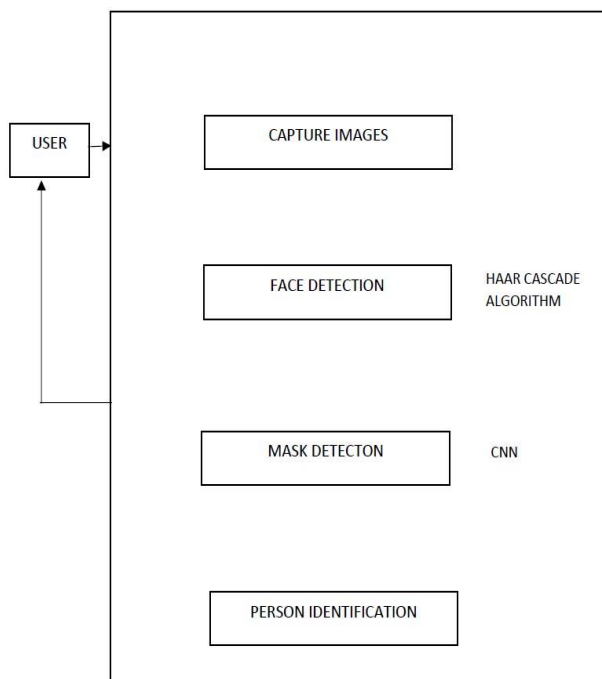


Fig1. System Architecture

A. User

User refers to person standing in front of a webcam or camera in a real world scenario.

B. Capture Images

The webcam or camera captures images which are then used as dataset to train the model. If the dataset captures human faces in different masks and in different backgrounds along with large number of human face images, then the accuracy of the training model increases.

C. Face Detection

For face detection, we use HAAR Cascade algorithm. In this method all black pixels in greyscale images was accumulated. They then deducted from the total number of white boxes. Finally, the outcome is compared to the given threshold, and if the criterion is met, the function considers it a hit.

D. Mask Detection

For Mask Detection, we use a sequential CNN model along with inbuilt Keras Library in Python. The sequential CNN model is trained from dataset of human faces with or without masks on the faces. It forms a logic from the pre-processed images like a human brain, then the model detects the face along with mask using feature extraction and feature selection. After identification of the mask along with face of the person, it forwards to the prediction or identification stage.

E. Person Identification

In this stage, the trained model predicts the face of the person behind the mask according to the trained model. The prediction is based on the number of images trained by the model and its accuracy. Finally, the system displays that the person name along with the indication of he or she wearing a mask or not.

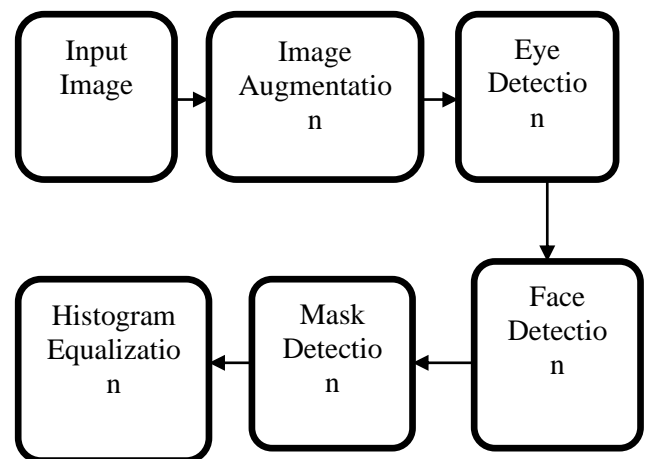


Fig2. Pre-processing

F. Dataset

The proposed model has datasets captured from individual's person. The dataset of faces is classified into with masks and without masks and is stored in different databases. Each folder consists 40 to 60 images of an individual person respectively. The individual's person face images should have images captured from different masks and different backgrounds so the accuracy of training model increases. The dataset is integrated with Keras Library in Python. Larger the dataset more accurate the training model. So dataset images are directly congruent to accuracy of the training model.

G. Data Pre-Processing

This module is used for read image. After reading we resize the image if needed me rotate the image and also remove the noises in the image. Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise. Later normalization is done to clean the images and to change the intensity values to pixel format. The output of this stage is given to training model.

Input: image

Output: pixel format

H. Segmentation

Segment the image, separating the background from foreground objects and we are going to further improve our segmentation with more noise removal. We separate different objects in the image with markers.

Input: pixel format

Output: image

I. Edge detection

Sobel edge detector is using. It is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. 2-D spatial gradient measurement on the image is performed by Sobel operator. Each pixel of the image is operated by Sobel operator and measured the gradient of the image for each pixel. Pair of 3×3 convolution masks is used by Sobel operator, one is for x direction and other is for y Direction. The Sobel edge enhancement filter has the advantage of providing differentiating (which gives the edge response) and smoothing (which reduces noise) concurrently.

Input: image

Output: image

J. Localization

Find where the object is and draw a bounding box around it.

Input: image

Output: localized image

K. Feature Selection

The biggest advantage of Deep Learning is that we do not need to manually extract features from the image. The network learns to extract features while training. You just feed the image to the network (pixel values).

What you need is to define the Convolutional Neural Network architecture and a labelled dataset. Principal Component Analysis (PCA) is a useful tool for doing this. PCA checks all the factors influencing the faces rather just checking its geometrical factors. Thus using PCA gives accurate and precise detection and recognition result of faces.

Input: image pixel format

Output: labels

L. CNN Architecture creation

A sequential CNN model is designed specifically for analyzing the human faces with mask on it or not. The Convolutional Neural Network Architecture layers will be created using the Keras library in Python. The convolutional layer is used for mask detection. It extracts the features of face images using Principal Component Analysis (PCA) and converts them into a lower dimension without losing the image characteristics. The output of the convolutional layer will be the input of the next Batch Normalization layer. The Batch Normalization layer standardizes the inputs to a layer for each mini-batch. This has the effect of

stabilizing the learning process and dramatically reducing the number of training epochs. After this, the face images undergo classification. If the images are tested, then model accuracy calculations and predications takes place. If non-test images come, then first the images are trained along with it its validation testing is also done. If it is validating, then the model is trained and saved for further calculations. Otherwise, if it is non-validate, then it undergoes network training and calculations are done for losing weights and are adjusted accordingly. Finally, the CNN model gives accuracy and prediction of the human face behind the mask.

M. Training

The pre-processed face images are directed to the CNN model for training. Based on the dataset given, a logic is formed in the CNN to categorize the faces according to their features. This trained model is saved. The trained model is capable of categorizing human faces based on with or without masks on it. Training model is done with the help of a sequential CNN model and HAAR Cascade Algorithm.

N. Predication

In this phase, when a person comes in front on a webcam, the image is captured and predicted by the CNN model according to the logic learned by the sequential model. The image undergoes pre-processing. This pre-processed images and the saved CNN model are then loaded. Based on the algorithm interpreted by the system predicts and detects the human faces according to trained model.

VI. RESULT

This proposed work uses a sequential Convolutional Neural Network for detecting and recognizing human faces of individuals with mask or without it. CNN model and Haar Cascade Algorithm facilitates automatic detection and recognition of human face which overcome the noise variations and background variations caused by the surrounding and provide more accurate and precise result. It also helps to overcome the uneven nature of the current trend of face recognition and detection. From the experiments it is clear that the proposed CNN achieves a high accuracy when compared to other architectures. The proposed algorithm works effectively for different types of images. These results suggest that the proposed CNN model reduces complexity and make method computationally effective. The proposed system works well effectively for grayscale as well as for the colour image with masks on it or without masks on it.

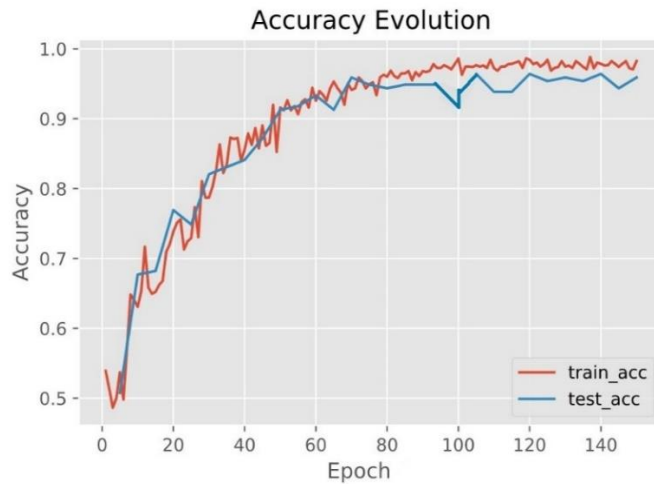


Fig3. Accuracy analysis

| | Precision | Recall | F1-Score | Support |
|--------------|-----------|--------|----------|---------|
| With_mask | 0.99 | 0.99 | 0.99 | 138 |
| Without_Mask | 0.99 | 0.99 | 0.99 | 138 |
| Accuracy | | | 0.99 | 276 |
| Macro Avg | 0.99 | 0.99 | 0.99 | 276 |
| Weighted Avg | 0.99 | 0.99 | 0.99 | 276 |

Table1. Performance Analysis

VII. CONCLUSION

Our proposed system can detect and recognize human face(s) in real-time world. Compared to the traditional face detection and recognition system, the face detection and recognition based on CNN model along with the use of Python libraries has shorter detection and recognition time and stronger robustness, which can reduce the miss rate and error rate. It can still guarantee a high test rate in a sophisticated atmosphere, and the speed of detection can meet the real time requirement, and achieve good effect. The proposed CNN model shows greater accuracy and prediction for detecting and recognising human faces. The results show us that the current technology for face detection and recognition is compromised and can be replaced with this proposed work. Therefore, the proposed method works very well in the applications of biometrics and surveillance.

VIII. REFERENCES

- [1] Zheng Jun, Hua Jizhao, Tang Zhenglan, Wang Feng "Face detection based on LBP", 2017 IEEE 13th International Conference on Electronic Measurement & Instruments.
- [2] Q. B. Sun, W. M. Huang, and J. K. Wu "Face Detection Based on Color and Local Symmetry Information", National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore.
- [3] Based on Color and Local Symmetry Information", National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore
- [4] Wang Yang, Zheng Jiachun "Real-time face detection based on YOLO", 1st IEEE International Conference on Knowledge Innovation and Invention 2018
- [5] Dr. P. Shanmugavadivu, Ashish Kumar, "Rapid Face Detection and Annotation with Loosely Face Geometry", 2016 2nd International Conference on Contemporary Computing and Informatics (ic3i)
- [6] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," IEEE Transactions on pattern analysis and machine intelligence, vol. 23, pp. 681-685, 2001.
- [7] S. Saypadith and S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 2018, pp. 1318-1324
- [8] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.
- [9] Matthew D Zeiler, Rob Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014: Computer Vision – ECCV 2014 pp 818-833.
- [10] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, 2017, pp. 650-657.