A STUDY ON FACE AND FACE MASK DETECTION USING HAAR CASCADE

by

Raisa Mallik (160021072)
Iffat Rezwana (160021090)
Rokaiya Sharmin (160021138)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering

Islamic University of Technology (IUT)
Gazipur, Bangladesh

February 2021

A STUDY ON FACE AND FACE MASK DETECTION USING HAAR CASCADE

Approved by:

Dr. Golam Sarowar

Supervisor and Professor

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Board bazar, Gazipur-1704.

Date: ……

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

· BLEU	Bilingual Evaluation Understudy (BLEU)
· BRNN	Bidirectional Recurrent Neural Network (BRNN)
· CAE	Contractive autoencoder (CAE)
· CMMs	Conditional Markov Models (CMMs)
· CNN	Convolutional Neural Networks (CNN)
· CTC	Connectionist Temporal Classification (CTC)
· F1 Score	Harmonic Precision Recall Mean (F1 Score)
· FFT	Fast Fourier transform (FFT)
· LDA	Latent Dirichlet allocation (LDA)
· LDA	Linear discriminant analysis (LDA)
· LSI	Latent Semantic Indexing (LSI)
· LSTM	Long Short-Term Memory (LSTM)
. MAP	Maximum A Posteriori (MAP) Estimation
· MCMC	Markov Chain Monte Carlo (MCMC)

· MDL Minimum description length (MDL) principle

· NER Named Entity Recognition (NER)

· NERQ Named Entity Recognition in Query (NERQ)

· NFL No Free Lunch (NFL) theorem

· NTM Neural Turing Machine (NTM)

· POS Part of Speech (POS) Tagging

· RF Random Forest (RF)

· RL Reinforcement Learning (RL)

· RLFM Regression based latent factors (RLFM)

· RNNLM Recurrent Neural Network Language Model

· ROC Receiver Operating Characteristic (ROC)

· ReLU Rectified Linear Unit (ReLU)

· SCH Stochastic convex hull (SCH)

· SGD Stochastic Gradient Descent (SGD)

· SSVM Smooth support vector machine (SSVM)

· SVD	Singular Value Decomposition (SVD)
· SVM	Support Vector Machine (SVM)
· object detection algorithm	YOLO (object detection algorithm)
· object detection algorithm	YOLO9000 (object detection algorithm)
· object detection algorithm	YOLOv2 (object detection algorithm)

DEDICATION

This study is dedicated to our beloved parents.
For their unwavering affection, support, and inspiration.

ACKNOWLEDGMENT

All praise is due to Allah, who has blessed us with the wisdom and ability to complete this study by his mercy.
We would like to express our heartfelt gratitude to Dr. Golam Sarowar, Professor, Department of Electrical and Electronic Engineering, Islamic University of Technology, for his unwavering support, constant inspiration, helpful attitude, careful guidance, and invaluable suggestions during this thesis project. This research was made possible by his thoughtful ideas and thoughts.
We'd also like to thank family, teachers and those who helped us finish the thesis by providing useful feedback and details, either directly or indirectly.

ABSTRACT

Computer technology used in various programs to recognize people's faces in digital images is apprehended as facial recognition. There is a wide range of applications in the fields of content-based content retrieval, video encoding, video conferencing, crowd viewing, and smart computer interaction. Object detection is a computer vision approach that allows us to recognize and locate objects in an image or video. The main objective of this research was to develop a security system that could detect a face mask, a person's face, and the number of people standing in front of the camera in real-time using real-time video capture. OpenCV is a cross-platform library of over 2500 algorithms that have been optimized and were chosen for our project due to its benefits. One of the most basic machine learning-based methods is the Haar cascade classifier. Haar features are particularly useful for mask detection because they are excellent at detecting edges and lines, simultaneously, cascade classifiers are one of the few real-time algorithms available. The accuracy level was very poor when a built-in Haar cascade was used for the project, to solve this issue, a new Haar cascade was trained to improve accuracy which was proved to be effective. Three separate features, such as face mask, face, and human detection, have been brought under one project. A GUI was created using the Tkinter module to achieve that. Face mask detection, face detection, and human detection are the three key features of this project to ensure social distancing digitally. This designed system can be used effectively in applications like, keeping any location protected from intruders, ensuring face-mask wearing, and preserving social distance.

CHAPTER ONE
INTRODUCTION

To stop the COVID-19 pandemic from spreading, precautions must be taken. Furthermore, with the increase of violence, a proper security system is now a must-have. Keeping these factors in mind, we created a comprehensive framework that includes programs for face recognition, mask detection, and human detection. Our framework was primarily built using Haar cascade classifiers. We used a data set with positive and negative images to train Haar cascades, validate, and evaluate the model. Photos and live video streams were used to create the model. For better accuracy, we used the HOG approach in our face recognition system. This entire system can be used in a variety of settings, including schools, universities, airports, and other locations where proper protection is needed.

1.1 LITERATURE REVIEW:

The goal of facial recognition is to locate the region of the face and extract it from the background. This is a necessary initiative in facial recognition systems. According to the WHO and therefore the United States Centers for Disease Control and Prevention, maneuvering masks are a portion of a wide-ranging platform of inhibition and control procedures that preclude the escalation of definite diseases initiated by lung viruses, including COVID-19. People should wear masks to avoid the risk of transmission of the virus and maintain a social distance of at least 2 meters between individuals [1] to stop the escalation of the disease from person to person. Main qualifications of the pose of a face within the detection window. The labelling of objects within the image and the classification according to certain classes is understood as image classification. The neural network has to process, recognize and classify different images with different objects depending on the type of image element, so it is considered a more advanced version of image recognition. The ability of AI to recognize, classify and recognize the object is known as image recognition.

The system first recognizes the face and then classifies it as a person's face. Then decide if it belongs to the owner of the smartphone [1]. Faces stand recognized in three stages: For a piece image opening, the pretense of the "face" is paramount appraised; the pose assessment is then cast-off to exchange the image opening [9]. The succeeding detector at that juncture catalogue the window. A concealed flaw in such a designation is that the perilous recognition proportion is crudely the product of the accurate classification proportions of the two classifiers, because the inaccuracies of the twofold classifiers are somewhat self-governing. First, the position of the window is estimated and then one of the specific rotation detectors is named to classify the window. If this detachment is a reduced amount of than a preset inception, the warship of these checkbox will be distributed and set constructed on the former identified face point. Due to Python coding, we switched to OPENCV to measure the speed of this face tracking scheme. We found

that Viola and Jones' face recognition is better suited for real-time face recognition as it requires fewer CPU resources [9] and takes less time. The detection is defined by the coordinates of the upper left corner and the height and height of the rectangle that makes up the area detected. It means the recognized face; we will draw a rectangle on it. The coordinates indicate the line in addition to the pixel column in the image. We will get these synchronizations from the variable surface. The procedure of aspect face mask partakes is endorsed by health mavens and political the established order. To diminish the menace of taint. 95% of the world's population sleep in countries that publicly recommend or mandate the use of masks during pandemic detection. Recognition seems to be an amazing problem in the field of image processing and image processing. To hinder the transmission of COVID-19, the proposed model is generally integrated into surveillance cameras [21], so they are often used to detect people who do not wear face masks. If a face is recognized in this version of the algorithm, then the use of the mask is assumed to be valid if the eyes are recognized and if at the same time the recognition of the nose, mouth and therefore chin fails. The tactic is believed to analyze faces in traditional masks. The main motive of the algorithm uses in particular the real-time recognition of various facial movements with various resolutions in one loop video sequences and cascading classifiers based on hair characteristics [21], which create in particular the target of hair-like characteristics to encode the variations. of the pixels contained in the image.

The process of marking objects within the image and arranging them according to certain classes is understood as image classification. The neuronal network must process various images with different objects, recognize and classify them [21] after the type of element in the image. It is considered as a more advanced version of the picture ID. The ability of the KI to recognize, classify and recognize the thing is referred to as image recognition. The system first recognizes the face, then classifies it as a face of a person and then decides whether it belongs to the owner of the smartphone. The goal of facial recognition is to locate and extract the area of the face from the background. This is a necessary initiative in face recognition systems. According to the WHO and thus the US [21] centers for the control and prevention of diseases, the mask maneuver is part of a comprehensive package of prevention and control measures that reduce the escalation of certain viral respiratory diseases. It is recommended to carry facial masks to avoid the risk of virus transmission and also a social separation of at least 2 m between people up to stop the escalation of the person's disease to a person. Main assessments of the pose of a face within the recognition window.

Faces are recognized in three steps: For each image window, the pose of the "face" is first estimated; the estimate is used to reduce the rotation of the image window. Then the second detector classifies the window. A bad mistake with such a name is that the critical recognition rate is roughly a product of the correct classification rates of the two classifiers (since the errors of the two classifiers are somewhat sovereign) [25]. First the window attitude is estimated and then a certain rotation rate detector. For this purpose, a tactile detection window, which exists as an example for two adjacent rectangular zones, is arranged in the image; Then the variation of this part of the image is calculated by

subtracting the sum of the resulting pixel intensities from the areas covered by the black and white areas. The value obtained from this calculation corresponds to a coded characteristic similar to Haar, which recognizes a change in texture or the things of a boundary within the image.
The window moves in such a way that when scanning the entire surface of the image and its size increases to form sure robustness against scaling fluctuations. Also, to optimize the time it takes to calculate these properties, the use of human presence detection can be a variety of technologies and methods to detect the presence of a person's body during an interest intonation or review. That a computer, smartphone (or another software-controlled device) is operated by one person. Social distance could also be a recommendation to the Earth Health Organization to publicly minimize the spread of Covid-19 [11]. A graphical interface is nothing but an interface that provides an interface that allows you to interact with computers and share your experience by entering a command in your Several applications that are loaded by the facility's GUI are: Create a calculator with an interface and functions that are retained throughout a calculator. The next step is to initialize the window manager using the Tkinter method [12] and assign a variable that leaves a blank window. It is the most widely used standard method using the Python interface for the Tk GUI toolkit that comes with OpenCV python. In the future, we would like to implement the hardware as part of our system using Raspberry Pi.

1.2 THESIS OVERVIEW:

Covid-19, one of the biggest threats the world is facing right now. After the massive breakout of this pandemic, there emerges an austere need of adapting some cautionary measures and wearing a face mask while being among people is the fundamental one. According to the WHO [29] and the Centers for Disease Control and Prevention (CDC) of the United States, the maneuvering of masks is part of a complete package of the prevention and control measures that can curtail the escalation of certain respiratory viral diseases, including COVID-19. The process of interim guidance development during emergencies consists of a transparent and powerful process of evaluation of the available evidence on benefits and harms, synthesized through operated systematic reviews and expert consensus-building facilitated by methodologist. Surveys have shown that mask-off activities clearly have potential implications for COVID-19 transmission [29] in the community. So far, there is no news about any clinically approved antiviral medicine or vaccines that are fruitful against COVID-19. It has escalated speedily across the world, bringing colossal health, economic, environmental, and social challenges to the entire human population [29].
One of the biggest obstacles to viral infection is maintaining distance within the community. Infection occurs when a verminous individual coughs, sneezes, or speaks. The lungs can also be done on the lungs. According to recent research, it is possible that even people who are sick but asymptomatic may be involved in the outbreak of COVID-19. The novel corona virus reaches the final stage of human-to-human infection, but as

far as in place we know, the corona virus infection that causes corona virus (COVID-19) in 2019 can also cause transient carriers. Supporting skin of at minimum 2 m [1] In order to prevent the individual growth of the disease, a separation of communities between entities occurs.

1.3 INTRODUCTION TO MACHINE LEARNING:

Machine learning (ML) is the analysis of computer algorithms that refines perpetually through experience [16]. It is a sub field of AI (artificial intelligence), and its goal is to realize the formation of data, arrange that data into models that can be acknowledged and deployed by people. Despite being a field within computer science, it diverges conventional computational methods. Algorithms are blocks of explicit program commands applied by computers to enumerate or hurdle elucidation in conventional computing. Proceeding the other hand, machine learning procedures allow training workstations to input data and generate price lists using math tests. As a result, machine learning produces faster decisions when a computer represents a document sample. Another area that we have used extensively in machine learning is face recognition machinery. This allows users to tag and share snaps of their friends on social media. With OCR [16], you can: Turn text interested in motion pictures. You can now get search engine suggestions using machine learning. The movies and TV show you can watch below depend on your preference. [16] Self-driving cars based on contraption learning roaming are expected to hit consumers soon.

The field of machine learning is constantly advancing. In this, the tasks are normally systematized into wide categories on the basis of how learning is being received or how feedback on the learning is being given to the system developed. Machine learning has two most widely adopted methods. They are

- *Supervised Learning*: In this method, algorithms are trained on the basis of example input and output data that is labeled by humans. It utilizes patterns to anticipate label values on additional unlabeled data [16]. The motive of this method is to make the algorithm able to "learn" by weighing up its authentic output with the "taught" outputs to find inaccuracy and alter the model accordingly.

  Supervised learning commonly utilized is to use historical data to forecast statistically likely future events.

- *Unsupervised learning:* This method allows the algorithm without tagged data to recognize the structure in your input data. It is mainly used for transitional data. You can look at complex data that is bigger and seemingly unrelated to organize it

in potentially meaningful ways without getting the "right" answer [16]. With this method, the untagged dog photos can be used as input to the algorithm to find similarities and classify the dog photos together.

A few of the popular approaches used in machine learning [16] are

- k-nearest neighbor
- Decision Tree Learning
- Deep Learning

1.3.1 IMAGE DETECTION:

Computer technology that processes the image and detects objects in it is known as image detection. Though people often confuse image detection with image classification, there exists a clear difference between them [10]. We use Classification to classify image items. And if we only require to locate them, such as perceive the number of objects in the picture, we should use Image Detection.
A popular application area of this technology is fake image detection. It is possible to differentiate the original picture from the photo shopped or counterfeited [10] one by using this technology.

HOW MACHINE LEARNING IMPROVES IMAGE DETECTION

We need to display these objects first in order to train the AI tool to recognize these defined objects. First, we need to "feed" the AI [10] with the tagged data: images containing the required objects, the coordinates of the elements, the location and class names. It is better to choose images with different positions of the object so that the elements change their coordinates and sizes during machine learning. It helps the AI to see that although this object is in different places in the image and is both large and small, these changes have no effect on its class. To apply deep machine learning to image recognition, developers use Python in conjunction with open-source libraries such as OpenCV Image Detection, Open Detection, Luminoth, ImageAI [10], and others. Simplified by the use of these libraries, they provide a ready-to-use environment.

- ***IMAGE CLASSIFICATION***

The process of labeling objects in a picture and then separating them into a single layer is called image separation. A network of terrified systems [10] has to process, detect and classify different images using different materials. It is considered the most advanced version of image acquisition for this reason, depending on the object type of the image.

The most accurate method of machine learning solutions for image classification and recognition is CNN or Convolutional Neural Network. It applies filters to detect certain features in the image. The way it will work completely depends on the type of the applied filter.
So, the network should be provided with as many different features as possible when applying machine learning solutions to image classification.

- ***IMAGE RECOGNITION***

The ability of AI to detect the object, classify, and recognize it is known as image recognition. It is a mix of Image Detection and Classification [10]. An example of image recognition is unlocking our smartphones by scanning our faces. In this, the system detects the face first, then classifies it as a human face, and then decides if it belongs to the owner of the smartphone.

1.4 FACE DETECTION:

Computer technology used in various programs to recognize people's faces in digital images is known as facial recognition. This problem occurs with the computer finding and locating one or more faces in the image. The purpose of face recognition is to be able to identify and remove more areas in the background, which is a necessary first step in the face recognition process. There is also a wide range of applications in the areas of content-based content retrieval, video encoding, video conferencing, crowd viewing, and smart computer interaction [14]. The facial recognition problem often attracts a lot of attention from investigators. Finding a face in an image is easy for humans, but difficult for computers because of its strong facial expressions. There are many different methods, from simple advanced algorithms to complex advanced methods [15] that use advanced pattern recognition methods. In fact, there are two main processes involved in awareness raising:
i)feature-based methods: it uses hand-crafted filters to search for and detect faces.
ii)image-based methods: it learns holistically how to extract faces [14] from the entire image.
It is assumed in many of the current face recognition techniques the availability of frontal faces of similar sizes [14, 163]. But actually, this assumption may not be valid because of the varied nature of face appearance and environmental conditions.

Figure 1.1: Typical training images for face recognition

Figure 1.1 represents typical training images for face recognition. They are typical test images used in face classification research. For reliable face classification techniques, the exclusion of the background in these images is very important. Although, when we do consider the realistic application scenarios like the given example in Figure, here the face might be occurred in a different and another background and in many different positions [14]. The recognition systems based on the standard face images might mistake some areas of the background as a face. To remove the difficulty, all we need to do is, we need a visual front-end processor to localize and extract the face region from the background.

Figure 1.2: A realistic face detection method situation

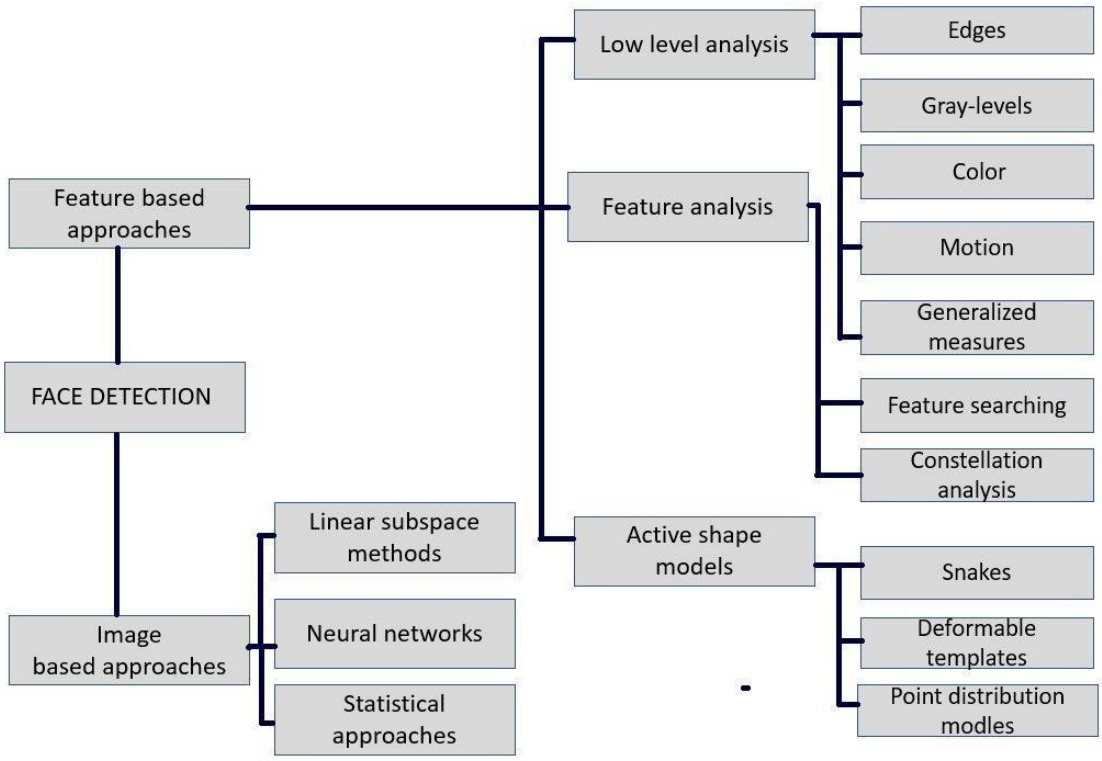


Figure 1.3: Face detection divided into approaches

Face recognition technology can also be applied to a wide range of applications, bio-metrics, video conferencing, photo and the video data indexing, and computer integration. One of the many possible usable methods of bio-metrics. Therefore, many biometric systems [14] rely on facial recognition and other biometric features such as voice and fingerprints. As the number of digital images available on the Internet increases and the use of digital video in archives increases, facial recognition has become a staple of many photo search systems. Content (CBIR).

1.4.1 FACE DETECTION PROCESS:

As discussed formerly, Face recognition may be a technique of identifying or verifying [4] the characteristics of a private expending their face. There are various algorithms which will do face recognition but their accuracy might vary. Face detection is typically the primary step enroute for many face-related expertise, like face recognition or verification. However, face detection can have very convenient applications. The foremost successful application of face recognition would probably be photo taking [4]. Once taking a photograph of friends, the face detection algorithm erected into your camera detects where the faces are and regulates the main target accordingly. The consequences of the detection give the face location strictures and it might be required in a number of forms, as an illustration, a rectangle casing the significant part of the face,

eye centers or bench mark including eyes, nose and mouth corners, eyebrows, nostrils, etc.

Face detection process using Feature Based Approach

Items are often viewed as separate. The face recognition process has many characteristics that can be found in the middle of the face and many other subjects. [4] Structural features such as eyes and nose have been removed following the face. , Mouth etc. Can be used for face recognition, a fine-grained human face can distinguish the faces of other objects, and the formation of spikes helps identify objects on the face.

Face detection process using Image Based Approach

Photo-based methods are generally based on mathematical research and machine learning processes to discover the Germanic nature of face and surface images. Cognitive factors lie in some typical blood circulation or specific functions [4]. After that, facial recognition begins. These all use different techniques like neural networks, HMMs, SVMs, and AdaBoost reads. In the next section, we will look at how to use MTCNN [4] or Multi to distinguish faces. A convincing neural network cascade, image-based detection method [4].

1.4.2 FACE DETECTION ALGORITHM:

One of the ideal algorithms that use a feature-based approach is the Viola-Jones algorithm. The Viola Jones algorithm was technologically advanced by Paul Viola and Michael Jones in 2001 and is an object recognition framework that enables the recognition of image features in reality. The principle assesses the pose of a face within the recognition window [9]. The second can be a conventional facial detector. Faces are recognized in three steps: For each image window, the pose of the "face" is first estimated. [9] The pose estimate will not rotate the image window. The second detector then classifies the window. A latent error of such an arrangement is that the decisive detection rate is approximately the product of the correct classification rates of the 2 classifiers (given that the errors of the 2 classifiers are somewhat sovereign) the window pose is estimated, then one of the Specific rotation detectors are asked to classify the window. Throughout this document, detection of non-erect faces using the 2-step method [9] is covered. Within the main stage, the pose of each window is estimated using a decision tree built with structures such as those described by Viola and Jones. In the second stage, one of the vehicle-specific Viola Jones detectors catalogs the window.

1.4.3 FACE DETECTION USING PYTHON:

Face detection and Face Recognition are often used interchangeably but these are quite different. In fact, Face detection is only a part of Face Recognition. Face recognition may be a method of identifying or verifying the identity of a remote consuming their face. As cited before, here we are accomplishing to see how we will detect faces by using an Image-based approach. MTCNN [4] or Multi-Task Cascaded Convolutional Neural Network is certainly one among the foremost popular and most precise face detection tools that employ this principle. So, let's examine how we will use this algorithm in [4] Python to detect faces in real-time. The very paramount task we perform is detecting faces within the image or video stream. Now that we all know the precise location/coordinates of the face; we extract this face for further processing ahead. At this instant that we've cropped the face out of the image, we extract features from it. Here we are getting to use face embedding to extract the features out of the face. A neural network grosses a picture of the person's face as input and outputs a vector which represents the primary vital features [4] of a face. In machine learning, this vector is named embedding and thus we call this vector as face embedding. We aren't getting to train such a network here because it takes a big amount of knowledge and computation power to coach such networks. We'll use a pre-trained network trained by Davis King on a data set of 3 million images. The network outputs a vector of 128 numbers which represent the foremost important features of a face. Now that we all know how this network works, allow us to see how we use this network on our own data. We pass all the pictures in our data to the present pre-trained network to itch the respective embedding and save these embedding during a file for successive phases. Now that we've face embedding for each face in our data saved during a file, the subsequent step is to recognize a replacement t image that's not in our data. Therefore, the initiative is to compute the face embedding for the image using an equivalent network we used above then associate this embedding with the residue of the embedding we've.

1.4.4 FACE DETECTION USING OPENCV:

The facial recognition file is cross-platform. These are the images extracted from the HD video stream. Sometimes the face detection algorithms may get a result to some extent even with only the one face in the frame. In this case post-processing of the image was used to extract the coordinates of the face defined with the OpenCV and SimpleCV Hair Classifier libraries. When the system output provides a silent rectangle that shows the situation of the face and so the intergalactic of the centers of these. If this distance is less than a preset threshold, the emblem of these rectangles will be divided [25] and set as the end position. During this work we also implemented the face tracking application in the Python language using face recognition.

This method is validated to detect system limitations by checking for code errors and removing them. Also, due to Python's limited performance, go to OpenCV and measure the speed of this facial tracking system. Viola and Jones face detection requires less CPU resources and is less time consuming, making them ideal for real-time face detection [25]. Import the target image. Finding truth is determined by the upper left corner and the width and height of the rectangular surface. Draw a rectangle over it to identify the found surface. The rectangle OctOpenCV () is the rectangle above the image and I need to know the pixel links in the upper and lower left. Links point to different lines of pixels in the image. This sync can be easily found on different faces.

1.4.5 FACE DETECTION USING HAAR CASCADE:

In the 19th century a Hungarian mathematician, Alfred Haar gave the concepts of Haar wavelets, which are the sequence of rescaled “square-shaped” functions which together form a wavelet a family or basis. Voila and Jones adapted the thought via Haar wavelets and so developed the so-called Haar-like features.

Haar-like features are digital image features utilized in visual perception. All faces share some universal properties of the human face so much just like the eye’s region is darker than its neighbor pixels, and therefore the nose region so much is brighter than the attention region.

A simple way to determine which region is lighter or darker is to sum to up the pixel values of both the regions and compare them. The sum of pixel values within the darker region are going to be smaller than the sum of pixels within the lighter region. If the one side is the lighter than the opposite, it's going to be a foothold of an eyebrow or sometimes the center portion could also be shinier than the encompassing boxes, which may be interpreted as a nose this will be the accomplished using the Haar-like features and with the assistance of them, we will interpret the various parts of the face.

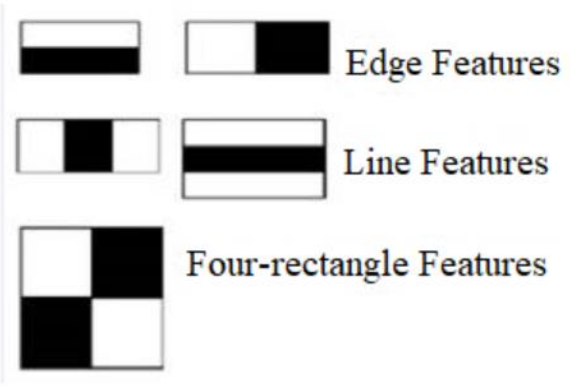


Figure 1.4: Haar Cascade features

Choose a facial recognition method for research that supports the actual needs of your device. No existing system is designed for everyone and works well in any application. The digital image feature is a Haar-like feature used for visual identification. It is named after its exact similarity to the Haar wavelength and is used in the first order to find the actual surface [17]. Features such as Haar find adjacent rectangular areas within the selected area and consider them when summarizing the pixel size of each area. This distinction does not subdivide the visual joints. The main advantage of Haar-like features over other features is the speed of editing. You need to run a face recognition algorithm. There is a balance between the two elements. Accuracy and speed are two important factors. Viola and Jones, known as integrated images using other photographic presentations, described a rapid performance test [17]. This has proven to be an effective way to speed up classified jobs.Adaboost, representing Adaptive Boost, could be an algorithm for machine learning. The Daboost algorithm captures training data and describes the weak differentiation function of all training data samples. Sensitivity to audio data and individual values.

1.5 MASK DETECTION:

Face masks are recommended to stop onward transmission within the general population publicly areas, particularly where distancing isn't possible, and in areas of community transmission. The utilization of face masks has been recommended by health professionals and political authorities to scale back the danger of contagion. About 95% of the world's population sleep in countries that recommend or mandate the utilization of masks publicly during the pandemic. mask detection has turned up to be an astonishing problem within the domain of image processing and computer vision. Face detection has a variety of benefits, from facial expressions to facial expressions. In the end, the face is described with a straight face. Thanks to rapid progress in machine learning, the risks of machine detection technology

[21] seem insurmountable. Probably because of the sudden onset of COVID-19, there are now many face recognition techniques that are used for people wearing masks. In Li, Wang, Li and Fei, the author used YOLOv3. To prevent COVID-19 transmission, the products are designed with surveillance cameras and people who do not wear face masks. It is usually used for detection. The designed preliminary method combines Haar-like feature descriptors to detect the face also as key features of the face from the camera-based acquisition of a mobile phone; namely e.g detection of eyes, mouth, nose. During this research, the planning of our method conjointly employs Haar-based face feature detection techniques. It's assumed that a video selfie is taken when the camera is facing the face for facilitating the face detection. During this version of the algorithm, if a face is detected, then it's assumed that the mask-wearing is valid if the eyes are detected and if at an equivalent time the detection of the nose, the mouth and therefore the chin fails. it's assumed that the tactic analyzes faces wearing conventional masks; namely with a singular and opaque colour. The essential principle of the algorithm notably exploits the real-time detection of multiple faces at different resolutions in video streams and Haar feature-based cascade classifiers that believe. especially, the goal of Haar-like features is to code the variations of pixels contained within the image. To the present end, a little detection window, composed as an example of two adjacent rectangular zones, is positioned on the image; then the variation on this part of the image is calculated by subtracting the sum of pixel intensities resulting from the areas covered by black and white zones, respectively. The worth obtained from this calculation corresponds to an encoded Haar-like feature which will detect a texture change or the situation of a boundary within the image. The window is moved in such a way as to scan the entire surface of the image and its size increases to make sure robustness against scale variations. Moreover, several patterns are exploited on the detection window to code different types of relevant information existing within the image. To optimize the computation time of those features the usage of integral images is very recommended. away is meant for checking the right wearing of face protection mask from a video selfie. The performance of the designed method relies on the efficiency of the exploited face and face-feature detectors. The utilization of rigid masks seems preferable because they reduce possibilities of wrong positioning on the face. Moreover, this self-checking of the right mask-wearing might be exploited by monitoring-related applications as a conformity attribute.

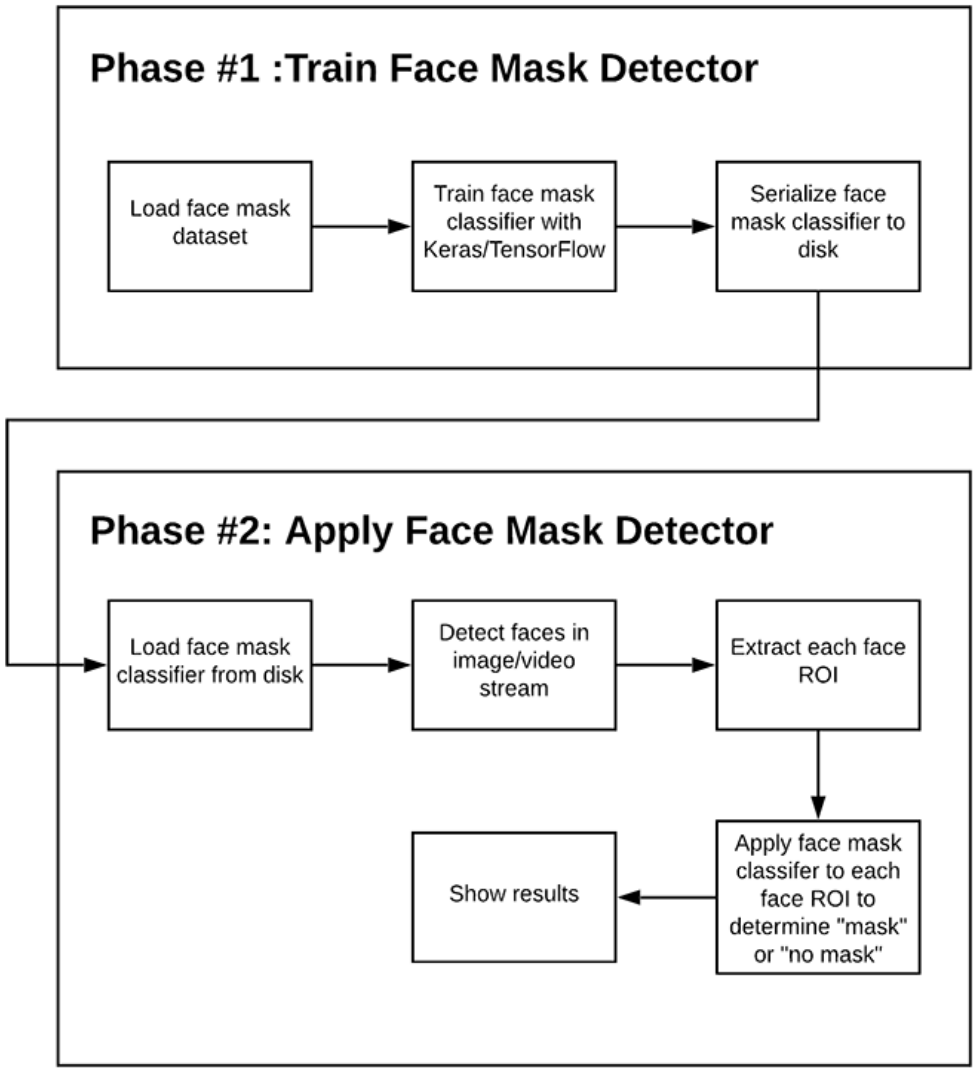


Figure 1.5: Face mask detection

We'll use the new procedure of the Python script to impart a mask detector besides review the fall outs. Specified the proficient of COVID-19 mask detector, we'll ensure to contrivance the supplementary additional Python scripts that is inclined to Detect COVID-19 face masks in the pictures, Sense the face masks in the real-time videotape running video, we'll accomplish by the surveillance of the grades of spread over our mask detector. It's a two-phased detector. Phases too individual step ladder for construction of a COVID-19 mask detector with processor vision [4] and deep learning algorithm using Python, Tensorflow, OpenCV, and Keras.

In direction to clarify a convention mask detector, our requirement to intrude our thesis into two distinct segments, every with its particular individual sub-steps:

Training: At this point, we'll specify in piling our mask recognition data set since disk, drill a model (expending Keras, TensorFlow) on this data set, formerly serializing the mask detector to the data set.
Deployment: As soon as the mask detector is proficient, we'll then spread to stacking the mask detector, carrying out face detection, at that moment classifying each appearance.

1.6 HUMAN DISTANCE DETECTION:

Human presence detection may be a range of machineries and methods intended for detecting the company of a being's body in a region of interest, or authentication that a computer, smartphone (or other device controlled by software) is operated by a person. Social distancing may be a not compulsory solution by the World Health Organization (WHO) to minimize the spread of COVID-19 publicly. Most governments and national health agencies have a traditional 2-mile distance as a mandatory avoidance in supermarkets, schools and other target areas. Monitor human detection and social isolation. During the study, models were trained on two databases, the Microsoft Common Objects database and Google OpenImage. The system is effective against three cutting-edge technologies and is based on the Oxfordtown Center database, which includes case studies of 150,000 people. It is being continuously evaluated under the optical conditions such as the concealment, the partial visibility and the low light transformation with an accuracy of 99. which is consistently used in many other areas such as driver less cars. Expression of human behavior, irritability detection, exercise, population, or other aspects of research focus on the human identification.

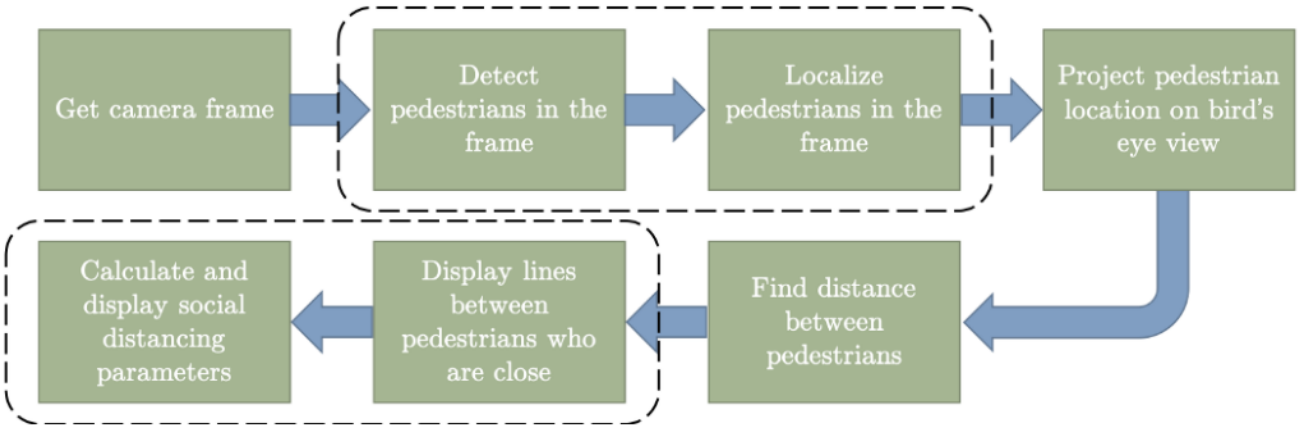


Figure 1.6: Human detection and distance between them

Evaluated a person's detection framework, for monitoring the social distancing and safety misuse within the pandemic situation. These blobs are tracked concerning the opposite to leave the space between the opposite persons [5]. They faced the challenge of detecting the person's body blobs during the outdoor area due to the correlations of other objects nearby. This journal on face recognition provides the challenges in recognizing the multi-face model and the way to beat an equivalent in future research. In order to scale back the inference speed, a small change within the R-CNN workflow [5] was made and proposed, referred to as Fast R-CNN. The modification was wiped out of the feature extraction of region proposals. Then the relevant ROI features are chosen to support the situation of the region proposals. a method is to ask the user for specific inputs resulting in a distance estimation between the pedestrians. The closer the pedestrians are to the camera the larger they're.

The closer the 2 points (which are an equivalent number of pixels apart) on the frame to the camera, the smaller is the actual distance between them. Deep Learning is employed to detect and localize the pedestrians which are then mapped to a gerrymander speedwell view projection of the camera as explained above. Once we have the coordinates of the pedestrians within the gerrymander speedwell view the social distancing parameters become straightforward. The algorithm is often wont to analyze social distancing during a public area and perform necessary actions to raise the pandemic. Automating the task will cause effective actions taken during a short time hence equipping us better to affect things. On the opposite hand, the proposed feature learning approaches are cheaper and easier because highly abstract and discriminative features are often produced automatically without the necessity for expert knowledge. The models are trained and tested on the publicly available and highly challenging UCF-ARG aerial data set.

The comparison between these models in terms of coaching, testing accuracy, and learning speed are analyzed. Experimental results demonstrated that the proposed methods are successful for the human detection task.

1.7 REAL TIME FACE DETECTION:

Here we used a real time face detection system for better accuracy. By answering some questions, we will be able to know what a real time face detection system is and how it works.

What is Real Time Face detection?

It is a basic machine learning system where tiered work is taught from multiple positive and negative images and is used to identify objects in other images in the data. This section discusses face detection, face mask detection, and human detection. The goal is to build a real-time video output system on a small computer-based collaboration platform that can recognize multiple faces in a single video series.

It is basically a machine learning based approach where a cascade function is trained from tons of positive and negative images. it's then wont to detect objects in other images present within the data sets. Here we'll work with face detection, mask detection and human detection.
The goal is to create a video face recognition system running in real-time on low computational power embedded platforms capable of recognizing multiple faces in video sequences.

Why is real time face detection difficult?

It is easy for a human brain to recognize and examine the face, but still not so easy for the computer, since human facial expressions [15] are changed by various internal and external factors such as lighting, pose variation, mustache, bear, and background colors of the glasses, Emotions, etc. and the computer cannot easily and perfectly recognize such changes. In real-time scenarios, it is difficult to obtain facial images that can be used as input to point automation or surveillance systems. Face recognition modules need to be able to find, sketch, and extract accurate faces [15] from the messy images so that case,we can provide faces as input to the face recognition module we created. Face detection module must have the following properties:

(1) It should be able to do the outline of the exact faces as soon as possible.

(2) It should be able to handle the exact borders of malfunction faces.

(3) The probability of the false face identification should be nil for a more accurate result.

(4) Restrictions of the system in the context of human interaction should be low.

(5) It should have the ability to remove the noise and the Background.

In the case of the research for face detection problems is in process and a wide range of variety of approaches has been proposed for solving the detection problems. Face detection is difficult in the video sequences as compared to the still images [15]. Video sequences are largely affected by internal and external factor changes and thus maximize the difficulties for detection modules.

Why is it advantageous?

Video face recognition has the main advantage compared with more traditional still face recognition is of using multiple instances of the same individual in sequential frames for recognition to occur.

For image recognition, the system only has one image to determine if there is anyone in the data [11]. If the image is not compatible with the output or is perfect (due to facial expressions, facial expressions, emotions), quality, or face mask), the output may be

incorrect. Of course, there are several video images that can be searched to increase the output value. Although some images are not compatible with the output, it seems. Some of them are functional and the decisions made have a high level of security. [11] Once faces are identified, tracking technology will make them easier to spot.

The disadvantage or demerits in the video imaging technique is in most cases the quality and size of the input frames are inferior compared to the still images.

How is an image captured from a video?

In reality what we see in video is not actually a video but frames being captured so fast in real time that it appears like video is being taken. Video is being captured through Video Capture objects. Then, the frame is being captured and converted into gray scale image through cvtColor function. The gray scale images are used as less information is needed to pass per image and hence, it reduces the computation power.

There are some basic processes of the architecture of a generalized VFR system:

From we know that, most of the face recognition systems for still and video image technology follow the same classical workflow:

1. The first step will be to focus at first the faces have to be detected in the images.
2. The faces are normalized to the same size and usually the same in-plane orientation.
3. A pre-processing step, the system tries to minimize the effect of illumination over the face.

4. Features are being extracted from the facial region for detection.
5. Test faces are then compared with the database of people.

The main difference between video technology and still image technology is that video mode can use a tracking algorithm with a detection algorithm to track each face in the video sequence. The use of face tracking combined with face detection has three advantages:

1. It allows the system to follow the faces across a wide range of variations in pose and lighting where tracking can be done easier than detection.

2. The time and memory requirements of a face tracking algorithm are lower than those of a face detection algorithm. Here, freed resources can be accessed once a face is detected in a frame. Tracking from that moment forward is a very important aspect when achieving real-time functionality.

3. Once a face in a particular frame is recognized with a high degree of confidence, that particular face does not need to be processed for the next frames.
4. Only track the face and keep the association between the recognized person and the tracked face.

In the classification stage of the video imagery, a history of the recognized face offers greater accuracy than that of a still image.

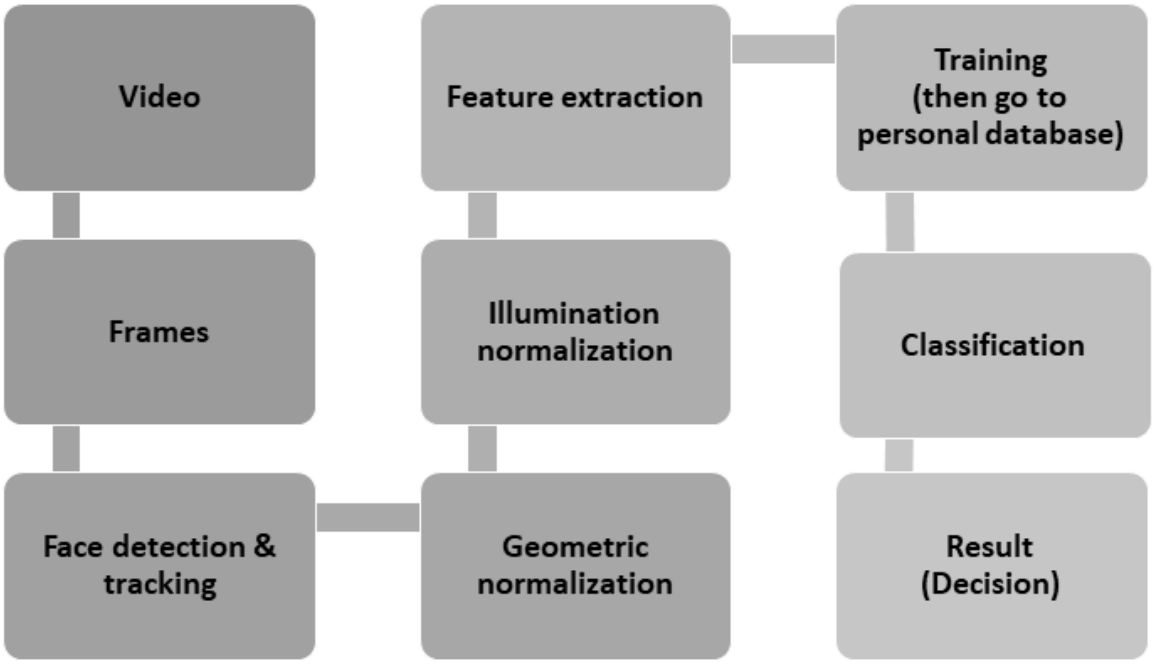


Figure 1.7: A distinctive structural design of a video face detection scheme

1.8 OBJECT DETECTION:

By answering these questions, we will be able to know what is human detection and how it works.

What is object detection?

Object detection process is a computer vision-based approach that allows us to recognize and locate objects in an image or the video. With this kind of identification approach and localization, object detection can be used to count the objects in a scene and determine and track their accurate locations, all while correctly being able to label them. Well-researched domains of object detection include face detection and pedestrian (a person travelling on foot, whether walking or running) detection.

Object detection techniques:

There are eight most effective techniques available:

- Faster R-CNN
- Histogram of Oriented Gradients (HOG)
- Region-based Convolutional Neural Networks (R-CNN)
- Region-based Fully Convolutional Network (R-FCN)
- Single Shot Detector (SSD)
- Spatial Pyramid Pooling (SPP-net)
- YOLO (You Only Look Once)

Haar Cascade classifiers are an effective way for object detection.

What is computer-based vision?

Computer based vision is an area of study which encompasses how any computers see and understand the digital images and the videos.

Computer vision involves the process of seeing or sensing a visual stimulus, making sense of what it has seen and also able to extract complex information [4] that could be used as a tool for other machine learning activities.

Applications of Computer Vision:

1. Autonomous Vehicles

2. Facial Recognition

3. Image Search and Object Recognition

4. Robotics

Now that we know what Computer Vision means and some of its application, let's dive into the implementation of it. To implement various examples of computer vision, we are going to use the OpenCV library.

OpenCV (Open-Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.

What is Haar cascade and how does it work?

This is basically a machine learning based approach where a cascade function is trained from a lot of images both positive and negative. Based on the training it is then used to detect the objects in the other images.

So, how this works is they are huge individual .xml files with a lot of feature sets and each xml corresponds to a very specific type of use case.

How to detect object from a video:

The steps are:
1. Creating appropriate Haar cascade classifier
2. Initiating video capture for video file
3. Loop once video is successfully loaded
4. Reading first frame
5.Passing frame to our appropriate Haar cascade classifier
6. Extracting bounding boxes for any bodies identified

Why are we working with object detection?

Social isolation is very important for people at high risk of infection with COVID-19. People can be infected before they become aware of the disease, so it is important to keep at least 6 people away from other people if possible, even if they are not infected. Physical distance helps limit the spread of COVID-19- [28] This means we maintain a distance of at least 1 mile from each other and avoid wasting time in congested areas.

Application of object detection:

There are several areas where we need object or human detection. They are:
1. Tracking objects
2. Face detection and recognition
3. Identity verification
4. Object extraction from an image or video
5. Activity recognition
6. Pedestrians' detection
7. Object counting
8. Automatic target recognition
9. Object detection as image search
10. Medical Imaging

Object detection has been around for years, but is becoming more apparent across a range of industries now more than ever before.

Object detection Process:

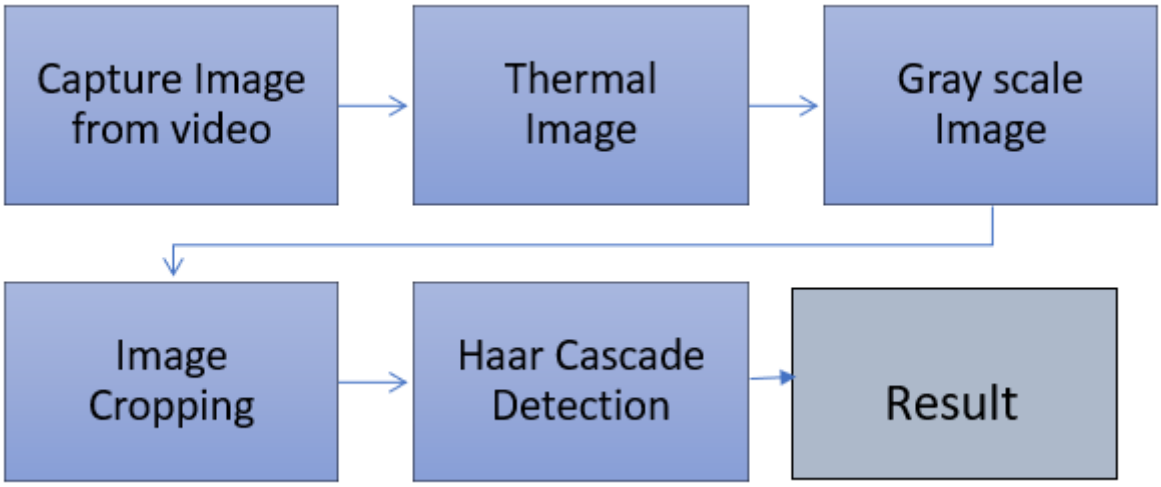


Figure 1.8: Object detection

1.9 APPLICATION OF OUR SYSTEM:

SURVEILLANCE

User satisfaction with security applications in public places, such as surveillance by face recognition systems, is poor [18]. Face recognition may be used as a method of access control to ensure that only designated persons have access to sensitive areas such as laboratories, boardrooms, bank vaults, athlete training centers, and other sensitive locations [18], and people entering and leaving those areas are required to wear masks.

Figure 1.9: CCTV camera for Surveillance system

UNIVERSITIES

University main gate also needs this type of system. So that everybody can take the safety measures and no stranger will be allowed.

BANKS

The secure rooms of a bank also need these types of safety and security measures because they contain precious things.

HOSPITALS

Some rooms of any hospital need proper safety and security also. They contain precious chemical items and personal data of patients. So, they need to protected

IMPROVE THE CONVENIENCE OF AIR TRAVEL

Face recognition and mask detection can be used by airlines to help travelers check luggage, check into flights, and board planes faster [18] while maintaining proper protection.

Mainly our system will create awareness among people to maintain safety measures and will give protection from strangers in any organization

CHAPTER TWO

BACKGROUND

We have studied the previous works and found this information. We summarized those future studies and included them here.

2.1 PREVIOUS WORKS ON THE FIELD OF FACE MASK DETECTION:

For preventing the spread of respiratory infections, face masks are being used as a type of personal protective equipment, and they might be proven to be effective at helping prevent the transmission of respiratory viruses and bacteria.

During the COVID-19 outbreak, one patient from Chongqing, China, infected five people in one vehicle when he did not wear a face mask, but no one was infected in the second vehicle he took when he did, showing the value of wearing face masks for anyone in a closed room.

Face masks and respirators for respiratory aerosols are recommended for diseases spread by droplets and can help avoid infection in public places. In a survey of influenza-vaccinated healthcare workers' attitudes toward masks, 65.7 percent of the participants decided that wearing a mask to avoid influenza transmission was a good idea.

Individuals with sub clinical or moderate COVID-19 might be able to help regulate COVID-19 through reducing the amount of contaminated saliva and respiratory droplets released into the environment. Despite the fact that the HKSAR [29] government initially only recommended people with respiratory symptoms to wear a surgical mask in compliance with WHO and the US Centers for Disease Control and Prevention (CDC) guidelines, the general population voluntarily volunteered to wear face masks from the pre pandemic to the pandemic process of COVID-19 [29].

Masks, on the other hand, are only recommended for those with implication covid-19 [19] symptoms, according to the World Health Organization, which reports that masks should be reserved for healthcare professionals. However, WHO agrees that masks worn by the general public have a place in serious pandemics since even a partial defensive effect can have a substantial impact on transmission. Under the covid-19 pandemic, there is ample circumstantial evidence to support the case for the general population wearing masks. When released in an aerosol, the virus has been shown to remain viable in the air for several hours under experimental conditions, 18 and such aerosols tend to be blocked by surgical masks in laboratory experiments. They suggest two theories, which we believe should be tested in natural experiments as soon as possible [19]. The first is that many people can be taught to use masks properly in the light of covid-19 and would do so regularly without abandoning other essential anti-contagion steps. The second

argument is that, if political will exists, mask shortages can be easily alleviated by re purposing manufacturing power, which is already taking place informally.

Ever since, a lot of progress has been developed in the field of face mask detection. In this section, we'll go through the mask detection methods in a little more depth.

2.1.1 ARTIFICIAL INTELLIGENCE BASED FACE MASK DETECTION SYSTEM:

One module is designed using AI which automatically detects a person without wearing a mask and triggers an alarm when such a person is detected. The Methodology used to create this module using Artificial Intelligence is Training an AI-based model to detect and distinguish human faces with and without masks. Then it Captures the face from live video streaming/ images with the help of an AI algorithm and Extracts relevant features from the detected portion of the face. Then by applying the trained model to identify and distinguish faces with and without a mask and activate an alarm accordingly. The module was trained using Keras for Tensorflow in python. This module will not only automate the entire process single-handedly but also would be small, compact, and easy to install. Once trained, the model is saved and when it is presented with a new set of faces via images or live video, it extracts ROI from the face and based on its learning accurately detects faces with and without a mask.

2.1.2 DEEP TRANSFERRING LEARNING (RESNET50) AND CLASSICAL MACHINE LEARNING:

For mask detection, a hybrid model mixing deep and classical machine learning is presented. The developed framework is composed of two parts. The critical aspect is used to extract features leveraging Resnet50. The second component employs decision trees, Support Vector Machines (SVM), and an ensemble algorithm to recognize face masks. There were three different types of data sets used. They're the Named Faces in the Wild (LFW) [13], the Real-World Masked Face Dataset (RMFD), and the Virtual Masked Face Dataset (SMFD). In RMFD, the SVM classifier had a testing accuracy of 99.64 percent. It accomplished 99.49 percent testing accuracy in SMFD and 100 percent testing accuracy in LFW [13].

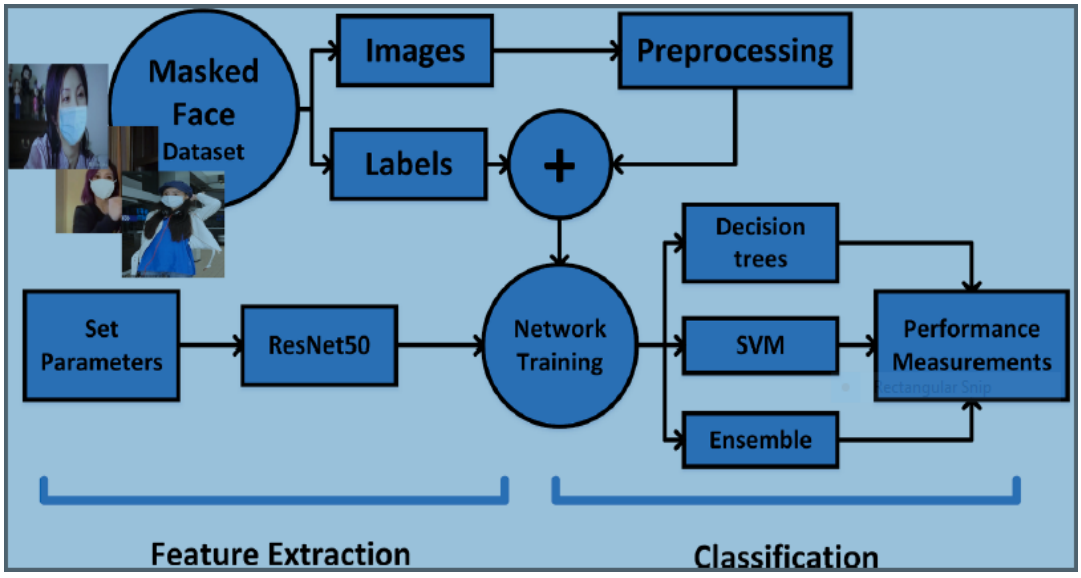


Figure 2.1: The proposed deep transfer learning model

2.1.3 FACE MASK DETECTOR:

A framework should be built to ensure that citizens obey this basic safety concept. To validate this, a face mask detector framework can be used. Face mask detection corresponds to determining whether or not anyone is wearing a mask. To detect the presence of a mask on a face, the first move is to detect the face, which splits the technique into two parts: detecting faces and identifying masks upon these faces. The working of the Single Shot Detector algorithm relies on an input image with a specified bounding box against the objects. The methodology of predicting an object in an image depends upon a very renowned convolution fashion. For the SSD, we have the VGG-16 network as our base model. VGG-16 is a very dense network having 16 layers of convolution which are useful in extracting features to classify and detect objects. To detect the face, we have used the OpenCV library. The current edition of OpenCV offers a Deep Neural Network (DNN) module that includes a pre-trained convolutional neural network for face detection (CNN). The second part of the model is trained by us using a data set consisting of images with a mask and without a mask. We have used Keras along with Tensorflow to train our model. We are planning to improve our Face Mask Detection tool and release it as an open-source project. Our software can be equated to any existing USB, IP cameras, and CCTV cameras to detect people without a mask.

2.1.4 SEMANTIC SEGMENTATION:

The order to assign a mark from each pixel in the image is recognized as semantic segmentation. The paper proposed a model for face detection in an image based on semantic segmentation by classifying each pixel as a face or non-face, effectively creating a binary classifier, and instead identifying the segmented zone. A methodology was provided for creating accurate face segmentation masks from any arbitrary size input image. For feature extraction, the method begins with an RGB image of any size and uses Predefined Training Weights of VGG – 16 Architecture. Completely Convolutional Networks are used in the training to semantically segment out the faces in the picture [22]. For training, Gradient Descent has been used, and Binomial Cross-Entropy is used as a loss function. Consequently, the FCN's output image is filtered to remove unwanted distortion, eliminate any false assumptions, and establish bounding box coordinates across the faces.

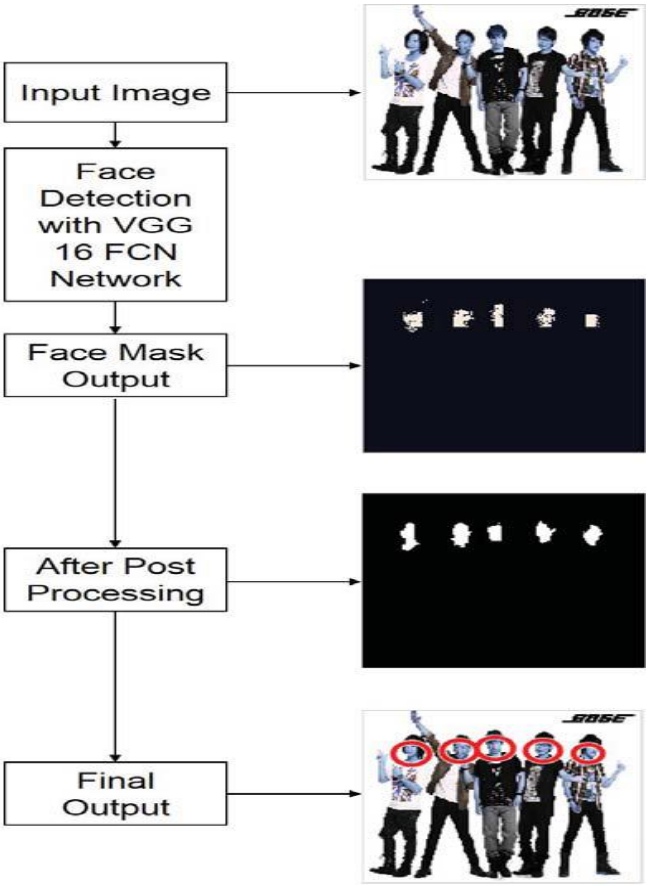


Figure 2.2: Flowchart of the proposed method

2.1.5 MULTI-STAGE CNN ARCHITECTURE FOR FACE MASK DETECTION:

In this method, they proposed a two-stage CNN model in which the first stage detects human faces, and the second stage uses a lightweight image classifier to identify the faces detected in the first stage as either ‘Mask' or ‘No Mask' faces, and creates boundary boxes into them with the identified base class. There are two portions. The first portion is to detect the face using the train set and test set. For this point, three different pre-trained models were used: Dlib, HOG-based face detector. MTCN [12] For detecting and localizing faces and facial key points, it employs a cascade architecture with three stages of CNN. Retina Face is a pixel-wise localization single-stage design which implements a multi-task instructional strategies to forecast face box, face ranking, and facial important aspects all at the same time.A CNN-based classifiers are trained for the mask detection level, based on the three image classification models: MobileNetV2, DenseNet121, and NASNet [13]. Besides that, the authors added Centroid Tracking to the framework, that enhanced its efficiency on streaming video.

2.1.6 GAUSSIAN MIXTURE MODEL (GMM):

A real-time face detection and recognition proposed approach based on GMM in the field of financial anti-fraud is proposed in this procedure. It's also divided into three sections: 1) Detecting and extracting a face in photographs using Haar classifiers in OpenCV. 2) A GMM is trained mostly on feature vectors of the human face for a large number of images. 3) The deep learning algorithm's dlib library is then used to identify and create face features once more.

2.1.7 DEEP LEARNING- BASED SAFE SOCIAL DISTANCING

They use Single Shot Detector MultiBox (SSD) in this proposed model, which appears to be a reasonable option for real-time object recognition with a slight accuracy trade-off. To find valuable image features, SSD uses the VGG-16 model, which has been pre-trained on ImageNet. SSD [4] adds several convolutional layers of decreasing sizes to the top of VGG16. The approach for effective aids in the protection of people in a public place by automatically detecting whether or not they maintain a safe social distance and whether or not they are wearing a face mask. For the very first task, deciding the necessary distance of two people During the retraining of the architecture, a very low learning rate is used to ensure that the previously trained convolutional filters may not deviate dramatically, and tests have also been conducted out with OpenCV, TensorFlow inspects the secure social gap among observed individuals and face masks detection in real-time video streams using Deep Learning and [4] Computer Vision. A face mask

detection model with variation in the styles of face masks used by the public would be sent to a pre-trained SSD face detector in real-time using a transfer of learning for the mask detection section. The Raspberry Pi 4 Model-B [1] is used as the chosen edge unit, with an ARMv8 1.5 GHz processor and 4 GB of RAM. The camera is connected to the Raspberry Pi4 and the trained model with the custom data set is mounted. The camera sends real-time video of crowded locations to the Raspberry Pi4 model, which tracks public spaces constantly and instantly, detecting whether people maintain secure social distances and whether they wear masks.

2.1.8 RETINA FACEMASK:

Retina Face Mask combines elevated semantic information using several feature maps and a feature pyramid network (FPN). It suggested a background attention detection head and often a cross-class object removal algorithm to boost detection capabilities. Besides that, because the face mask database is a small data set with difficult-to-extract attributes, they employed transfer learning to convert learned kernels through networks trained for a common face detection task on a huge database to the face mask data set.Faces with masks, faces without masks, faces with and without masks in one image, and confusing images without masks are among the masked or unmasked face images shown in the data set.Experiment findings indicate that [8] Retina Face Mask achieves state-of-the-art results in face and mask detection precision, respectively, of 2:3 percent and 1:5 percent higher than the baseline score, and 11:0 percent and 5:9 percentage points higher than the longitudinal outcome in recall [8].

2.1.9 FACE MASK DETECTION USING TRANSFER LEARNING OF INCEPTION V3:

The primary objective of this research is to recognize people who do not wear masks in crowded locations in order to prevent COVID-19 from spreading further. So, using the SRCNet classification network, a specification for a face mask recognition system was introduced, with a precision of 98.7% in categorizing the images through three categories: appropriate face mask wearing, incorrect face mask wearing, and no face mask wearing. Transfer learning is used to detect the mask, that enables the learned information of the neural network consisting of para-metric weight to be passed to the new design. Even when the new classifier is developed on a small data set, training data enhances its performance. As a consequence, an automatic model that utilizes InceptionV3's transfer learning to identify people who aren't wearing a mask is developed. The model's performance for a random sample shows the bounding box around the face, with green and red color indicating whether or not a person is wearing a mask, however, as well as an approval rate of over 90%. Image augmentation strategies are used to boost the model's efficiency by increasing the variety of the training data. On the SMFD data set, the proposed transfer learning model achieved 99.92 percent accuracy

and 99.9% septicity during training [8], and 100 percent accuracy and 100 percent septicity during testing.

2.1.10 FACE MASK NET DEEP LEARNING NETWORK:

This project distinguishes face masks from images and live video streams.
It is separated into 2 phases: face mask detector preparation and face mask detector implementation. In the training process, the data set is loaded for the model to be trained, and the model is serialized. The training set is then loaded, faces in images and video streams are identified, and the region of interest (ROI) is retrieved. Eventually, the face mask detector is used to identify the images or faces in the streaming video as wearing a mask, incorrectly wearing a mask, or not wearing a mask. The data set contained 15 images of masks that had been worn wrongly, 10 masked images, and 10 images without a mask. Our face mask detector model was developed using MATLAB programming. They used Face mask Net architecture to train the model, which succeeded in a precision of 98.6%.

CHAPTER THREE
ADOPTED METHOLODOLOGY

We have designed our own Haar cascade here and compiled three individual projects. Our main purpose is to build an overall system for security surveillance.

3.1 OVERVIEW OF OUR PROPOSED METHOD

In Python, we built a user interface with three sections for detecting three different features. It will

1. Recognize the identity of an individual
2. Make sure they are wearing a mask properly, if not it will ask the person to wear a mask
3. Make sure people in the frame are maintaining social distance by capturing real-time video

To implement this application of computer vision, we have used the OpenCV library and Haar Cascade classifier.

We wanted the framework to have a good user interface, so we developed a graphical user interface (GUI) that allows us to communicate with the code and enhances our command-giving experience.

Using the GUI, if we press the face recognition button, it will identify people based on the data set using Haar cascade classifiers. All we have to do now is train a classifier that can take measurements from a new test image and tell us which known individual is the most similar. This classifier will recognize the person's name and show the result after it has been run.

If we trigger the mask detection function, it will decide whether or not the person is wearing a mask. Each function of a Haar cascade is checked against an input image to make detection.

In the beginning, the decision thresholds for the features are initially tiny, meaning that some faces with masks and some other things will be detected as well. It will work more efficiently if we keep improving stages.

Pressing the human detection button will ensure social distancing. It will detect how many people are in one frame per time elapsed so that the social distancing can be ensured.

3.1.1 OpenCV

OpenCV (Open-Source Computer Vision Library) is a BSD-licensed open-source computer vision library with hundreds of algorithms. It is

1. A cross-platform library, available for both Android and iOS.
2. An Open-source computer vision library

3. Has more than 2500 optimized algorithms
4. Written natively in C++

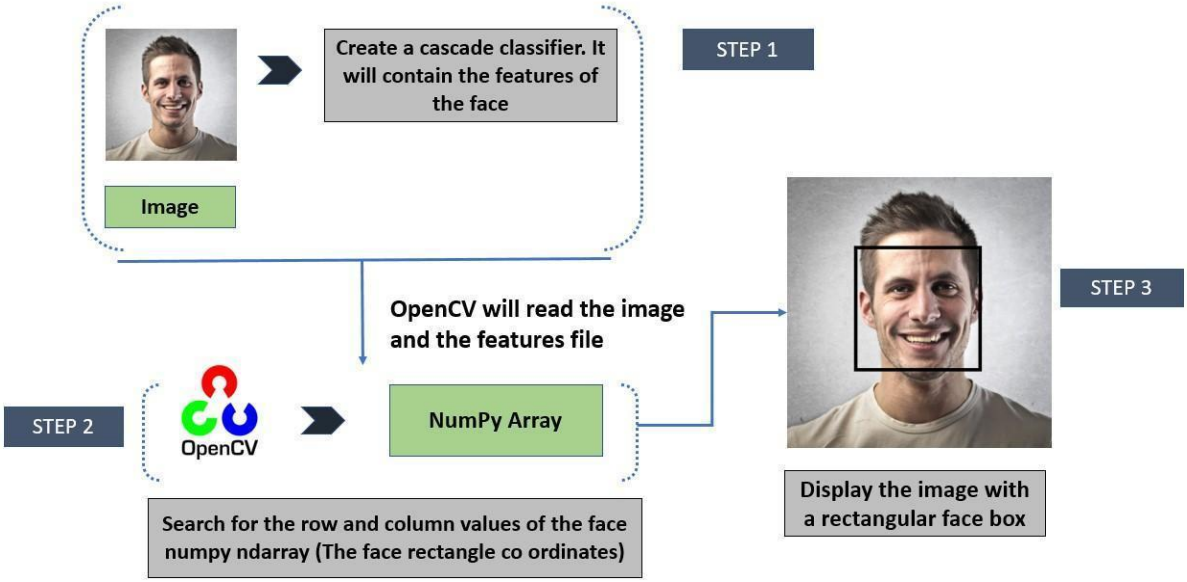


Figure 3.1: How OpenCV works

This is how an OpenCV works with a cascade classifier. To make a cascade classifier, you'll need a picture. It will have the facial features. The picture and features file will then be read by OpenCV. It will look for the face NumPy ndarray's row and column values. Then it will display the image with a rectangular face box.

3.1.2 HAAR CASCADE

The Haar Cascade classifier is a machine learning-based technique that uses a large number of positive and negative images to train it. It is then used to detect the objects in the other images, depending on the training. It is a very powerful tool for detecting objects.

Positive images – These images include the images that our classifier is expected to recognize.
Negative Images – There are photos of everything that isn't the object we're looking for. We need positive and negative images to train the classifier [23]. We can define how many stages it will contain and that will have an impact on accuracy level. Then the XML file will be generated.

As long as we have the required XML file, we can use the Haar Cascade classifier to detect any form of object. Using the cascade trainer GUI, we can also build our own XML files to detect any type of object we want. They're large.xml files with a lot of features, and each one corresponds to a very specific form of use case. These are the basic three features of the Haar cascade classifier. They are 1. Edge 2. Line and 3. Center-surround features.

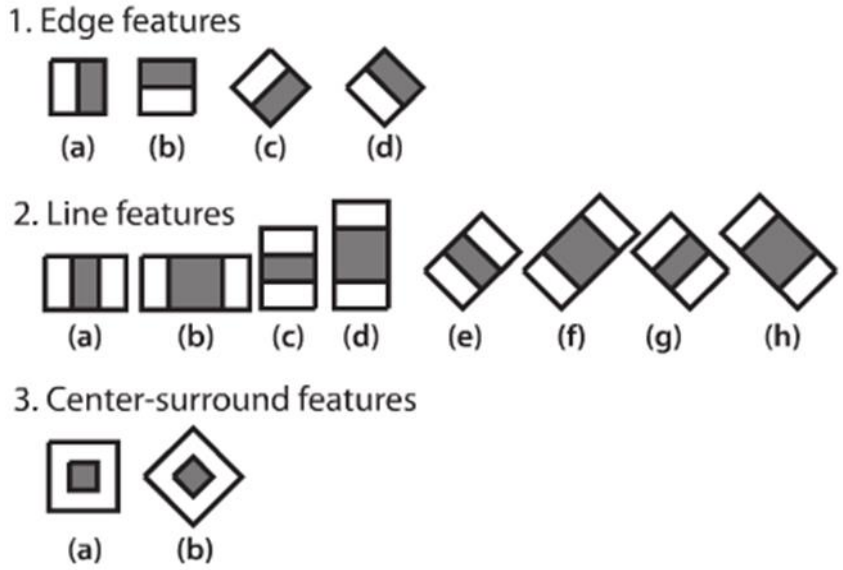


Figure 3.2: Features of a Haar cascade classifier

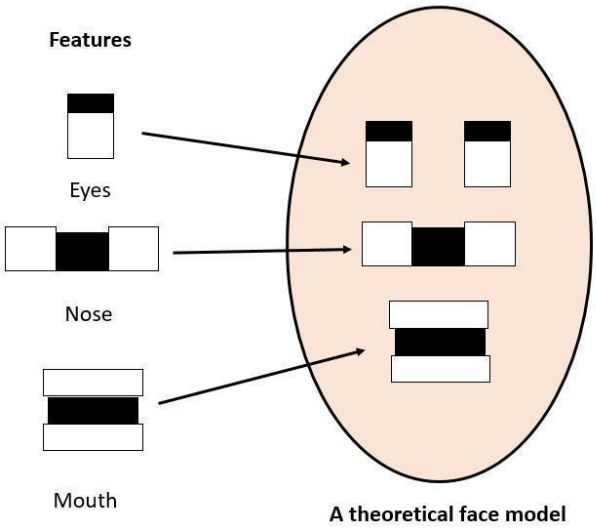


Figure 3.3: A theoretical face model

The above figure shows a theoretical face model containing eyes, nose, and mouth Haar features. By adding more features or classifiers like eyebrows distance, the distance between lips and nose, we can increase the accuracy level of the Haar cascade.

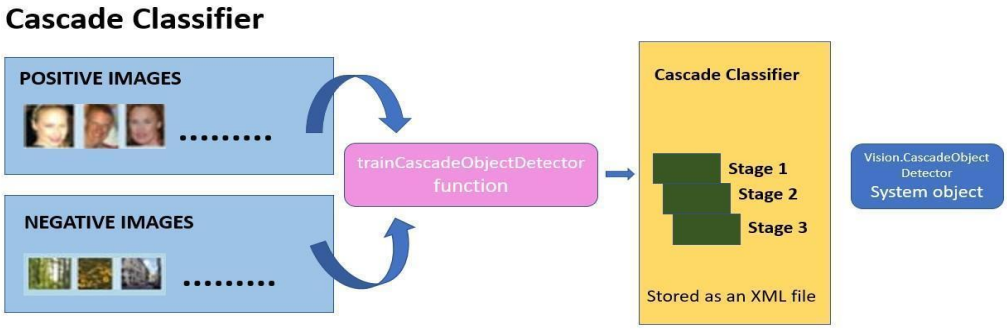


Figure 3.4: How Haar cascade classifier works

Negative and positive images are trained using a function named train cascade object detector. After going through several stages, the Haar cascade classifier is stored as an XMLfile.

3.1.3 POSITIVE IMAGES

We use a collection of negative and positive images to train a Haar cascade classifier. Positive images are those that we want our classifier to recognize. To create a positive image data set, we have to use the photos of the objects we are interested to detect. All the positive sample object regions should have about the same aspect ratio. The number of positive images should be at least double the negative images.
To start the training, we needed to create a folder within that One folder should be “p” (for positive images).

3.1.4 NEGATIVE IMAGES

Negative images are those that don't include the object we're looking for. We need positive and negative images to train the classifier. To create a negative image data set we have to use the photos of the objects we do not want to detect. The number of positive images should be at least double the negative images.
To start the training, we needed to create a folder within that One folder should be “n” (for negative images).

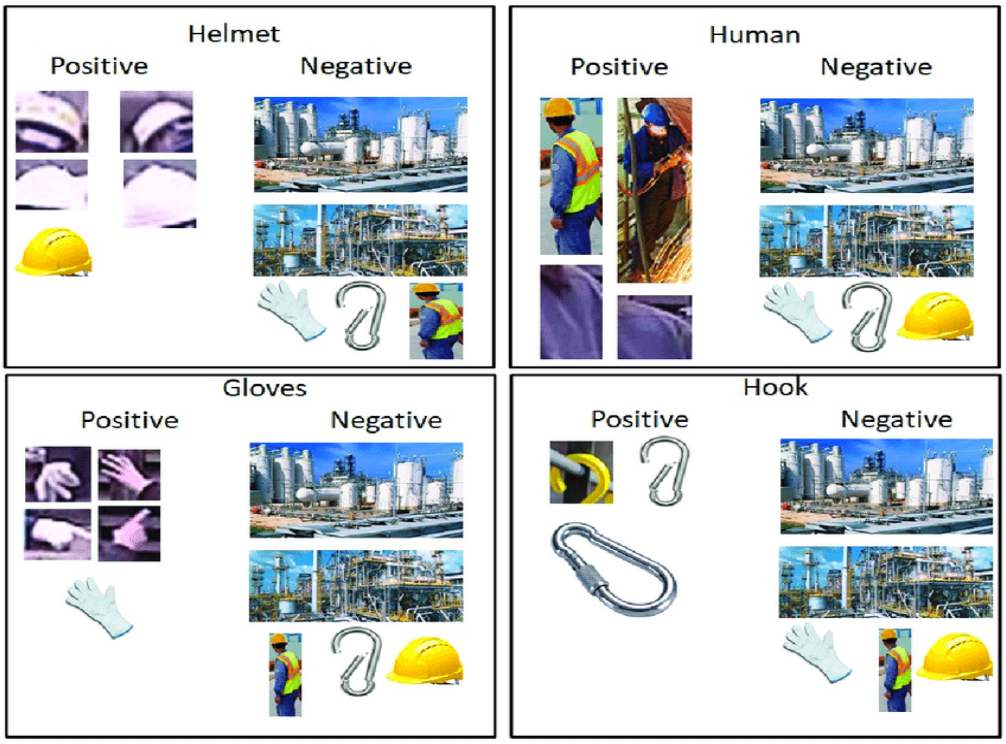


Figure 3.5: Examples of positive and negative images

3.2 GUI

3.2.1 WHAT IS GUI

A graphical user interface, also known as a GUI, is a way of displaying interactive visual components for computer applications. It represents behavior that the user may take by showing objects that transport information. The objects change color, scale, or visibility when the user interacts with them.
In contrast to a text-based command-line interface, such as MS-DOS, or the shell of Unix-like operating systems, a GUI is thought to be more user-friendly. Unlike command-line operating systems, such as Unix or MS-DOS, GUI operating systems are easy to learn and use because commands are not memorized. Furthermore, users are not expected to be familiar with any programming languages. GUI operating systems have come to dominate today's market due to their ease of use and more modern look.

A pointing device, such as a mouse, is used to communicate with almost every part of the graphical user interface. A touch screen is utilized in more modern (and mobile) devices. Also, it is feasible to navigate a GUI using a keyboard.

3.2.2 HOW DOES A GUI WORK

A graphical user interface (GUI) uses windows, icons, and menus to provide commands such as opening, deleting, and moving files. While a mouse is used to control a GUI operating system, keyboard shortcuts or arrow keys may also be used. A graphical user interface (GUI) enables a computer user to communicate with it by navigating a cursor around a screen and pressing a button. A pointer can be moved around the screen in many ways.

Setup phase:

- Describe how you want the GUI window to look
- For windows, widgets, and layout, use libraries.
- Insert advanced code to be used later.

Personalization (provided during setup)

- Modern widgets that show themselves in a number of ways
- How to Respond to Situations

Putting it into effect

- The mechanism receives events from the operating system.
    - The window appears, the mouse clicks, the keypresses, and so on.
- In answer, the system executes application code.
    - The aforementioned customization

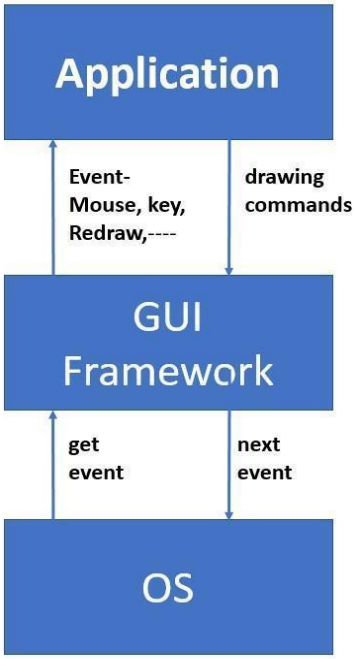


Figure 3.6: How GUI works

3.2.3 Tkinter

The Python binding for the Tk GUI toolkit is Tkinter. It is Python's de-facto standard GUI and a standard Python interface to the Tk GUI toolkit. Tkinter is made up of Python builds for Linux, Microsoft Windows, and Mac OS X.

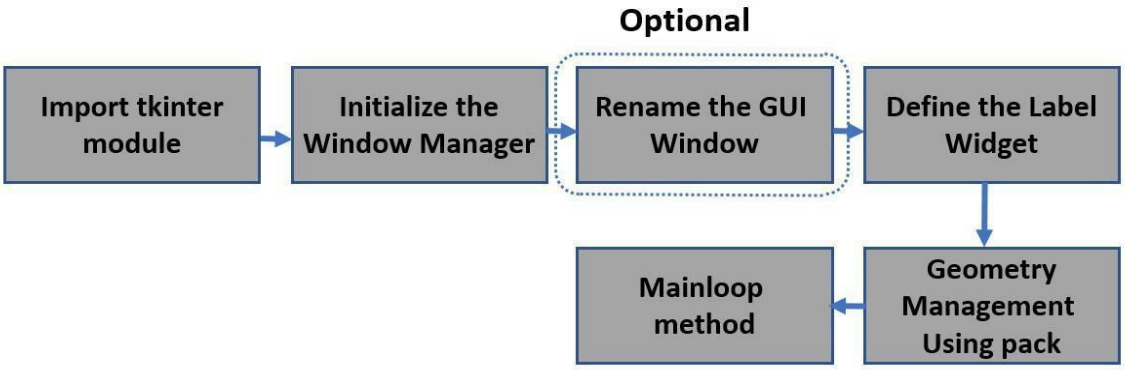


Figure 3.7: Tkinter module

From the flow diagram of this graphical user interface, it can be seen that first, we need to import the module named Tkinter module. The next move is to use the Tkinter method to configure the window manager and assign it to a variable that generates a blank window. Then we added labels and buttons as required. Finally, we used the key loop () method to hold the window open before we closed it manually. In the backend, it runs an infinite loop.

3.2.4 BUTTON

The Button widget is a basic Tkinter widget [30] that can be used with various types of buttons. A button is a type of widget that allows the user to interact with it. When the button is pressed with a mouse click, an operation is initiated. It can also be used to store text and pictures, such as labels. A button can only display text in one font, while labels can display text in multiple fonts. A button's text can be more than one line long [30].

A button can be connected to a Python function or process. Only if the button is pressed in any way will this feature or process be called.

3.2.5 USE OF THE GUI IN THE SYSTEM DESIGNED USING TKINTER

We designed a GUI button using the Tkinter module. First, we added the Tkinter module to our project. The next move was to use the Tkinter method to configure the window manager and assign it to a variable that generates a blank window. Then, as needed, we added labels and buttons. Finally, we used the key loop () method to hold the window open before we closed it manually. The back end was trapped in an endless loop. When we click the Face recognition button, it will identify people based on the data set using Haar cascade classifiers. Upon clicking the mask detect button, it will be ensured whether someone is wearing a mask or not. And the human button will detect how many people are in one frame and thus ensure social distancing.

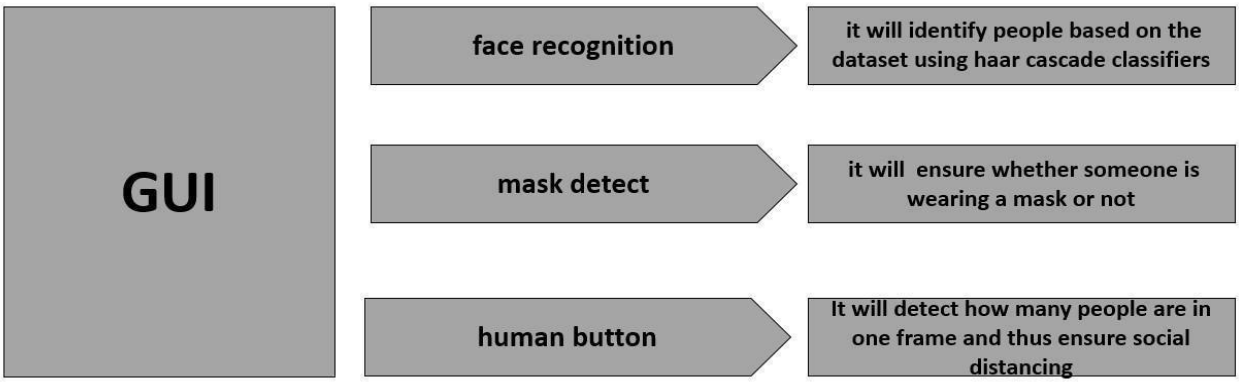


Figure 3.8: The GUI designed for our system

3.3 FACE RECOGNITION

Facial recognition is a progressing expertise that's typically rummage-sale for security determinations, but now encompasses beyond safekeeping to promoting and augmenting software user involvement. Face recognition is the process of recognizing and checking individuals in a photograph based on their faces. Even when faces are reformed by age or blocked with accessories in addition to facial hair, it's a task that humans can easily

complete, even in varying well-lit conditions and when faces are reformed by age or blocked with accessories in addition to facial hair. Deep learning procedures may take advantage of large datasets of faces to construct rich and compact illustrations of faces, enabling current prototypes to work well at first and then abandon their face recognition capabilities. We will find the faces in a picture and comment on who the individuals are, if they're known. We will do that alright, like when the people are elderly, are wearing shades, have different highlighted hair, are beholding in several directions, and so on.

3.3.1 FACE RECOGNITION BASIC IDEA

Face recognition is a technique for recognizing or confirming an individual's identity by looking at their face. This device can recognize people in pictures, videos, and in real time. During police stops, officers can also use mobile devices to identify individuals.

What is the mechanism of facial recognition?

1. Image is needed to be captured
2. The positions of the eyes are determined.
3. After that, the image is converted to grayscale and trimmed.
4. The image is turned into a prototype that the search engine uses to produce facial comparison results.
5. To compare the template to other templates on disk, a sophisticated algorithm is used to search and match the image.

Face recognition systems use computer algorithms to recognize unique, recognizable features on a person's face. These details, such as the distance between the eyes or the shape of the jaw, are then translated into a mathematical rendition and compared to data from a face recognition database. The information about a specific face is usually referred to as a face template, and it is visible in photos since it is intended to only contain some data that can be used to distinguish one face from another.

Rather than positively identifying an unknown individual, some face recognition systems measure a likelihood match score between the unknown person and certain face models stored in the database. Rather than returning a single result, these systems would present a list of potential matches, ordered by probability of correct recognition.

Face recognition methods differ in their ability to identify individuals in difficult circumstances such as inadequate lighting, low-quality picture resolution [18], and a suboptimal angle of view, such as when looking down on an unknown person in a photograph taken from above.

There are two key concepts to understand, when it comes to errors:

False negative: When a face recognition device fails to fit a person's face to a picture in a database [18]. In other words, the machine would respond to a question with zero results, which is incorrect.

False positive: When the face recognition system finds a match to a person's face with an image in a database, but in reality, the match is incorrect.

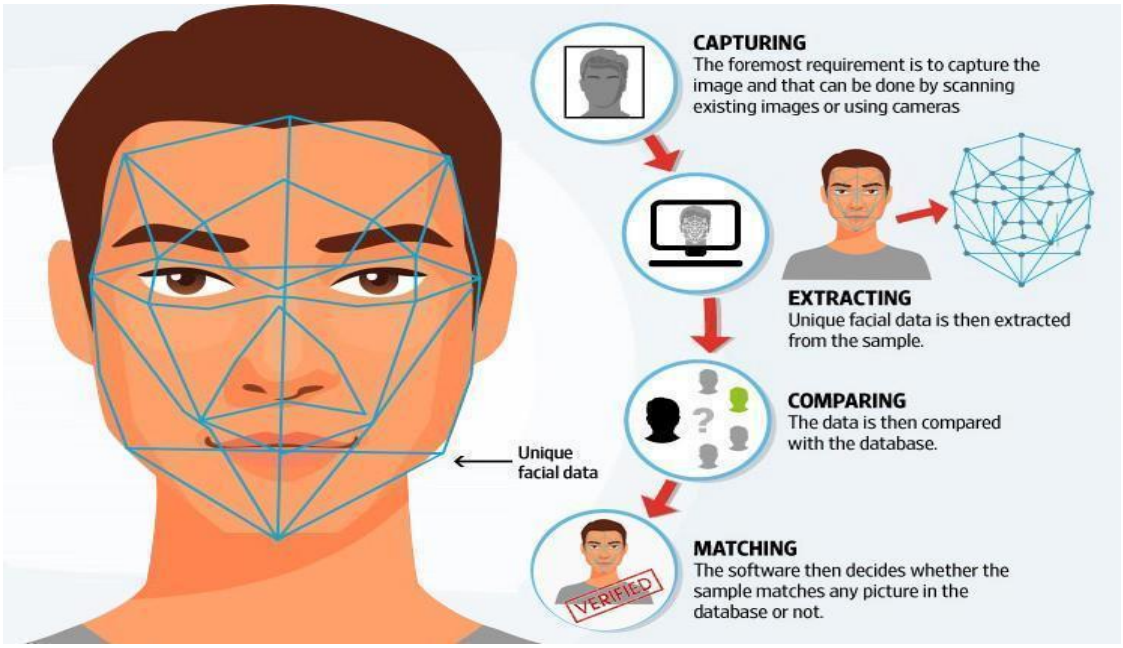


Figure 3.9: Face recognition - how does it work?
[Source: https://www.shadowsystem.com/page/20]

In a face recognition device [20], the "false positive" and "false negative" frequencies must be closely controlled, since there is almost always a trade-off. When using face recognition to unlock our phone, for example, it is preferable for the system to misidentify us a few times (false negative) than for the system to misidentify other people and allow them to unlock our phone (false positive).

3.3.2 OpenCV IN FACE RECOGNITION

OpenCV is an image and video processing library that can be used for a range of tasks including facial recognition, license plate interpreting, photo editing, advanced robotic vision, optical character recognition, and more. The Single Shot Detector (SSD) architecture with a ResNet base network is used by OpenCV for deep learning face detection.

How OpenCV works with a cascade classifier:

It takes an image to create a cascade classifier. It contains the features of the face. Then OpenCV reads the image and features file. After that, it looks for the face NumPy ndarray's row and column values. Then it displays the image with a rectangular face box.

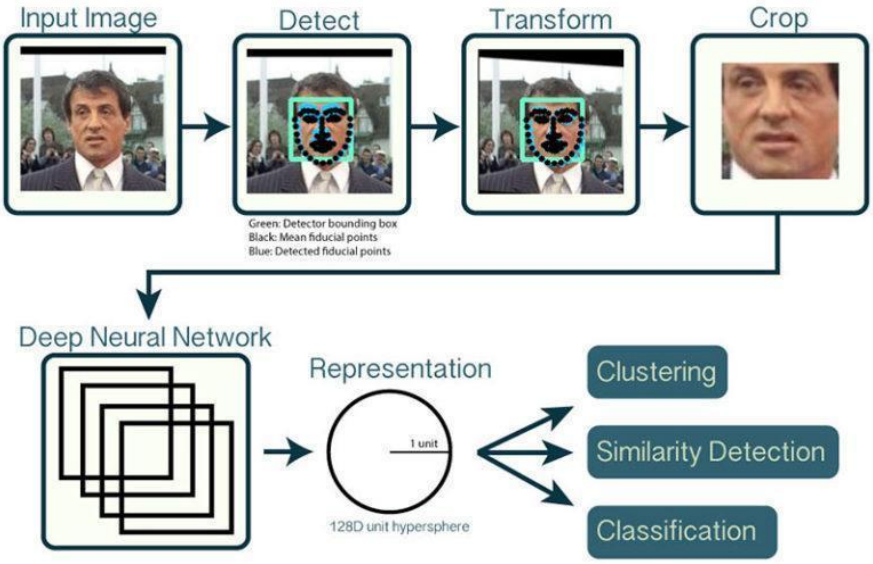


Figure 3.10: OpenCV in facial recognition system

OpenCv is the most popular library for face face recognition because it is a cross-platform library, available for both Android and iOS, an Open-source computer vision library, has more than 2500 optimized algorithms and written natively in C++.

3.3.3 HOG METHOD

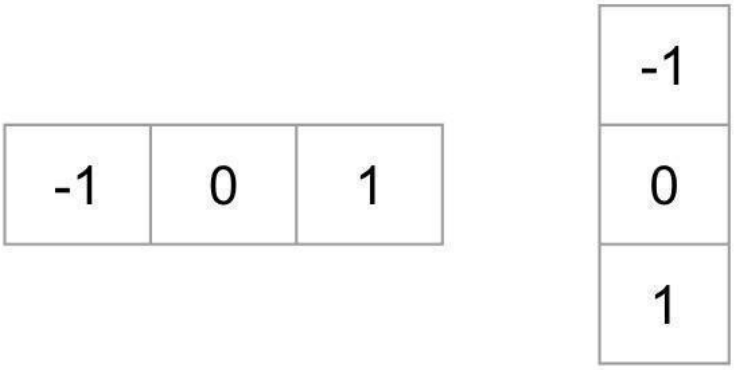
The Histogram of Directed Gradients, or HOG, is a function descriptor used in object detection in computer vision and image processing. This method involves counting instances of gradient orientation in specific areas of an image.

Preprocessing

A 64 128 patch of an image is used to measure the HOG function descriptor for pedestrian detection. We all know that an image may be any size, but patches at different sizes are usually evaluated at multiple image locations. The only problem is that the patches under consideration have a fixed aspect ratio.

Calculating the Gradient Images

After all, we want to measure the histogram of gradients to calculate a HOG descriptor [27], so we need to calculate the horizontal and vertical gradients first. Filtering the picture with the kernels below is an easy way to achieve this.



Then, using the following formula we can find the magnitude and direction of gradient.

$$g = \sqrt{g_x^2 + g_y^2}$$
$$\theta = \arctan \frac{g_y}{g_x}$$

This HOG approach calculates how dark a given pixel is in contrast to its immediate surroundings. Then draw an arrow [27] indicating the direction in which the picture is darkening. Gradients are the arrows that display the transition from light to dark throughout the entire picture.

Calculating Histogram of Gradients in 8×8 cells

The image is divided into 8x8 cells in this step, and a gradient histogram for each 8x8 cell is determined.
The 9-bin histogram is generated by adding up the contributions of all the pixels in the 808 cells.

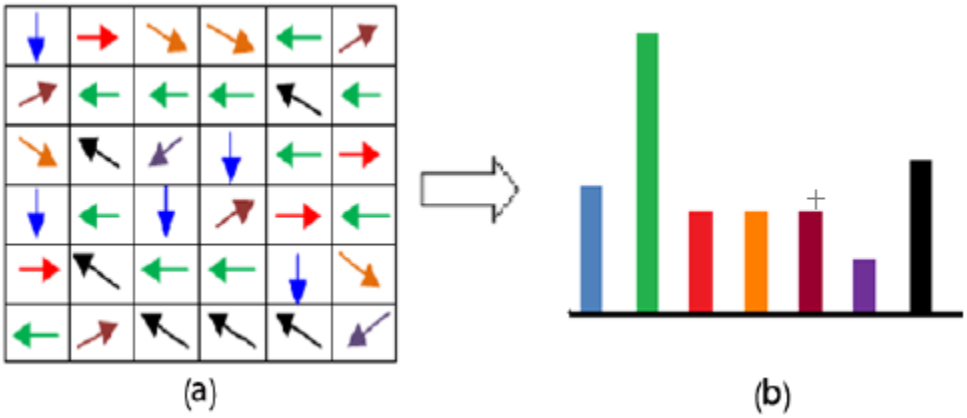


Figure 3.11: HOG calculation

[Source:https://www.researchgate.net/publication/269074001_Texture_classification_using_dominant_gradient_descriptor/figures?lo=1]

16×16 Block Normalization

A histogram based on the image's gradient was generated in the previous stage. We need to normalize that into 16x16 blocks in this stage.

The overall lighting of an image has a major effect on the image's gradients. If we divide all pixel values by 2 to make the image darker, the gradient magnitude will change by half, and the histogram values will change by half. In general, we'll want our descriptor to

be unaffected by lighting changes. To put it another way, we like to “normalize” the histogram so that it is unaffected by lighting changes.

Calculating the HOG feature vector

By defining the key landmarks in the face, we can decide the pose of the face. Every face has 68 landmarks that we can recognize. Once we've found certain landmarks, we can use them to warp the picture to focus the eyes and mouth. Then we have to do Face Encoding. First, we must run the oriented face image through a neural network that understands how to quantify facial features. After that, we must save those 128 measurements. All that matters is that the network produces virtually identical numbers when comparing two photos of the same human. Finally, in order to determine the person's name, we must decrypt the encoding. A simple linear SVM classifier will be used. Examine all of the faces we've already weighed to see whose measurements are the closest to ours.

3.3.4 FACE RECOGNITION IN OUR SYSTEM:

From the GUI, when clicking the face recognition section, it identifies people based on the data set using Haar cascade classifiers.

Figure 3.12: Face recognition method

First, we need to find all the Faces. To do so, we'll need to Encode an image using the HOG (Histogram of Directed Gradients) algorithm to construct a simplified version. This HOG approach calculates how dark a given pixel is in contrast to its immediate surroundings. Then draw an arrow to indicate which way the picture is darkening. Gradients are the arrows that display the transition from light to dark throughout the entire picture. Now we need to divide the image into tiny 16x16 pixel squares. We count how many gradients point in each main direction in each square. Thus, we can turn the original image into a very simple representation. Find the part of the image that most closely resembles a generic HOG encoding of a face using this simplified image.

Posing and Projecting Faces are needed after that. To begin, we must define the key landmarks in the face in order to determine the pose of the face. Every face has 68 landmarks that we can recognize. We use certain landmarks to warp the image so that the eyes and mouth are in the middle.

Then we have to do Face Encoding. First, we must run the oriented face image through a neural network that understands how to quantify facial features. After that, we must save

those 128 measurements. All that matters is that the network produces virtually identical numbers when comparing two photos of the same human.

Finally, we must decode the encoding to decide the person's name. A simple linear SVM classifier is used. Look at all of the faces we've already measured to see who has the most similar measurements to ours. That's how we meet our soul mate.

3.4 THE DETECTION OF MASK IN THE SYSTEM

After the successful completion of face detection, comes the mask detection process. There are few steps to detect masks in the system. In any advanced security system, it is essential to identify the security as well as maintain the necessary security precaution for mask detection. To detect the mask in our analysis, we used the Haar cascade trainer GUI. It has spread rapidly across the world, presenting major health, economic, environmental, and social problems for the entire human population.
Maintaining social isolation is another significant factor in stopping the virus from spreading. When an infected person coughs, sneezes, or speaks, the infection spreads.
Droplets from the infected person's mouth or nose float through the air and land in the mouths or noses of others who are close by. In addition, the droplets can be inhaled into the lungs.
According to recent reports, people who are infected but do not display symptoms are likely to contribute to the spread of the disease. So, in this system, this is operated by a system designed graphical user interface.

3.4.1 THE DETECTION OF MASK: AN OVERVIEW

For mask detection, we will add an infinite while loop that will repeatedly grab the images from the stream, and search for faces with the Haar Cascade Classifier. If the Haar Cascade matches a face, the pre-trained model will predict whether the person is or is not wearing a mask, and the cycle repeats. If the Haar Cascade does not match a face, it will show that no face was found in the frame. Cascade Trainer Interface, a software for teaching, evaluating, and developing cascade classifier models, was used to build the Haar cascade classifier. When we go to the face recognition section of the GUI, it will use Haar cascade classifiers to classify people based on the data set. All we have to do now is train a classifier to take measurements from a new test image and decide which known individual is the most similar. The mask detection function will decide whether or not the person is wearing a mask.

3.4.2 THE STEP-BY-STEP APPROACH USING HAAR CASCADE

Haar Cascade is a machine learning-based technique that uses thousands of positive and negative images to train a cascade function. The primary attribute chosen tends to reflect on the fact that the area around the eyes is typically darker than the area around the nose and cheeks. The second attribute chosen is based on the fact that the eyes are darker than the nose bridge. The bulk of the picture area in a photograph is non-face. As a consequence, having a simple way to say whether a window isn't a face region is a much better idea. Apply the second stage of functionality and resume the process if it passes. For example, when it comes to a person's face, it's common knowledge that the area around the eyes is darker than the area around the cheeks. As a result, a typical Haar function for face detection could be a pair of adjacent rectangles above the attention and, thus, above the cheek area. The rectangles' positions are specified in relation to a detection window that acts as a bounding box for the target object. This technique is based on two algorithms: a Haar-like image descriptor and a classifier style artificial neural network.

3.4.3 THE TRAINING METHODOLOGY USING HAAR CASCADE FEATURE

There are several methods of training a Haar cascade. one of them provides an in-depth discussion on what we would like to make cascade classifiers supported Haar-like features, and that is that the commonest system in computer-vision designed for face and eye recognition. The positive images are those that contain the thing, while the negative images are those that do not. Having a supplementary quantity of positive in addition to negative images will customarily cause an additional precise classifier. we prerequisite this negative file for the preparation of the classifier. During this step, we would like to make a knowledge folder that encompasses the labels of positive images also because of the location of the objects in the apiece image. Since the first is simpler and quicker, the second could be more adaptable but take longer to work out. Positive images are in the dossier, and there is a file for the pattern of the objects in positive images in the document. Notice that two directories, cv.dll and highgui.dll, must also reside in the current directory for objectmaker.exe to run properly. Before we run the objectmaker.exe, make sure we're comfortable and have enough time to label and crop tens or hundreds of photos! When we run the file objectmaker.exe, we get the following two windows: one illustration the loaded image, and therefore the other single shows the image tag. If we use the opposite two corners, objectmaker.exe will not write the selected objects synchronizes to the data.txt file.c- If we're satisfied with the rectangle we've selected, the location and size of the rectangle will appear in the left pane. If we employ the opposite two corners won't write the coordinates of the chosen object into the data .txt file.c- If we're proud of the chosen rectangle. The location and size of the rectangle will then

appear on the left pane. We should have catalogues after we finish the Haar-training stage, where N is the number of steps we've already identified. Now we should always merge all bent stages into a single XML file, which will be our final file, a cascade of Haar-classifiers.

Another alternative is to because of the particular nature and difficulty of this mission, it is taking a little longer than normal, but the payoff is enormous. Thus, use it every day, take the files we'd like, then terminate the server and pay a very little bit of cash. So, if we want to make a Haar Cascade, we'll need both "positive" and "negative" pictures. The "positive" images are images that contain the thing we'd wish to hunt down. this may either be images that just mainly have the thing, or it is often images that contain the thing, and we specify the ROI where the thing is. One nice thing about the positives is that we simply can't just have one image of the thing we'd wish to detect, then have a few of thousand negative images. From here, alongside our single positive image, we'll use command to create a bunch of positive examples, using our negative images. If we are looking to identify all screwdrivers, however, we'll be wanting to possess thousands of unique images of screwdrivers, rather than using the urge samples for we. In our case, we might like watches, so search for watches, and we will find many categories of watches. I found the sports/athletics link to possess a reported 1,888 images, but we will find plenty of those are broken. Well, first, we want all of these to be the same size, and a whole lot smaller! For positives, this file could also be a huge pain to form manually, since we'd wish to specify the precise Region of Interest for our object, per image. We just need a simple descriptor for the negatives, but that's no problem, we'll do that while we pull and manipulate the photographs. As we'll likely gather from that last sentence, the results of this command are indeed the great, and holy, cascade file.

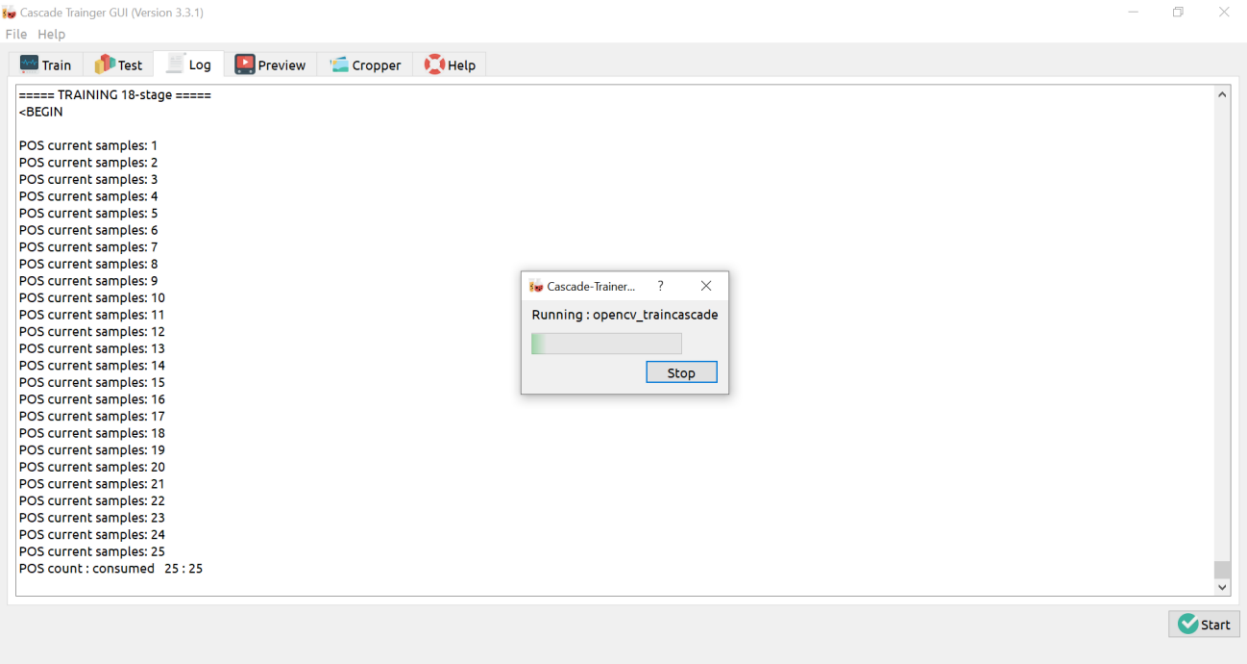


Figure 3.13: Training Haar cascade using positive images

Another viable alternative is to use the Haar cascade Interface. It uses a graphical interface to align the bounds and make it simple to use OpenCV software designed for drilling and checking classifiers. This is often the first show, and it can be used for teaching classifiers to teach classifiers. We may wish to include thousands of positive and negative image samples for ease, but there are times when we will do so with fewer. To begin the training, we will want to create a folder specifically for our classifier. Positive likeness samples are photos of the item we'd like our classifier to detect as well. If we want to coach and detect vehicles, for example, we'll need a large number of car pictures. Negative images are frequently images that are not the same as the positive image, but in reality, negative images must be complementary to the positive image. Following that, we may want to line the sample's width and height. If we have sample images, for example, we must determine the ratio, which in this case is. We'll choose one of the following for Input Settings, and we'd like to line the trial up according to this option: Due to the scene in which detection will be conducted, only one image will be used. Path should refer to at least one image file in this scenario. Only supported images are often selected. During this case, Path needs to be set to a dossier that holds one or more section images.

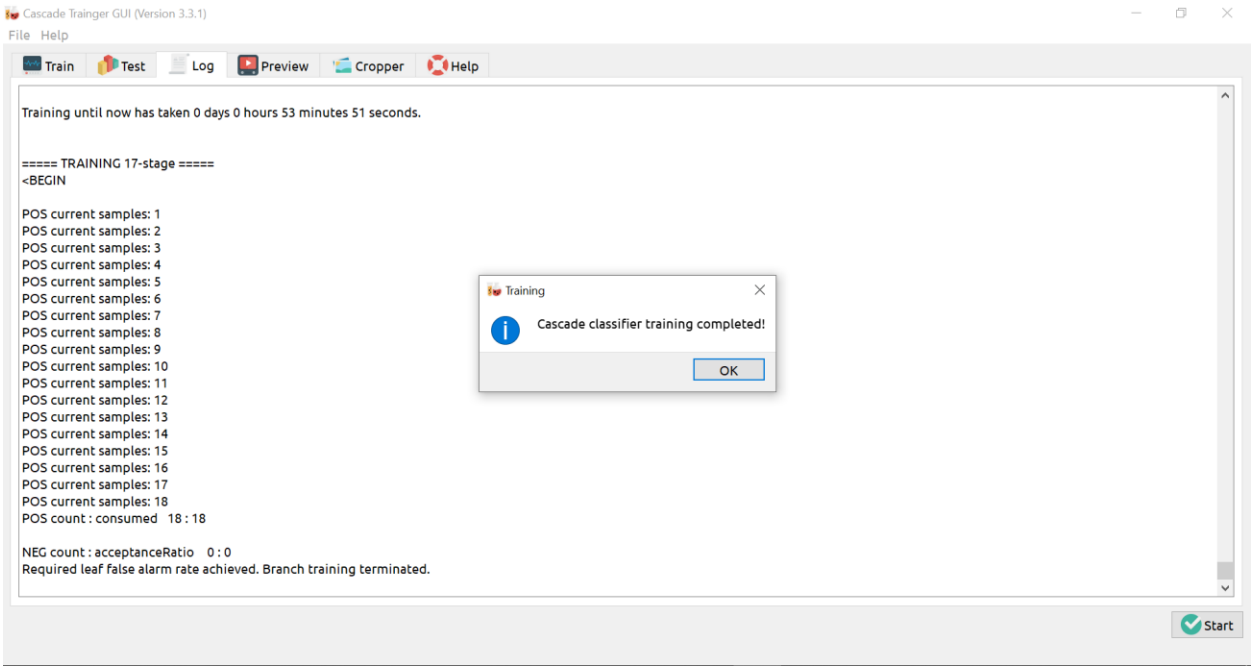


Figure 3.14: Training Haar cascade using negative images

The classifier will be evaluated using a video clip. Path should refer to a video file that can be used as input in this situation. Path is obligated to be set to a folder in this situation. This function is usually used in response to all of the possible choices. During this case Path had a far better [4] argument to a video file which can be generated subsequently the test put the finishing touches to.

3.4.4 THE TRAINING AND TESTING OF HAAR CASCADE FOR THE SYSTEM

The investigation also enhances cascade classifier prototypes. Cascade-Trainer GUI is a database that will be used to train. It uses a graphical interface to align the bounds and make OpenCV tools for drilling and testing classifiers more available. This is often the initial display, and it can be used to train classifiers. Normally, we would like to have thousands of positive and negative image samples to train classifiers, but there are periods when we can get away with less. We'd like to create a folder for our classifier to begin the training. Positive spitting picture samples are representations of the thing we want our classifier to learn to identify. For example, if we want to coach and detect cars, we'll need a large number of car pictures. Negative images are often images that are not positive, but they must be appropriate to the positive images in reality.

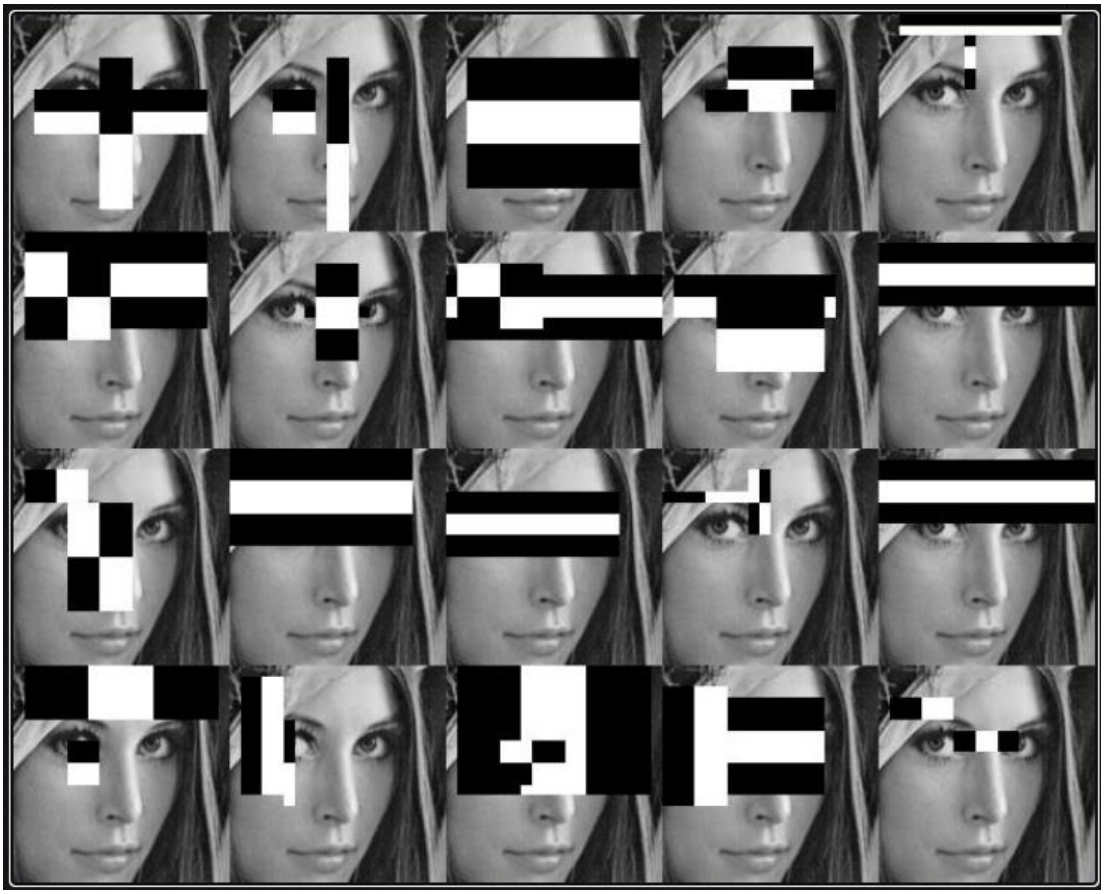


Figure 3.15: Haar features in a picture

The Communal, Cascade, and Boost tabs are commonly used to set a multitude of parameters for transforming classifier training. After that, we'll line up the sample's width and height. If we have sample images, for example, we must determine the ratio, which in this case is. We'll also manually enter the Cascade Classifier XML field and the trail to cascade XML [15]. We'll pick one of the following options for Input Settings, and we'll line the trial up with this option: One Image: Due to the scene in which identification will take place, only one image will be used. Path should point to a single image file in this case. Only supported images are often selected. During this case, Path ought to be set to a dossier that holds one or more section images. A video file is going to be used for testing the classifier. Path should refer to a video file that will be used as input in this scenario. Path is obligated to be set to a folder in this situation. This function is commonly used in response to all of the possible choices. During this instance, Path had a better argument to a video file that would be created after the test was completed.

3.4.5 THE ACCURACY COMPARISON BETWEEN TRAINING HAAR CASCADE

To encourage the doctor to choose from a range of instruments depending on his or her needs. The methods used were Convolutional Neural Networks and Haar classifiers, with the former supplementing a previous element detection step and the latter trained to allow it to not only detect but also classify items. 4 percent accuracy was achieved within the classification of the two groups of the primary branch of the tree, while Haar classifiers achieved 90 percent accuracy within the detection of one of the five instruments, whose classifier provided the simplest results. Simultaneous and automatic face recognition in images, in real-time video, or in offline video are three of the most widely studied topics in image processing prior to face tracking, landmark recognition, and face recognition. The ever-evolving era requires the event of more precise technology. In particular, many problems in technology and criminal security require facial classification identification in problem solving. In this article, the Viola Cascade Classifier-Jones Haar and Oriented Gradient Histogram are used for facial recognition because both methods have an honest level of precision in choosing an object. AD's tactics made a strong classifier. From left to right, the AD strategy is seen. The larger blue circles are caught by the classifier, so the misclassified orange circles are given more weight, while others are reduced. Convolutional Neural Networks (CNNs) are artificial intelligence methods that have been used in pattern recognition since 1998, image detection since 2012, and, more recently, cardiovascular disease diagnosis by ECG (Electrocardiogram) The versatility in which these networks may distinguish various types of categories is attributable to the fact that they do not have a set design or depth, but rather enable the user to customize and define these characteristics based on the application's requirements. The different layers that are added to the architecture are shown here, with the purpose of each one explained. CNN has a variety of applications, including facial recognition, classifying hand positions as open or closed, classifying hidden characters in a text, and assessing the popularity of a face pose, as well as object detection, scenario classification, and speech recognition and speech processing, among others.

On the other hand, other AI techniques are used for the popularity of patterns in images, such as those used in figure 3, using three different types of classifiers to identify vehicles so that a car can recognize its surroundings and drive itself. These techniques include cascading classifiers such as the oriented gradient histogram that helps ramp the image, local binary patterns that help the difference between adjacent pixels of small areas of the image, and hair-assisted filters of the type of hair. Similarly, for pedestrian identification in the street, the same classifiers are used, with training photos captured by surveillance cameras from various angles and heights. Underwater, cascade classifiers like hair type are commonly used to classify faces, artifacts, and other features like blood and heat to identify a species of fish. The fact that these classifiers calculate an "integral image" derived from an ingenious image, in which elements of the image, such as the context, are continuously extracted to eventually determine personal aspects associated with the

that he is going to a meteorological game captured, demonstrates their detection capabilities.

During training, it was discovered that by recommending a large number of stages for each classifier and setting a small ratio of positive images per stage, a decrease in the number of false positives was achieved, even though the training was stopped before the ten stages were reached. Establishing the Ots for each instrument, on the other hand, improved the classifier's popularity capabilities because the windows were produced with the expected proportions for each method, resulting in a much better capture of the element with a reduction in false positives but an increase in training time.

3.4.6 METHODOLOGY IN A GLIMPSE

For the detection of masks, we have used our trained Haar waterfall to improve precision. after importing modules and libraries, captures a real-time video entry. Then, we convert it to an image in grayscale to reduce computational time. Then we will add an Infinite While Loop, which will repeatedly capture the images of the transmission and look for faces with the Haar cascade classifier. If the Haar cascade coincides with a face, the pre-trained model will predict if the person is using or not a mask, and if the Cascade Haar does not match a face, he will show that no face was found in the frame. It was built for the MASK detection process. It uses a graphical interface to configure the parameters and facilitate the use of OpenCV tools to train and test the classifier. From the GUI when we go to the Facial Recognition section, it will identify people based on the data set using Cascade HAAR classifiers. Everything we need to do now is prepare a classifier to take the measurements from a new test image and find out which known individual is the best equivalent. The output of the classifier is shown once it has been run, and the person's name is listed. The mask identification segment decides whether or not the individual is wearing a mask. Each function of a hair cascade is compared to an input image to perform recognition. The decision thresholds for the features are initially low, however as the levels increase, the system becomes more efficient.

3.5 HUMAN DETECTION USING HAAR CLASSIFIERS

In order to delay the transmission of contagious diseases, social spinning is an effective method. Citizens are being urged to minimize their contacts with one another in order to reduce the risk of disease transmission by physical or near contact. AI helped in the automation of a number of everyday activities. In this section, we'll go over what to track social distancing using Python in conjunction with Deep Learning and Computer Vision. The full article illustrates how artificial intelligence was used to track social distance in a public space using already-existing street cameras.

3.5.1 BASIC IDEA BEHIND HUMAN RECOGNITION

The entire world is focusing on shielding people from the corona virus's increasing prevalence. People are in charge of ensuring that their life expectancy and culture are drawn to the consequences. We will use Deep Neural Network to define the social distancing which is most essential to us in order to overcome this specific challenge. The most recent object recognition techniques (SSD, RCNN, and others [5]) were discussed. to decide the work is superior) to people at the stated location first. We can also see if the people we detect are maintaining social distance using the 6-foot distance calibration techniques. We construct a deep neural classifier to distinguish people with and without a mask in order to define the face mask. We recognize the face using a hair cascade classifier and then apply this deep neural classifier to decide if the individual has a mask or not. We can address the underlying problem this way. The comprehensive research is carried out with the assistance of previous literature, and the findings are based on social distancing and engineering estimates. The proposed design suggests a global projection based on the AWS IoT system, which will be improved further by the implementation of a comprehensive forecasting system and real-time tracking of the face mask detection system with large quantities of data. In order to delay the transmission of contagious diseases, social distancing is an effective method. They are told to minimize their contacts with one another in order to reduce the risk of the disease spreading by physical or near contact. Artificial intelligence has been used to simplify the stranded everyday activities. We'll go over how Python could be used in combination with deep learning and computer vision to track Social Distancing. Once we're in the specifics, let's take a look at the final performance. The complete Python code with installation instructions is available on GitHub. The footage is the product of using artificial intelligence to track social distance in a public space, which is now possible with Street cameras.

3.5.2 HUMAN DETECTION USING OpenCV HAAR CASCADE

We go through state-of-the-art object detection methods (SSD, RCNN, etc. to make sure you're doing a great job) [5] to first identify individuals at a given spot, and then use 6ft distance diagnosis and management to see if what's detected is really what's there. Humans hold their distance from one another. We construct a deep neural classifier to identify masked and unmasked individuals, and then we detect the face using a Haar cascade classifier, and applying this deep neural classifier will help us decide if the person wears a mask or not. The thorough research is carried out with the assistance of previous literature works, and the findings are based on social distancing and technical estimates. The developed framework must also be applied in a considerable lot to increase the prediction model's accuracy in contrast to the training model's progress.

3.5.3 OpenCV HAAR CASCADE BUILT IN OUR WORK

A good example for our work is Google Lens, which helps us to focus on a particular object in an image by clicking on the object in the image. The computer vision algorithm looks at the image catalog and collects data from it. To classify a new image, the algorithm requires a large number of positive and negative images. The first feature appears to be based on the fact that the eye area is often darker than the nose and cheek area. The secondary attribute chosen is based on the fact that the eyes are deeper than the nose bridge. The features with the lowest error rate are selected, suggesting that they are the features that better identify facial images rather than non-facial images. The technique is not straightforward. The method is repeated until the necessary precision or error rate is reached, or until the required number of features is discovered. It’s called poor as it can't identify the picture on its own, but when combined with others, it makes an efficient classifier. The bulk of the picture region in a photograph is non-facial. If you're successful, move on to the second level of features. The region of the face that consists of all the steps is referred to as the window. The procedure is almost identical for the human detection process. The libraries and modules must first be imported. To classify the faces in - frame, we used the human identification Haar cascade. Just after real-time video is shot, the frames are down scaled and Gray scaled to improve system throughput. The Haar cascade is then used to assess how many people are in the picture at any given time. They are capable of detecting individuals from both the front and back views, making them ideal for ensuring the social distancing system.

3.5.4 REAL TIME WORK

The human detection Haar cascade was used to identify the faces for each frame; it includes both identifying the identity of the items and identifying the rectangular boundary that encompasses each object, and then using the Haar cascade to assess how many individuals are involved in the frame in the time elapsed. To detect the presence of a person in a thermal picture, the Haar-Cascade classifier method was used. The range between the object and the camera, the angle between the camera and the object, the number of objects, and the environmental conditions during image acquisition were all varied for complete performance. The findings showed that as the range between the camera and the object improved, so did the detection accuracy of human detection results. The system was able to detect multiple humans in the picture, with their positions facing each other, beside each other, or overlapping. If any other objects in the image had human-like features, the object would be detected as a human as well, resulting in false detection.

3.6 COMPILATION OF THE TOTAL SYSTEM

The entire device is built to locate and remove the face area from the surrounding environment. It will have facial features. It will look for the face NumPy ndarray's row and column values. The Haar Cascade classifier is a machine learning-based technique

that uses a large number of positive and negative images to train it. Our software's aim is to keep people safe by testing whether they're wearing a mask and maintaining social distance. When we go to the face recognition portion of the GUI, it will use Haar cascade classifiers to classify people based on the data set. All that we have to do now is train a classifier to measure the size from a new test image and decide which known individual is the most similar. After performing this classifier, this should show the classifier's results as well as the particular person.

3.6.1 THE ROLE OF GUI IN OUR SYSTEM

The graphical user interface is anything but a framework that helps you to communicate with computers more easily and enriches the experience of entering commands into your code. On desktops, notebooks, and other digital equipment, they are used to perform a number of functions, [12] for example. The following are the examples of applications that make use of the GUI's capabilities: Creating a calculator that has a user interface and features that are close to those found in a calculator. A GUI app includes text editors and IDEs for coding. Chrome, Firefox, Microsoft Edge A graphical user interface for managing a drone from your desktop is another fascinating use case. The graphical user interface would most likely include keys for controlling the drone [12] and therefore a screen that shows the monitor and real-time video feed recorded mostly by drone.

Let us just take a closer look at a few of the frameworks Python offers for building a graphical user interface: PyQT is one of the Qt library's preferred cross-platform Python bindings for the Qt software development framework., and It blends the best of Python and Qt programming languages, offering programmers the option of creating pure Python code or using Qt Designer to construct visual dialogues. Find this graphical user interface's process flow chart. We can now see that we need to import the Tkinter module first. The next move would be to use the Tkinter method to configure the window manager and assign it to a parameter that generates an empty window. Eventually, we used the key loop method () to hold the window open before we closed it remotely. In the back end, an infinite loop is run.

3.6.2 THE COMPILATION OF THE WHOLE SYSTEM

Face recognition is a computer technology that is used in a number of applications to recognize human faces in digital images. It is a necessary first step in facial recognition systems and a very important feature of computer vision. It works to locate and extract the face region from the bottom. To implement this computer vision application, we have used the OpenCV library and the Haar Cascade classifier. OpenCV is a programming library targeted mainly at real-time computer vision. It is available for both Android and

iOS. It's a free and open-source computer vision library with over 2500 algorithms. The Cascade classifier will construct a cascade classifier from an image. Which will include the features of the face. The picture and features file were then read by OpenCV. It looks for the face Numpy Ndarray's line and column values. A rectangular Face Box is used to image and the corresponding.

Haar Cascade Classifier is a machine-learning technique that trains system classifiers using a large number of positive and negative images. Positive images are those that hold the pictures that identify our classifier. Negative pictures are those who are not that Object contained that we want to recognize. It can detect any sort of object as long as the required XML file is available. With the Cascade coach Interface, we can even build our own XML files from scratch for any sort of object. They want to make use of it. Case in point. Hair cascade classifiers have three specific characteristics. The features are edge, line, and middle surround. The purpose of our system is to ensure safety by checking that the person is wearing a mask and maintaining social distance. At the same time, the security is confirmed by the face recognition process. To this end, we developed a three-part graphical user interface for recognizing various functions. Since we wanted the system to have a properly looking user interface, a graphical user interface is required. A graphical user interface is a user interface that helps us to communicate with it. The code enhances our command-entry experience. If we go to the Face Recognition section from the UI, people are identified from the data set using hair cascade classifiers. All we need to do now is train a classifier to recognize it. Based on the results of a new test picture, the measurements show which known individual is most likely to fit. The output of the classifier, as well as the particular person, are shown after it has been run [23]. The mask detection segment decides whether or not the individual is wearing a mask. Each function of a Haar waterfall is checked against an input image to make detection. The determination requirements for features are poor at first, but as the stages progress, it becomes more effective. It can detect the number of individuals in a span of time elapsed in order to ensure social distancing.

CHAPTER FOUR
SIMULATION AND RESULT

These are the results we got after the execution of the code. We used Python and PyCharm for coding. Our Web camera is used for capturing the video and Images.

4.1 GUI

After executing the code, at first, a GUI will appear and it includes three buttons. GUI means (Graphical User Interface). Python provides a variety of options for designing a graphical user interface. Tkinter is the most commonly used tool of all the others. It's the most commonly used Python interface to the Tk GUI toolkit that comes with Python.
The three buttons are named Face detection, Mask detection, Human detection. All of them have different functionaries. We compiled them to make a wholesome system that includes everything any organization needs.
As three buttons work differently, we can control when we need to run which code. This GUI will appear when we first run the code.

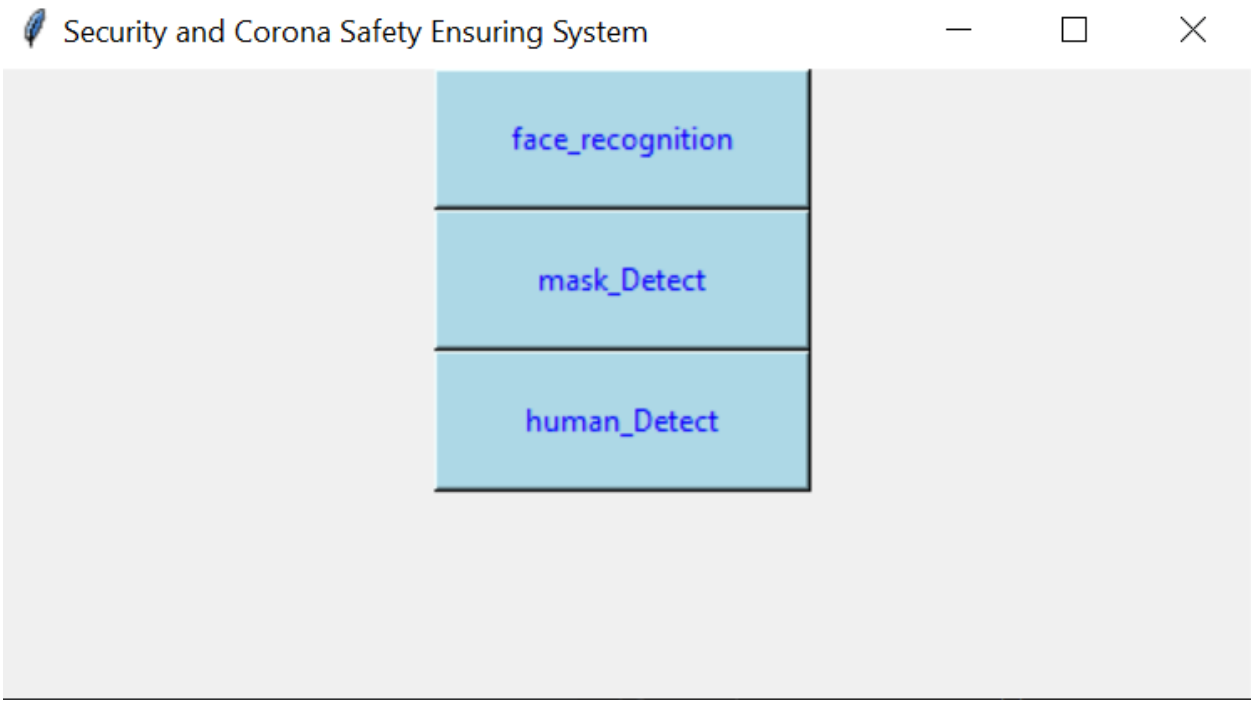


Figure 4.1: GUI

4.2 FACE RECOGNITION

We have seen that our GUI has three buttons. When we click the first button, the code of face recognition will execute. This will detect those faces accurately that were labeled in our data set. We have created our own data set and labeled them according to the name of the persons. We can increase the data set and then our program will be able to recognize more people.
So, after pressing the first button named face detection, the code will call the function and that function includes the code of face recognition. So that code will execute and the webcam will be on. On the screen we will be able to see how it's working.

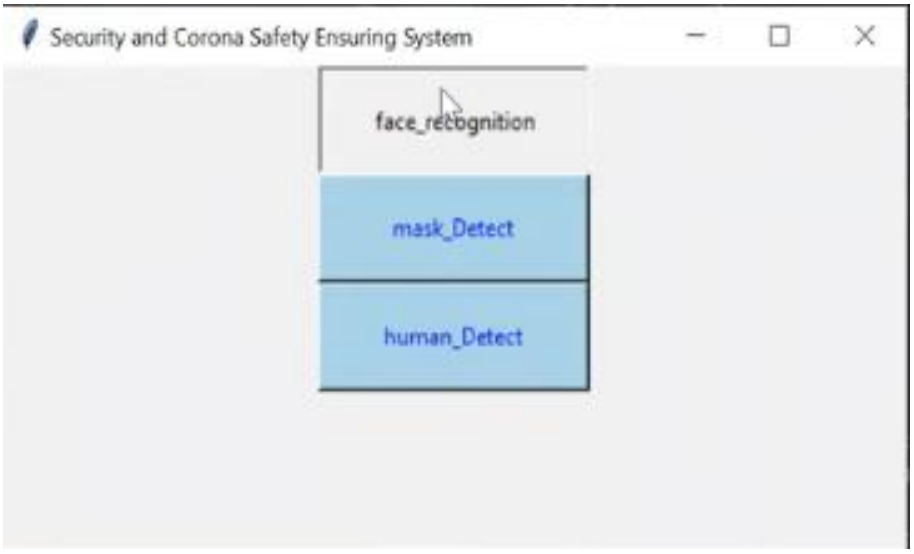


Figure 4.2: GUI (While pressing the button of face recognition)

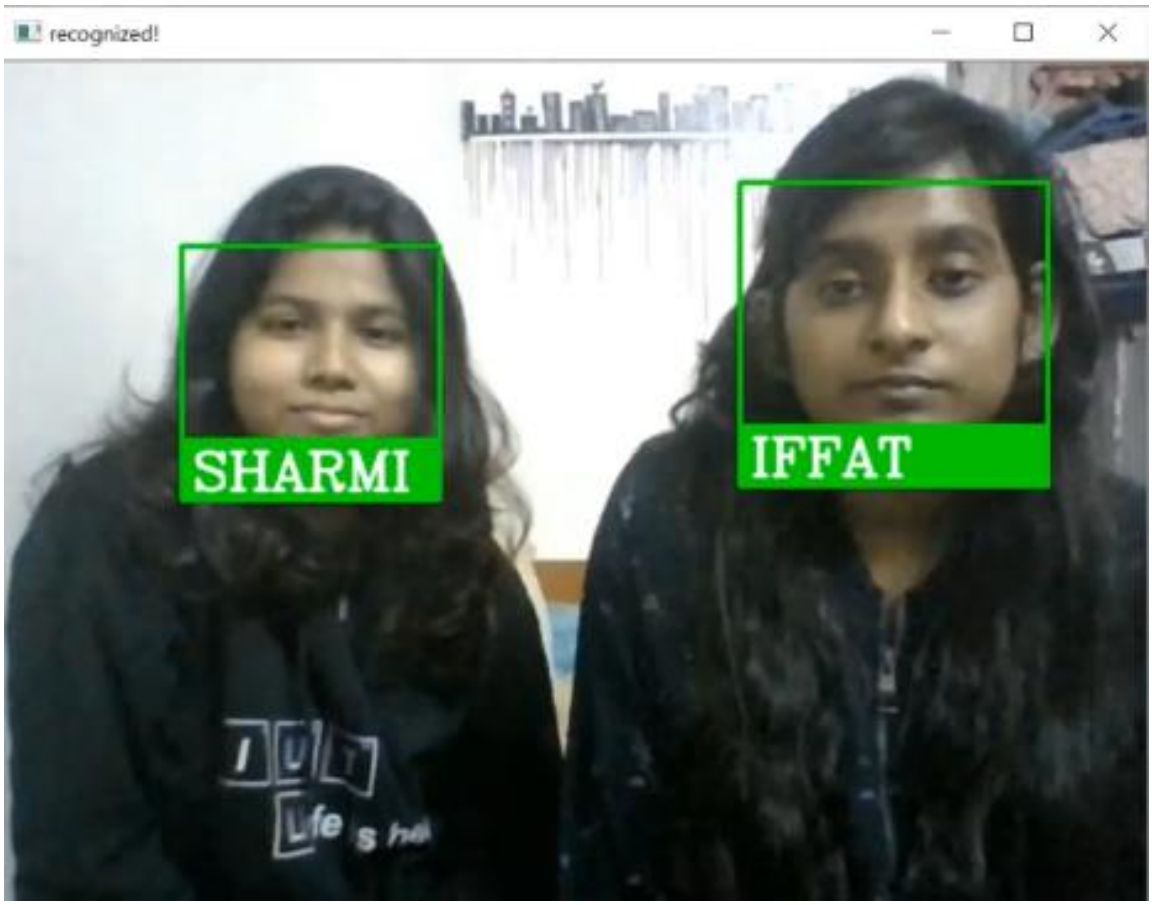


Figure 4.3: Execution of face recognition code

It is seen that this program detected and recognized persons perfectly. In our dataset, we have some demo pictures of these persons and they were labeled by the names of these individuals.

4.3 MASK DETECTION

Now, when the button named mask detection will be pressed, another code will execute. This will ensure whether one is wearing a mask or not. Wearing a mask is a basic need nowadays. Research shows that wearing a mask reduces the risk of spreading [13] different types of diseases.

In our mask detection code, we have used our designed Haar cascade named mask. When one will press the second button, then button will call a function that includes the code of mask detection. The attached screenshots below will show how it's working on our screen.

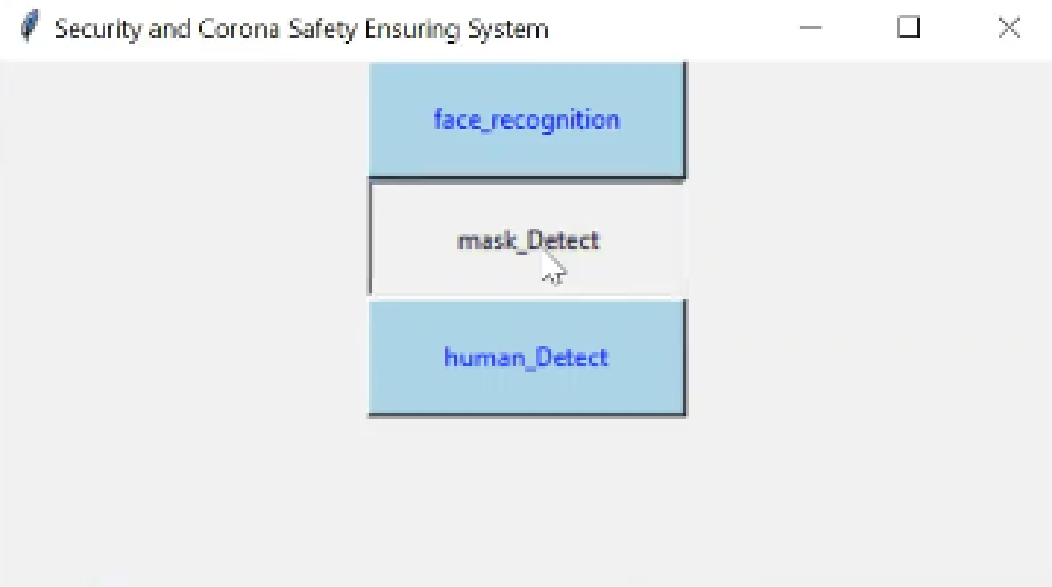


Figure 4.4: GUI (While pressing the button of mask detection)

If one is not wearing a mask, it will show a message “please wear a mask”.

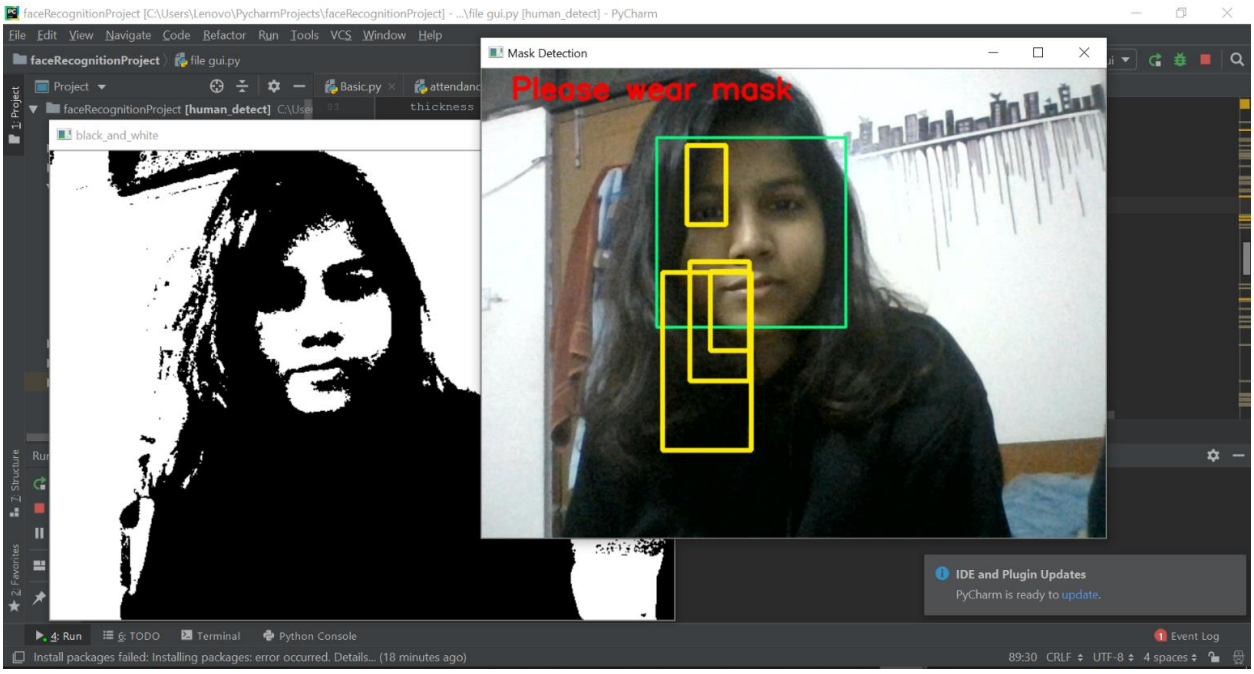


Figure 4.5: Person not wearing mask

We can see that there is a person in front of the camera and she is not wearing a mask.

Our code just not only detects whether one is wearing a mask or not. It also detects whether one is wearing the mask properly. If one is wearing a mask but not properly, it will still show that previous message “please wear a mask “.

Figure 4.6: Process of wearing mask properly

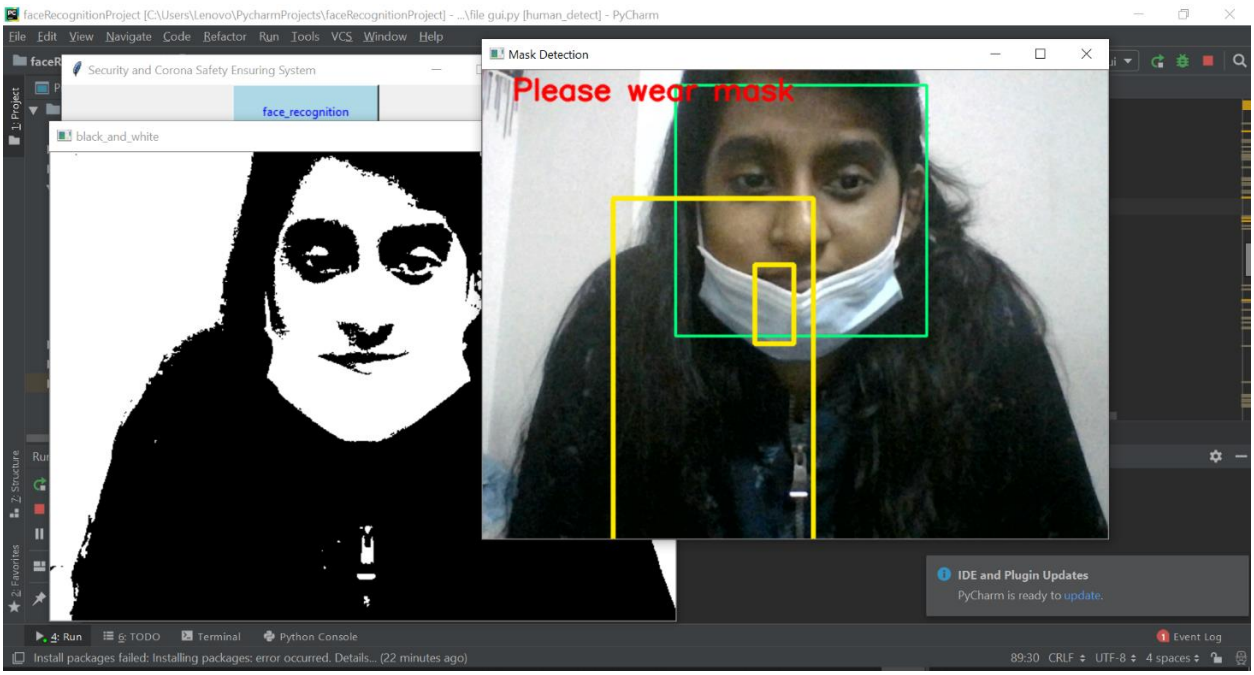


Figure 4.7: Person wearing mask but not properly

Now if our code finds that the person is wearing a mask properly, it will show “thank you for wearing a mask”. It shows a thank you message by appreciating the person. The mask

can be of any type, it can detect all masks properly.

Figure 4.8: Person wearing mask properly

As our code is able to detect humans, so if it finds that there are no people on the screen, it will show “no face found”.

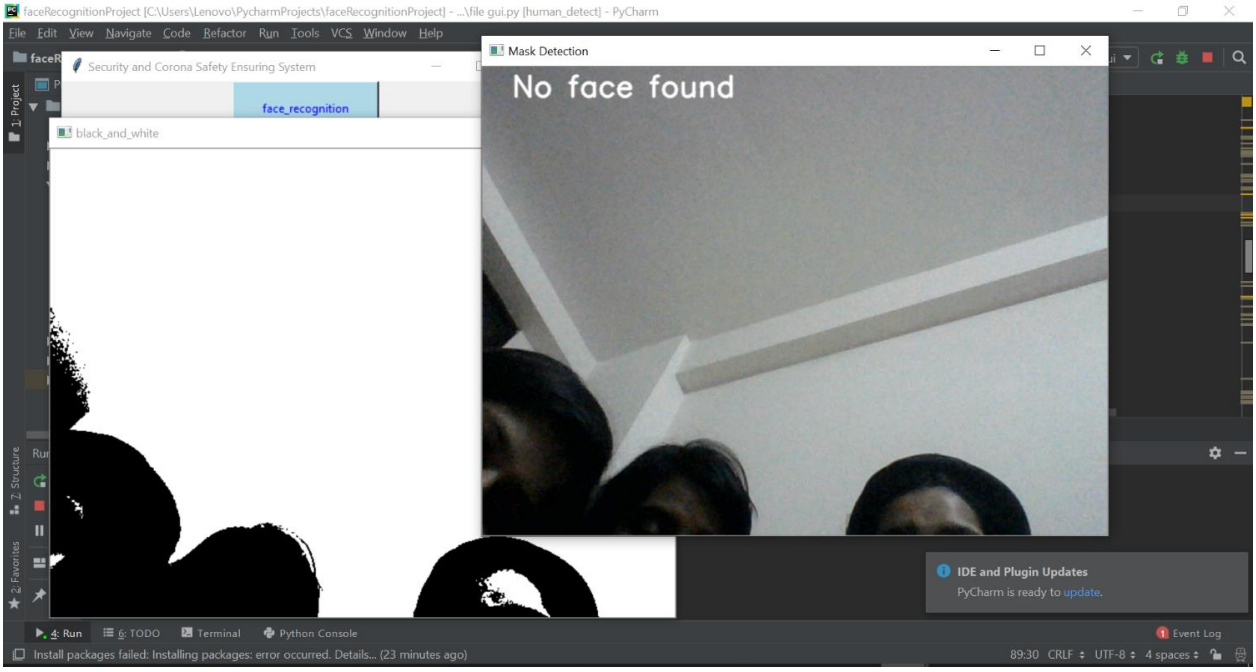


Figure 4.9: No human found in front of the camera

Thus, the mask detection process is being completed.

4.4 HUMAN DETECTION

After the successful face recognition and mask detection process, the only button left to press is human detect. When we will press that, the camera will be on and it will detect how many people are in that frame. This can be used for measuring social distance. Since social distancing helps to prevent COVID-19 from spreading, we should always keep a gap of at least 1 meter between us and avoid spending time in crowded places or in groups.

The attached screenshot below will show the process. If it detects too many people on the screen, then we will be able to be aware and take cautionary steps to maintain a proper distance.

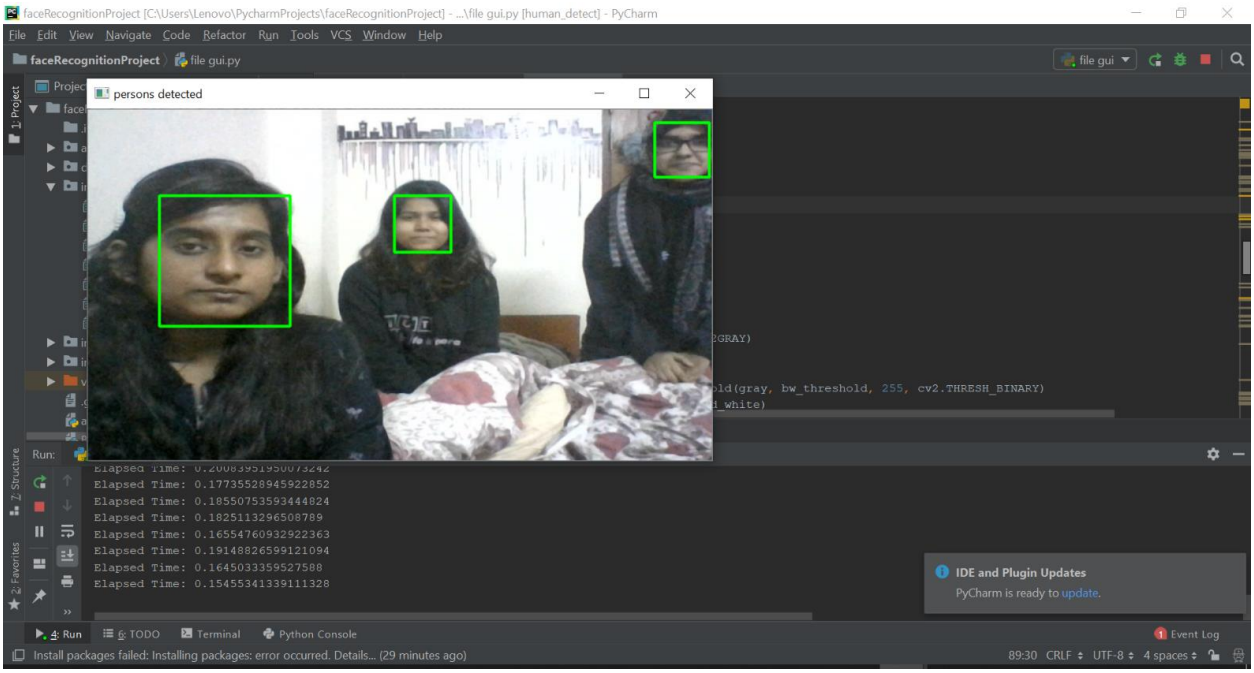


Figure 4.10: Human detection

4.5 THE WHOLE SYSTEM

Our system includes three programs. We reproduced and designed different codes from scratch and increased their accuracy level. After that, we compiled them by making a GUI in python and made a wholesome system that includes everything any organization needs nowadays.

CHAPTER FIVE
CONCLUSION & FUTURE WORK

5.1SUMMARY:

To control the spread of the COVID-19 pandemic, safety measures must be taken. Furthermore, since crime is on the rise, a proper protection system is a must-have nowadays. Keeping those in mind, the have modeled a wholesome system that includes face recognition, mask detection, and human detection programs. The mainly used Haar cascade classifiers to build the system. To train Haar cascades, validate and test the model, the used the data set that contains positive and negative images. The model was hypothesized on images and live video streams. In the face recognition system, the also used the HOG method for better accuracy. This whole system can be deployed in many areas like scroll, colleges, universities, airports, and other places that need proper security and safety. The system will monitor the public to avoid the spread of the disease by checking who is following basic rules of wearing masks and maintaining social distances and it will make sure the security of any place by face recognition process.

5.2CONTRIBUTION OF THE ADOPTED METHODOLOGY OF SYSTEM:

Here the system has compiled three individual projects by making a GUI. Three distinct functions, like face mask, face and human detection, have been added to a single project using the Tkinter module to construct a GUI. Face mask detection, facial detection and human detection are the three main features of this project that ensure social distance. This project can be used to shield any place from intruders, to ensure that people wear masks properly, and to maintain social distance.

5.3 FUTURE WORK:

The have a plan to work further on this system. In the future, the want to implement the hardware part of the system using raspberry pi. They will try to design more Haar cascades to improve the accuracy level of the system. By using this system, they can keep any place safe from intruders and ensure people are wearing masks properly, which the undoubtedly need in present days. The have the vision to make a proper security surveillance system where a CCTV camera will capture the video and a monitor will be used for the detection and recognition process. The think this will be an appropriate system for any organization that needs proper security and safety.

5.3.1HARDWARE IMPLEMENTATION:

The project is based on software now. It uses the webcam of the PC for detection. But the want to work more on this project. In future, they will try to implement it using raspberry pi. It has a pi camera and that is very useful for any detection system. Using this method, they can obtain highly accurate face recognition.

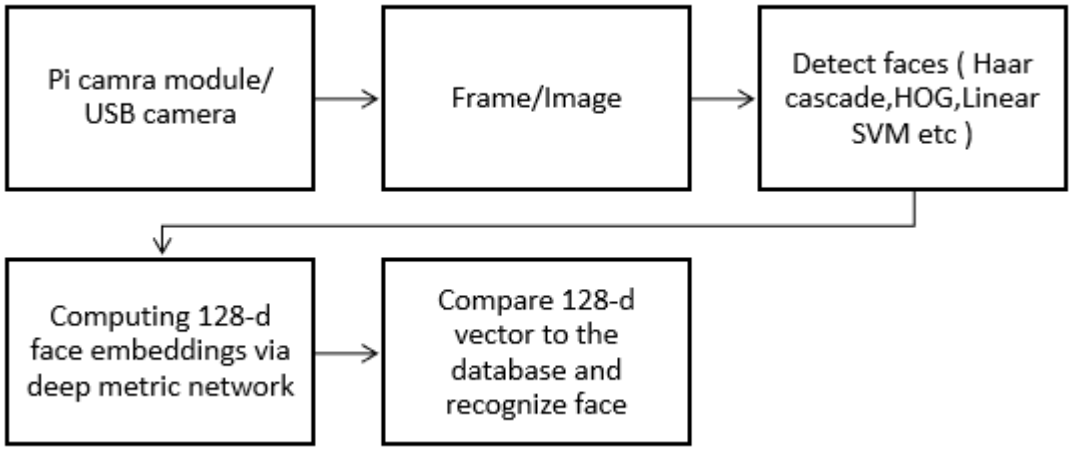


Figure 5.1: Raspberry pi in face detection and recognition

The have a vision to make a system for any place that needs this type of security. First, they will use a CCTV camera which will capture the real time video. Then using the face mask detection process, it will make sure whether one is wearing a mask or not. When it will find that one is wearing a mask, it will then ask to show the face for face recognition. If it recognizes the person, then it will open the door for him. It will also see how many people are on the frame and will ask them to maintain social distance. Thus, safety and security will be maintained.

5.4COMPARISON

The work did need some research in this face detection area and found that the previous works only contain one/two features. This is a wholesome system that contains three main features. That's why a system like this will prove to be very convenient and productive in this time of the pandemic.

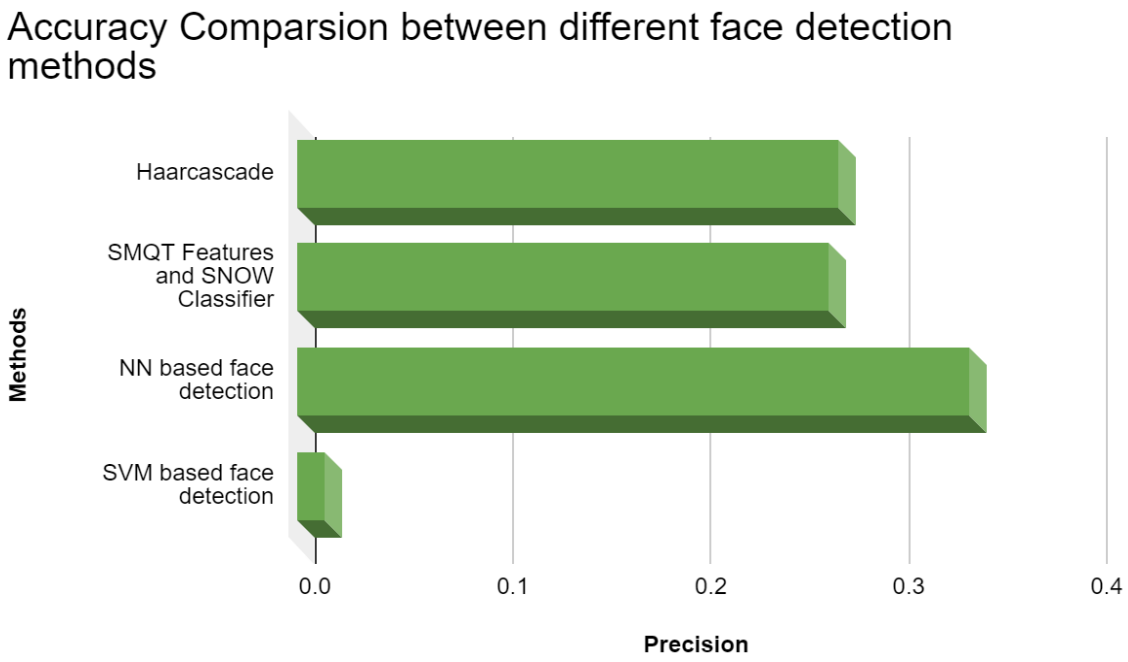


Figure 5.2: Accuracy comparison between face mask detection process

Here they can see a graph that shows a comparison of accuracy level or precision between different methods of face mask detection. The used a Haar cascade because Haar-Features are good at detecting edges and lines. This makes it especially effective in face mask detection areas. *Cascade classifier* is one of the fastest algorithms that runs in real-time. The have chosen it because the wanted to design it from scratch and it is the simplest and oldest way of using the method of the object detection of the system.

The faced some difficulties while implementing the project. While designing the Haar cascade classifier it was difficult to find a perfect data set. As the systems are working with real-time videos, the pictures taken from the video for the detection process are of low resolution so it has a negative impact on the accuracy level of the Haar cascades. When the used huge datasets, the program became very slow and complex. As the processing pother of the pc is not very high, it takes a long time to train as the Haar cascade needs more memory input and so eventually it needs more time for the processing system.

REFERENCES

[1] (PDF) Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence, 2021)

[2] A. Anwar, "Using Python to Monitor Social Distancing in a Public Area," *Medium*, 19-Jan-2021. [Online]. Available: https://towardsdatascience.com/monitoring-social-distancing-using-ai-c5b81da44c9f. [Accessed: 23-Feb-2021]

[3] *Paula C. Useche Murillo, Robinson Jimenez Moreno, Javier O. Pinzon Arenas, Comparison between CNN and Haar classifiers for surgical instrumentation classification*. [Online]. Available: http://www.m-hikari.com/ces/ces2017/ces25-28-2017/711157.html. [Accessed: 23-Feb-2021]

[4]https://www.thejournaldaily.com/real-time-face-recognition-real-time-face-recognition-opencv-python/?cv =1

[5] Imran Ahmed, Misbah Ahmad, Joel J.P.C. Rodrigues, Gwanggil Jeon, Sadia Din,A deep learning-based social distance monitoring framework for COVID-19,Sustainable Cities and Society,Volume 65,2021,102571,ISSN 2210-6707,https://doi.org/10.1016/j.scs.2020.102571.

[6] Tejas R. Phase , Dr. Suhas S. Patil, 2019, Building Custom HAAR-Cascade Classifier for face Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 08, Issue 12 (December 2019)

[7] A. Geitgey, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning," *Medium*, 24-Sep-2020. [Online]. Available: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78. [Accessed: 23-Feb-2021]

[8] M. Jiang, *RETINAFACEMASK: A FACE MASK DETECTOR*, Jun. 2020.

[9] G. Costache, *Real-Time Video Face Recognition for Embedded Devices*, 01-Aug-2011. [Online]. Available: https://www.intechopen.com/books/new-approaches-to-characterization-and-recognition-of-faces/real-time-video-face-recognition-for-embedded-devices. [Accessed: 23-Feb-2021]

[10] A. Team, "Image Detection, Recognition, And Classification With Machine Learning," *Azati*, 17-Sep-2020. [Online]. Available: https://azati.ai/image-detection-recognition-and-classification-with-machine-learning/. [Accessed: 23-Feb-2021]

[11] G. Costache, *Real-Time Video Face Recognition for Embedded Devices*, 01-Aug-2011. [Online]. Available: https://www.intechopen.com/books/new-approaches-to-characterization-and-recognition-of-faces/real-time-video-face-recognition-for-embedded-devices. [Accessed: 23-Feb-2021]

[12] "(Tutorial) Introduction to GUI With TKINTER in PYTHON," *DataCamp Community*. [Online]. Available: https://www.datacamp.com/community/tutorials/gui-tkinter-python. [Accessed: 23-Feb-2021]

[13] Chavda, Amit & Dsouza, Jason & Badgujar, Sumeet & Damani, Ankit. (2020). Multi-Stage CNN Architecture for Face Mask Detection.

[14] Hjelmås, E., & Low, B. K. (2001). Face Detection: A Survey. Computer Vision and Image Understanding, 83(3), 236–274. https://doi.org/10.1006/CVIU.2001.0921

[15] Schmidt, K. L., & Cohn, J. F. (2001). Human facial expressions as adaptations: Evolutionary questions in facial expression research. *American journal of physical anthropology, Suppl 33*, 3–24. https://doi.org/10.1002/ajpa.2001

[16] L. Tagliaferri, *An Introduction to Machine Learning*. [Online]. Available: https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning

[17] M. A. M. A. T. A. S. KALAS, *REAL TIME FACE DETECTION AND TRACKING USING OPENCV*. Associate Professor, Department of IT, KIT'S College of Engg., Kolhapur

[18] "Dr Rajiv Desai," *Dr Rajiv Desai FACIAL RECOGNITION TECHNOLOGY Comments*. [Online]. Available: https://drrajivdesaimd.com/2018/12/03/facial-recognition-technology/. [Accessed: 23-Feb-2021]

[19] T. Greenhalgh, M. B. Schmid, T. Czypionka, D. Bassler, and L. Gruer, "Face masks for the public during the covid-19 crisis," *The BMJ*, 09-Apr-2020. [Online]. Available: https://www.bmj.com/content/369/bmj.m1435.short. [Accessed: 23-Feb-2021]

[20] "Face Recognition," *Electronic Frontier Foundation*, 15-Feb-2021. [Online]. Available: https://www.eff.org/pages/face-recognition#:~:text=Face%20recognition%20is%20a%20method. [Accessed: 23-Feb-2021]

[21] Preeti Nagrath, Rachna Jain, Agam Madan, Rohan Arora, Piyush Kataria, Jude Hemanth,SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2,Sustainable Cities and Society,Volume 66,2021,102692,ISSN 2210-6707,https://doi.org/10.1016/j.scs.2020.102692.(https://www.sciencedirect.com/science/article/pii/S2210670720309070)

[22] T. Meenpal, A. Balakrishnan and A. Verma, "Facial Mask Detection using Semantic Segmentation," 2019 4th International Conference on Computing, Communications and Security (ICCCS), Rome, Italy, 2019, pp. 1-5, doi: 10.1109/CCCS.2019.8888092.

[23] "Python: Haar Cascades for Object Detection," *GeeksforGeeks*, 25-Nov-2019. [Online]. Available: https://www.geeksforgeeks.org/python-Haar-cascades-for-object-detection/. [Accessed: 23-Feb-2021]

[24] C. Rahmad1, R. A. Asmara2, D. R. H. Putra2, I. Dharma1, H. Darmono3, and I. Muhiqqin2, "IOPscience," *IOP Conference Series: Materials Science and Engineering*, 01-Jan-2020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/732/1/012038. [Accessed: 23-Feb-2021]

[25] http://www.ijecse.org/wp-content/uploads/2012/06/Volume-3Number-2PP-175-184x.pdf

[26] amin ahmadi , "Cascade Trainer GUI," *Amin*, 30-May-2020. [Online]. Available: https://amin-ahmadi.com/cascade-trainer-gui/. [Accessed: 23-Feb-2021]

[27] S. Mallick, "Histogram of Oriented Gradients: Learn OpenCV," *Learn OpenCV | OpenCV, PyTorch, Keras, Tensorflow examples and tutorials*, 05-Feb-2018. [Online]. Available: https://learnopencv.com/histogram-of-oriented-gradients/. [Accessed: 23-Feb-2021]

[28] "Social Distancing," *Centers for Disease Control and Prevention*. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html#:~:text=Social%20distancing,%20also%20called%20"phys. [Accessed: 23-Feb-2021]

[29] V. C.-C. Cheng, *The role of community-wide wearing of face mask for control of coronavirus disease 2019 (COVID-19) epidemic due to SARS-CoV-2*, Apr. 2020.

[30] https://iopscience.iop.org/article/10.1088/1742-6596/1201/1/012015/pdf

[31]https://www.researchgate.net/publication/335998006_Applying_the_Haar-cascade_Algorithm_for_Detecting_Safety_Equipment_in_Safety_Management_Systems_for_Multiple_Working_Environments/figures?lo=1

[32] https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/