A Project Report

on

ONLINE FACE RECOGNITION SYSTEM

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr.Arjun KP
Assistant Professor
Department of Computer Science and Engineering

Submitted By
Poonam Kumari Yadav-18SCSE1010325
Nandita Pandey - 18SCSE1010195

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
(December ,2021)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled**"ONLINE FACE RECOGNITION SYSTEM"** in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to December,2021, under the supervision of Dr Arjun KP, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Poonam Kumari Yadav , 18SCSE1010325

Nandita Pandey, 18SCSE1010195

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor
Arjun KP

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Poonam Kumari yadav: 18SCSE1010325 and Nandita Pandey: 18SCSE1010195 has been held on_____and his/her work is recommended for the award of Bachelor of Technology.

Signature of Examiner(s) **Signature of Supervisor**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2013
Place: Greater Noida

Acknowledgement

The feeling of gratitude when we expressed a holy acknowledgement and it's with deep sense of gratitude that we acknowledge the able guidance.
We express our grateful thanks to Mr. Arjun K P, Associate professor, Department of Computer Science and Engineering, Galgotias University for providing us an opportunity for the research report on "**THE FACE RECOGNITION SYSTEM USING PYTHON**" and for his keen interest and the encouragement, which was required for the fulfilment of our capstone project report. We would also like to thank him for giving us valuable guidance at all levels, help and suggestions, which prove to be valuable for preparation of the report.
Finally, I would also like to thank all our friends for their cooperation and interest, which was necessary for completing our project report.
Date:

Nandita Pandey & Poonam Kumari Yadav
School of Computing Science & Engineering,
Galgotias University,
Greater Noida, Uttar Pradesh

TABLE OF CONTENTS

Abstract

Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of face- dimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing.

Face recognition is currently being used to make the world safer, smarter, and more convenient.

CHAPTER-1
INTRODUCTION

In this work, a software for human face detection and recognition is prepared. Initial implementation of this work is service oriented. However, the analysis and design are done to promote the work as product based. The work is entitled as Face Detection and Recognition System. Face Detection is a application software to deal with human face. It has the provisions to collect image from the user so that they can detect the eyes, nose, mouth and whole face of human in the image. There are various advantages of developing an software using face detection and recognition in the field of authentication. Face detection is an easy and simple task for humans, but not so for computers. It has been regarded as the most complex and challenging problem in the field of computer vision due to large intra-class variations caused by the changes in facial appearance, lighting and expression. Face detection is the process of identifying one or more human faces in images or videos. It plays an important part in many biometric, security and surveillance systems, as well as image and video indexing systems. Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class.

Face recognition is part of computer vision. Face recognition is used to identifying a person in biometric method based on image on their face. A person is identified through biological traits. Human eyes can easily recognize people by simply looking at them but the concentration span for human eyes has its limit. Hence, a computerized method is invented to perform face recognition. Face recognition includes the operations of automatically detecting followed by verifying a person from either picture or video.
Facial expression recognition software is a technology which uses biometric markers to detect emotions in human faces. More precisely, this technology is a sentiment analysis tool and is able to automatically detect the six basic or universal expressions: happiness, sadness, anger, surprise, fear, and disgust.
Faces are made of thousands of fine lines and features that must be matched. The face recognition using Python, break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to face Recognition Python is the latest trend in Machine Learning techniques. OpenCV uses Machine Learning algorithms to search for faces within a picture.

Face recognition is performed by using classifiers. A classifier is essentially an algorithm that decides whether a given image is positive(face) or negative(not a face). A classifier needs to be trained on thousands of images with and without faces. Fortunately, OpenCV already has two pre-trained face

detection classifiers, which can readily be used in a program. The two classifiers are:

- Haar Classifier and
- Local Binary Pattern(LBP) classifier.

There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

Recognition algorithms can be divided into two main approaches:

1. **Geometric:** Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features. (Figure 3)
2. **Photometric stereo:** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006).

CHAPTER 2

LITERATURE SURVEY

Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a _specific‘ case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection, can be regarded as a more _general‘ case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically there are two types of approaches to detect facial part in the given image i.e. feature base and image base approach.Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match between training and testing images.

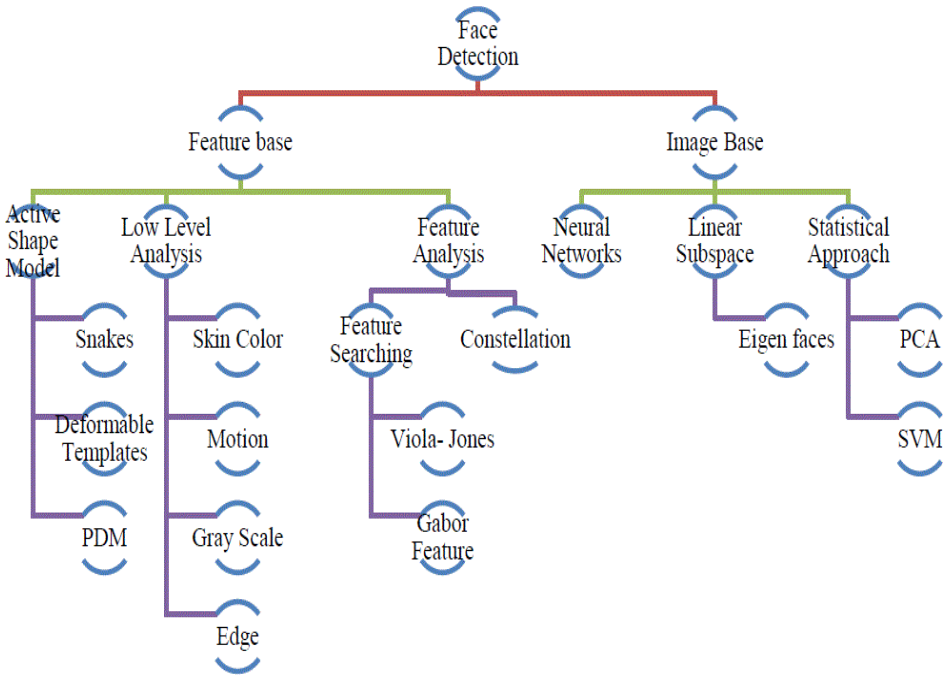


Fig 2.1 detection methods

FEATURE BASE APPROCH:

Active Shape ModelActive shape models focus on complex non-rigid features like actual physical and higher level appearance of features Means that Active Shape Models (ASMs) are aimed at automatically locating landmark points that define the shape of any statistically modelled object in an image. When of facial features such as the eyes, lips, nose, mouth and eyebrows. The training stage of an ASM involves the building of a statistical

a) facial model from a training set containing images with manually annotated landmarks.

ASMs is classified into three groups i.e. snakes, PDM, Deformable templates

1.1)Snakes:The first type uses a generic active contour called snakes, first introduced by Kasset al.

in 1987 Snakes are used to identify head boundaries [8,9,10,11,12]. In order to achieve the task, a snake is first initialized at the proximity around a head boundary. It then locks onto nearby edges and subsequently assume the shape of the head. The evolution of a snake is achieved by minimizing an energy function, Esnake (analogy with physical systems), denoted asEsnake = Einternal + EExternal WhereEinternal and EExternal are internal and external energy functions.Internal energy is the part that depends on the intrinsic properties of the snake and defines its natural evolution. The typical natural evolution in snakes is shrinking or expanding. The external energy counteracts the internal energy and enables the contours to deviate from the natural evolution and eventually assume the shape of nearby features—the head boundary at a state of equilibria.Two main consideration for forming snakes i.e. selection of energy terms and energy minimization. Elastic energy is used commonly as internal energy. Internal energy is vary with the distance between control points on the snake, through which we get contour an elastic-band characteristic that causes it to shrink or expand. On other side external energy relay on image features. Energy minimization process is done by optimization techniques such as the steepest gradient descent.
Which needs highest computations. Huang and Chen and Lam and Yan both employ fast iteration methods by greedy algorithms. Snakes have some demerits like contour often becomes trapped onto false image features and another one is that snakes are not suitable in extracting non convex features.

Deformable Templates:

Deformable templates were then introduced by Yuille et al. to take into account the a priori of facial features and to better the performance of snakes. Locating a facial feature boundary is not an easy task because the local evidence of facial edges is difficult to organize into a sensible global entity using generic contours. The low brightness contrast around some of these features also makes the edge detection process.
Yuille et al. took the concept of snakes a step further by incorporating global information of the eye to improve the reliability of the extraction process.

Deformable Templates:

Deformable templates were then introduced by Yuille et al. to take into account the a priori of facial features and to better the performance of snakes. Locating a facial feature boundary is not an easy task because the local evidence of facial edges is difficult to organize into a sensible global entity using generic contours. The low brightness contrast around some of these features also makes the edge detection process.Yuille et al. took the concept of snakes a step further by incorporating global information of the eye to improve the reliability of the extraction process.

Deformable templates approaches are developed to solve this problem. Deformation is based on local valley, edge, peak, and brightness .Other than face boundary, salient feature (eyes, nose, mouth and eyebrows) extraction is a great challenge of face recognition.E = Ev + Ee
+ Ep + Ei + Einternal ; where Ev , Ee , Ep , Ei , Einternal are external energy due to valley, edges, peak and image brightness and internal energy

Deformable Templates:

Deformable templates were then introduced by Yuille et al. to take into account the a priori of facial features and to better the performance of snakes. Locating a facial feature boundary is not an easy task because the local evidence of facial edges is difficult to organize into a sensible global entity using generic contours. The low brightness contrast around some of these features also makes the edge detection process.Yuille et al. took the concept of snakes a step further by incorporating global information of the eye to improve the reliability of the extraction process.

Deformable templates approaches are developed to solve this problem. Deformation is based on local valley, edge, peak, and brightness .Other than face boundary, salient feature (eyes, nose, mouth and eyebrows) extraction is a great challenge of face recognition.E = Ev + Ee
+ Ep + Ei + Einternal ; where Ev , Ee , Ep , Ei , Einternal are external energy due to valley, edges, peak and image brightness and internal energy

PDM (Point Distribution Model):

Independently of computerized image analysis, and before ASMs were developed, researchersdeveloped statistical models of shape . The idea is that once you represent shapes asvectors, you can apply standard statistical methods to them just like any other multivariateobject. These models learn allowable constellations of shape points from training examplesand use principal components to build what is called a Point Distribution Model. These havebeen used in diverse ways, for example for categorizing Iron Age broaches.Ideal Point Distribution Models can only deform in ways that are characteristic of the object. Cootes and his colleagues were seeking models which do exactly that so if a beard, say, covers the chin, the shape model can \override the image" to approximate the position of the chin under the beard. It was therefore natural (but perhaps only in retrospect) to adopt Point Distribution Models. This synthesis of ideas from image processing and statistical shape modelling led to the Active Shape Model.The first parametric statistical shape model for image analysis based on principal components of inter-landmark distances was presented by Cootes and Taylor in. On this approach, Cootes, Taylor, and their colleagues, then released a series of papers that cumulated in what we call the classical Active Shape Model.

LOW LEVEL ANALYSIS:

Based on low level visual features like color, intensity, edges, motion etc.Skin Color BaseColor is avital feature of human faces. Using skin-color as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features. Under certain lighting conditions, color is orientation invariant. This property makes motion estimation much easier because only a translation model is needed for motion estimation.Tracking human faces using color as a feature has several problems like the color representation of a face obtained by a camera is influenced by many factors (ambient light, object movement, etc Majorly three different face detection algorithms are available based on RGB, YCbCr, and HIS color space models.In the implementation of the algorithms there are three main steps viz.

Classify the skin region in the color space,Apply threshold to mask the skin region andDraw bounding box to extract the face image.

Crowley and Coutaz suggested simplest skin color algorithms for detecting skin pixels.

The perceived human color varies as a function of the relative direction to the illumination
The pixels for skin region can be detected using a normalized color histogram, and can be normalized for changes in intensity on dividing by luminance. Converted an [R, G, B] vector is converted into an [r, g] vector of normalized color which provides a fast means of skin detection. This algorithm fails when there are some more skin region like legs, arms, etc.Cahi and Ngan [27] suggested skin color classification algorithm with YCbCr color space.Research found that pixels belonging to skin region having similar Cb and Cr values. So that the thresholds be chosen as [Cr1, Cr2] and [Cb1, Cb2], a pixel is classified to have skin tone if the values [Cr, Cb] fall within the thresholds. The skin color distribution gives the face portion in the color image. This algorithm is also having the constraint that the image should be having only face as the skin region. Kjeldson and Kender defined a color predicatein HSV color space to separate skin regionsfrom background. Skin color classification inHSI color space is the same as YCbCr color spacebut here the responsible values are hue (H) andsaturation (S). Similar to above the threshold be chosen as [H1, S1] and [H2, S2], and a pixel isclassified to have skin tone if the values [H,S] fallwithin the threshold and this distribution gives thelocalized face image. Similar to above twoalgorithm this algorithm is also having the same constraint.

MOTION BASE:

When useof video sequence is available, motion informationcan be used to locate moving objects. Movingsilhouettes like face and body parts can be extractedby simply thresholding accumulated framedifferences . Besides face regions, facial featurescan be located by frame differences .

Gray Scale Base:

Gray information within a face can also be treat as important features. Facial features such as eyebrows, pupils, and lips appear generally darker than their surrounding facial regions. Various recent feature extraction algorithms search for local gray minima within segmented facial regions. In these algorithms, the input image sare first enhanced by contrast-stretching and gray-scale morphological routines to improve the quality of local dark patches and thereby make detection easier. The extraction of dark patches is achieved by low-level gray-scale thresholding. Based method and consist three levels. Yang and huang presented new approach i.e. faces gray scale behaviour in pyramid (mosaic) images. This system utilizes hierarchical

Face location consist three levels. Higher two level based on mosaic images at different resolution. In the lower level, edge detection method is proposed. Moreover this algorithms gives fine response in complex
background where size of the face is unknown

Edge Base:

Face detection based on edges was introduced by Sakai et al. This workwas based on analysing line drawings of the faces from photographs, aiming to locate facialfeatures. Than later Craw et al. proposed a hierarchical framework based on Sakai et al.'swork to trace a human head outline. Then after remarkable works were carried out by many researchers in this specific area. Method suggested by Anila and Devarajan was very simple and fast. They proposed frame work which consist three stepsi.e. initially the images are enhanced by applying median filterfor noise removal and histogram equalization for contrast adjustment. In the second step the edge imageis constructed from the enhanced image by applying sobel operator. Then a novel edge trackingalgorithm is applied to extract the sub windows from the enhanced image based on edges. Further they used Back propagation Neural Network (BPN) algorithm to classify the sub-window as either face or non-face.

FEATURE ANALYSIS

These algorithms aimto find structural features that exist even when thepose, viewpoint, or lighting conditions vary, andthen use these to locate faces. These methods aredesigned mainly for face localization

Feature Searching

Viola Jones Method:

Paul Viola and Michael Jones presented an approach for object detection which minimizes computation time while achieving high detection accuracy. Paul Viola and Michael Jones [39] proposed a fast and robust method for face detection which is 15 times quicker than any technique at the time of release with 95% accuracy at around 17 fps.The technique relies on the use of simple Haar-like features that are evaluated quickly through the use of a new image representation. Based on the concept of an ―Integral Image it generates a large set of

features and uses the boosting algorithm AdaBoost to reduce the overcomplete set and the introduction of a degenerative tree of the boosted classifiers provides for robust and fast interferences. The detector is applied in a scanning fashion and used on gray-scale images, the scanned window that is applied can also be scaled, as well as the features evaluated.
Gabor Feature Method:

Sharif et al proposed an Elastic Bunch Graph Map (EBGM) algorithmthat successfullyimplements face detection using Gabor filters. The proposedsystem applies 40 different Gabor filters on an image. As aresult of which 40 images with different angles and orientationare received. Next, maximum intensity points in each filteredimage are calculated and mark them as fiducial points. Thesystem reduces these points in accordance to distance betweenthem. The next step is calculating the distances between thereduced points using distance formula. At last, the distances arecompared with database. If match occurs, it means that thefaces in the image are detected.

CONSTELLATION METHOD

All methods discussed so far are able to track faces but still some issue like locating faces of various poses in complex background is truly difficult. To reduce this difficulty investigator form a group of facial features in face-like constellations using more robust modelling approaches such as statistical analysis. Various types of face constellations have been proposed by Burl et al. . They establish use of statistical shape theory on the features detected from a multiscale Gaussian derivative filter. Huang et al. also apply a Gaussian filter for pre-processing in a framework based on image feature analysis.Image Base Approach.

Neural Network
Neural networks gaining much more attention in many pattern recognition problems, such as OCR, object recognition, and autonomous robot driving. Since face detection can be treate a two class pattern recognition problem, various neural network algorithms have been proposed.
The advantage of using neural networks for face detection is the feasibility of training a

system to capture the complex class conditional density of face patterns. However, one demerit is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance. In early days most hierarchical neural network was proposed by Agui et al. [43]. The first stage having twoparallel subnetworks in which the inputs are filtered intensity valuesfrom an original image. The inputs to the second stagenetwork consist of the outputs from the sub networks andextracted feature values. An output at thesecond stage shows the presence of a face in the inputregion.Propp and Samal developed one of the earliest neuralnetworks for face detection [44]. Their network consists offour layers with 1,024 input units, 256 units in the first hiddenlayer, eight units in the second hidden layer, and two outputunits.Feraud and Bernier presented a detection method using auto associative neural networks [45], [46], [47]. The idea is based on [48] which shows an auto associative network with five layers is able to perform a nonlinear principal component analysis. One auto associative network is used to detect frontal- view faces and another one is used to detect faces turned up to 60 degrees to the left and right of the frontal view. After that Lin et al. presented a face detection system using probabilistic decision-based neural network (PDBNN) [49]. The architecture of PDBNN is similar to a radial basis function (RBF) network with modified learning rules and probabilistic interpretation.

LINEAR SUB SPACE METHOD

Eigen faces Method:

An early example of employing eigen vectors in face recognition was done by Kohonen in which a simple neural network is demonstrated to perform face recognition for aligned and normalized face images. Kirby and Sirovich suggested that images of faces can be linearly encoded using a modest number of basis images. The idea is arguably proposed first by Pearson in 1901 and then byHOTELLING in 1933 .Given a collection of n by m pixel training.

Images represented as a vector of size m X n, basis vectors spanning an optimal subspace are determined such that the mean square error between the projection of the training images onto this subspace and the original images is minimized.They call the set of optimal basis vectors Eigen pictures since these are simply the eigen vectors of the covariance matrix computed from the vectorized face images in the training set.Experiments with a set of 100

images show that a face image of 91 X 50 pixels can be effectively encoded using only50 Eigen pictures.

STATISTICAL APPROCH

Support Vector Machine (SVM):

SVMs were first introduced Osuna et al. for face detection. SVMs work as a new paradigm to train polynomial function, neural networks, or radial basis function (RBF) classifiers.SVMs works on induction principle, called structural risk minimization, which targets to minimize an upper bound on the expected generalization error. An SVM classifier is a linear classifier where the separating hyper plane is chosen to minimize the expected classification error of the unseen test patterns.In Osunaet al. developed an efficient method to train an SVM for large scale problems,and applied it to face detection. Based on two test sets of 10,000,000 test patterns of 19 X 19 pixels, their system has slightly lower error rates and runs approximately30 times faster than the system by Sung and Poggio . SVMs have also been used to detect faces and pedestrians in the wavelet domain.

Problem

Over the past decade face detection and recognition have transcended from esoteric to popular areas of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. Because of the intrinsic nature of the problem, computer vision is not only a computer science area of research, but also the object of neuroscientific and psychological studies also, mainly because of the general opinion that advances in computer image processing and understanding research will provide insights into how our brain work and vice versa.

A general statement of the face recognition problem (in computer vision) can be formulated as follows: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces. Facial recognition generally involves two stages:

Face Detection where a photo is searched to find a face, then the image is processed to crop and extract the person's face for easier recognition.

Face Recognition where that detected and processed face is compared to a database of known faces, to decide who that person is. Since 2002, face detection can be performed fairly easily and reliably with Intel's open source framework called OpenCV . This framework has an inbuilt Face Detector that works in roughly 90-95% of clear photos of a person looking forward at the camera. However, detecting a person's face when that person is viewed from an angle is usually harder, sometimes requiring 3D Head Pose Estimation. Also, lack of proper brightness of an image can greatly increase the difficulty of detecting a face, or increased contrast in shadows on the face, or maybe the picture is blurry, or the person is wearing glasses, etc.

Face recognition however is much less reliable than face detection, with an accuracy of 30-70% in general. Face recognition has been a strong field of research since the 1990s, but is still a far way away from a reliable method of user authentication. More and more techniques are being developed each year. The Eigenface technique is considered the simplest method of accurate face recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate.

OpenCV was started at Intel in 1999 by Gary Bradski for the purposes of accelerating research in and commercial applications of computer vision in the world and, for Intel, creating a demand for ever more powerful computers by such applications. Vadim Pisarevsky joined Gary to manage Intel's Russian software OpenCV team. Over time the OpenCV team moved on to other companies and other Research. Several of the original team eventually ended up working in robotics and found their way to Willow Garage. In 2008, Willow Garage saw the need to rapidly advance robotic perception capabilities in an open way that leverages the entire research

and commercial community and began actively supporting OpenCV, with Gary and Vadim once again leading the effort . Intel's open-source computer-vision library can greatly simplify computervision programming. It includes advanced capabilities - face detection, face tracking, face recognition, Kalman filtering, and a variety of artificialintelligence (AI) methods - in ready-touse form. In addition, it provides many basic computer-vision algorithms via its lower-level APIs.

OpenCV has the advantage of being a multi-platform framework; it supports both Windows and Linux, and more recently, Mac OS X. OpenCV has so many capabilities it can seem overwhelming at first. A good understanding of how these methods work is the key to getting good results when using OpenCV. Fortunately, only a select few need to be known beforehand to get started.

OpenCV's functionality that will be used for facial recognition is contained within several modules. Following is a short description of the key namespaces:

CXCORE namespace contains basic data type definitions, linear algebra and statistics methods, the persistence functions and the error handlers. Somewhat oddly, the graphics functions for drawing on images are located here as well.

CV namespace contains image processing and camera calibration methods. The computational geometry functions are also located here.

CVAUX namespace is described in OpenCV's documentation as containing obsolete and experimental code. However, the simplest interfaces for face recognition are in this module. The code behind them is specialized for face recognition, and they're widely used for that purpose.

ML namespace contains machinelearning interfaces.

HighGUI namespace contains the basic I/O interfaces and multi-platform windowing capabilities.

CVCAM namespace contains interfaces for video access through DirectX on 32- bit Windows platforms.

Eigenfaces is considered the simplest method of accurate face recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate. Most resources on face recognition are for basic Neural Networks, which usually don't work as well as Eigenfaces does. And unfortunately there are only some basic explanations for better type of face recognition than Eigenfaces, such as recognition from video and other techniques at the Face Recognition Homepage and Active Appearance Models . But for other techniques, you should read some recent computer vision research papers from CVPR

and other computer vision conferences. Most computer vision or machine vision conferences include new advances in face detection and face recognition that give slightly better accuracy.

Proposed Solution:

When image quality is taken into consideration, there is a plethora of factors that influence the system's accuracy. It is extremely important to apply various image pre-processing techniques to standardize the images that you supply to a face recognition system. Most face recognition algorithms are extremely sensitive to lighting conditions, so that if it was trained to recognize a person when they are in a dark room, it probably won't recognize them in a bright room, etc. This problem is referred to as "lumination dependent", and there are also many other issues, such as the face should also be in a very consistent position within the images (such as the eyes being in the same pixel coordinates), consistent size, rotation angle, hair and makeup, emotion (smiling, angry, etc), position of lights (to the left or above, etc). This is why it is so important to use a good image preprocessing filters before applying face recognition. You should also do things like removing the pixels around the face that aren't used, such as with an elliptical mask to only show the inner face region, not the hair and image background, since they change more than the face does. For simplicity, the face recognition system presented in this paper is Eigenfaces using grayscale images. The paper demonstrates how easily is to convert color images to grayscale (also called 'grayscale'), and then to apply Histogram Equalization [8] as a very simple method of automatically standardizing the brightness and contrast of your facial images. For better results, you could use color face recognition (ideally with color histogram fitting in HSV or another color space instead of RGB), or apply more processing stages such as edge enhancement, contour detection, motion detection, etc. Also, this code is resizing images to a standard size, but this might change the aspect ratio of the face. In [9] is described a method on how to resize an image while keeping its aspect ratio the same. OpenCV uses a type of face detector called a Haar Cascade classifier. Given an image, which can come from a file or from live video, the face detector examines each image location and classifies it as "Face" or "Not Face." Classification assumes a fixed scale for the face, say 50x50 pixels. Since faces in an image might be smaller or larger than this, the classifier runs over the image several times, to search for faces across a range of scales. This may seem an enormous amount of processing, but thanks to algorithmic tricks, explained in the sidebar, classification is very fast, even when it's applied at

several scales. The classifier uses data stored in an XML file to decide how to classify each image location. The OpenCV download includes four flavors of XML data for frontal face detection, and one for profile faces. It also includes three non-face XML files - one for full body (pedestrian) detection, one for upper body, and one for lower body.

CHAPTER-3
REQUIRED TOOLS

Open CV-It is a library of programming functions mainly aimed at real-time computer vision.

Python-It is the language that is used in the backend part for the implementation

Spyder - It is a free and open source scientific environment written in python, for python, and designed by and for scientists,engineers and data analysts.

CHAPTER-4
SYSTEM REQUIREMENTS

1.4 GB RAM (Minimum)

2.80 GB HDD

3.Dual Core processor

4.CDROM (installation only). VGA resolution monitor

5.Microsoft Windows 98/2000/NT with service pack 6 / XP with service pack 2/ Windows 7 with

6.service pack 2

7.SQL Server 2008 R2

CHAPTER-5

SYSTEM ANALYSIS

To improve the ability of detection, we usually generate a small set of features from the unique input variables by feature extraction. The conventional Convolutional Neural Networks is used to project the detections in largeimages. But it need some more advanced techniques to get better result.

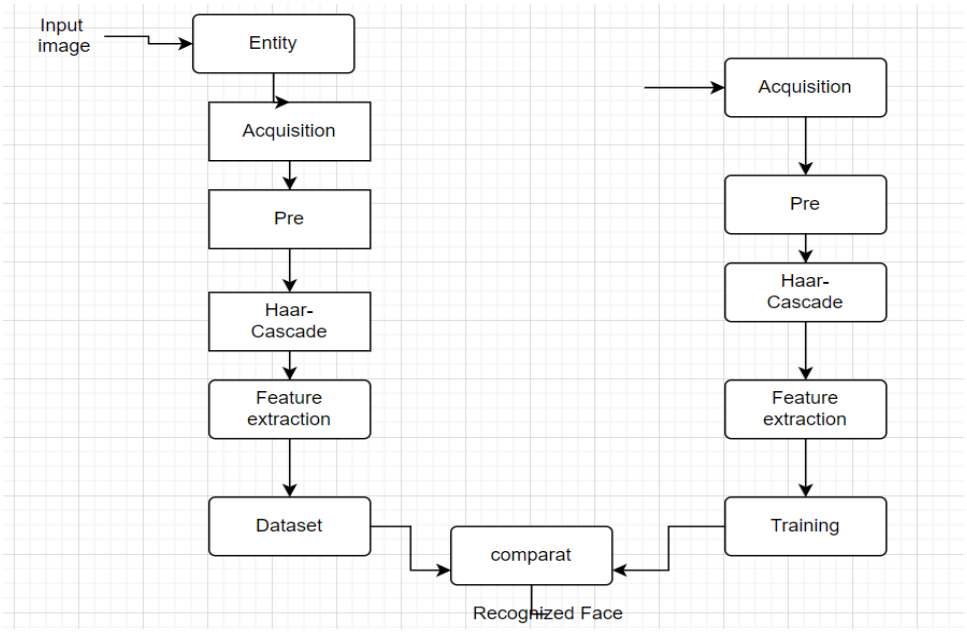
DISADVANTAGES:

- Low efficiency.
- Time consuming.
- High complexities.
- Resources consuming

PROPOSED METHOD:

The proposed technique deploys two progresses of images such as the input images and the image capturedthrough live streaming. Both these process undergoes four common procedures namely, face acquisition, pre- processing, face detection using Haarcascade classifier and feature extraction using Linear Binary Patternalgorithm to compute LBP values. These values are stored in the database only in case of processing an inputimage. Finally, comparison of the values in the database with the values computed via live streaming takes placewhich recognizes the human face as known or unknown based on the matching.

Flow of the project:



ADVANTAGES:

- High efficiency.
- Time Saving.

CHAPTER-6

DESIGN

Face detection is generally the first step towards many face-related applications like face recognition or face verification. But, face detection has very useful applications. One of the most successful applications of face detection is probably “photo-taking”.

Example: When you click a photo of your friends, the camera in which the face detection algorithm has built-in detects where the faces are and adjusts focus accordingly.

Overview of Face Recognition

Generally, Face Recognition is a method of identifying or verifying the *identity* of an individual by using their face. Various algorithms are there for face recognition but their accuracy might vary. Here I am going to discuss with you that how we can do face recognition using deep learning.

Now let us understand how we can recognize faces using deep learning. Here we use face embeddings in which every face is converted into a vector. The technique of converting the face into a vector is called *deep metric learning*. Let me divide this process into three simple steps for better and easy understanding:

Face Detection:

The first task that we perform is detecting faces in the image(photograph) or video stream. Now we know that the exact coordinates/location of the face, so we extract this face for further processing.

Feature Extraction:

Now see we have cropped out the face from the image, so we extract specific features from it. Here we are going to see how to use face embeddings to extract these features of the face. As we know a neural network takes an image of the face of the person as input and outputs a vector that represents the most important features of a face! In machine learning, this vector is nothing but called *embedding* and hence we call this vector ***face embedding.***

When we train the neural network, the network learns to output *similar vectors* for faces that look similar. Let us consider an example, if I have multiple images of faces within different timelapse, it's obvious that some features may change but not too much. So in this problem, the vectors associated with the faces are similar or we can say they are very close in the vector space.

Up to this point, we came to know how this network works, let us see how to use this network on our own data. Here we pass all the images in our data to this pre-trained network to get the respective embeddings and save these embeddings in a file for the next step.

3. Comparing Faces:

We have face embeddings for each face in our data saved in a file, the next step is to *recognize* a new image that is *not in our data*. Hence the first step is to compute the face embedding for the image using the same network we used earlier and then compare this embedding with the rest of the embeddings that we have. We recognize the face if the generated embedding is closer or similar to any other embedding.

STAGES OF EXECUTING FACIAL RECOGNITION AND DETECTION

Facial recognition is a complex process – the app should complete several tasks to deliver the final conclusion.

- Detecting the face: modern cameras have a precise vision and are equipped with automatic facial detectors. You need to download tools that will connect your app to the device's built-in camera and edit its settings and use image libraries. We use TensorFlow to detect objects real-time and Caffee to track detected faces.
- Identification of reference points: the app should detect particular features in the picture. Traditionally, eyes are the main identification point, but developers usually introduce up to 70 points to their solutions. You'll have to run manual tests and map out these spots, so the software can later perform these actions automatically. You can use the Dlib function and create a vector image with 128 dimensions.
- Frontal face modeling: if the face was pictured from non-direct angles, the application should be able to remodel the picture to get the frontal image. This is done by combining the reference points from original pictures into a single portrait.

- Calculating the descriptor: developers introduce the face vector: the type of characteristics that describe a person's face mathematically, as a combination of points, lengths, and quantitative characteristics. It also identifies such crucial variables as hairstyle, gender, age, race, etc.
- Comparing faces: the next step was applying Euclidean distance to cross-match the detected faces with others in the database. If there were no matches found, the system will create a unique ID for this face. If the image was matched with already existing inputs, the system will show an ID that corresponds to the vector.
- Facial recognition: for a new ID, you'll be asked to create a brand new profile – this is accomplished via an Admin panel. If there are matches that correspond to the input, you will receive a detailed personal profile with personal data and status. If a person is identified in the database as potentially dangerous, you'll want the system to send emergency alerts. It can be set up with a chatbot alert tool on the base of Microsoft Bot Framework or Python-based AI tools.

Fundamental steps in image processing are

1. Image acquisition: to acquire a digital image

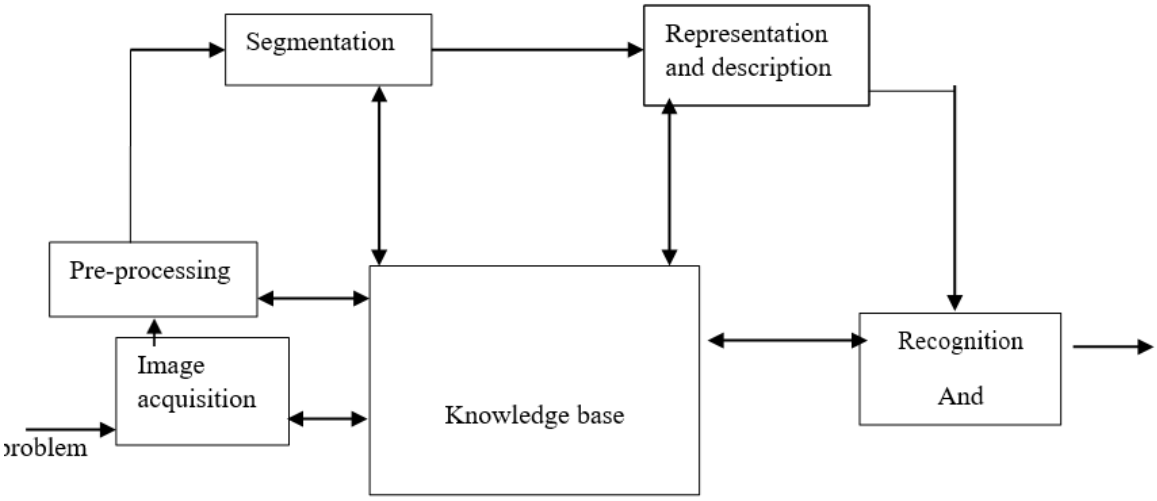
2. Image pre-processing: to improve the image in ways that increases the chances for success of the other processes.

3. Image segmentation: to partitions an input image into its constituent parts of objects.

4. Image segmentation: to convert the input data to a from suitable for computer processing.

5. Image description: to extract the features that result in some quantitative information of interest of features that are basic for differentiating one class of objects from another.

6. Image recognition: to assign a label to an object based on the information provided by its description.



USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

Use case diagram components

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

Overview of the properties of the actor:

- is human or non-human
- represents roles
- can specify multiplicity
- communicates with the system (connected to at least one use case)

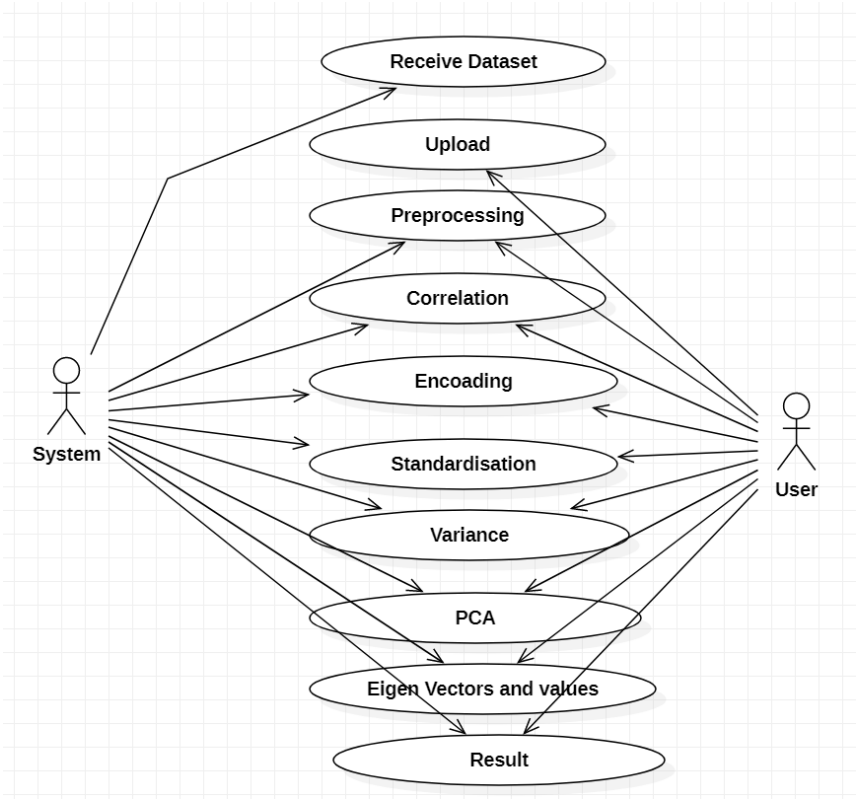
- uses the system (triggers use cases)
- is used by the system (to realize use cases)

Overview of the properties of the use case:

- describes the behavior of the system under development
- represents the perspectives of the user
- can specify multiplicity
- doesn't consider the details of the system behavior
- can be abstract (not feasible itself, only through other use cases)

Use case diagrams can be used for −

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.


Receive Dataset
Upload
Preprocessing
Correlation
Encoading
Standardisation
Variance
PCA
Eigen Vectors and values
Result
System
User

ER DIAGRAM

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

ER Diagrams Symbols & Notations

Entity Relationship Diagram Symbols & Notations mainly contains three basic symbols which are rectangle, oval and diamond to represent relationships between elements, entities and attributes. There are some sub-elements which are based on main elements in ERD Diagram. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Following are the main components and its symbols in ER Diagrams:

- **Rectangles:** This Entity Relationship Diagram symbol represents entity types
- **Ellipses :** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes

Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets.

Different types of cardinal relationships are:

- One-to-One Relationships
- One-to-Many Relationships
- May to One Relationships
- Many-to-Many Relationships

1.One-to-one:

- One entity from entity set X can be associated with at most one entity of entity set Y and vice versa.

2.One-to-many:

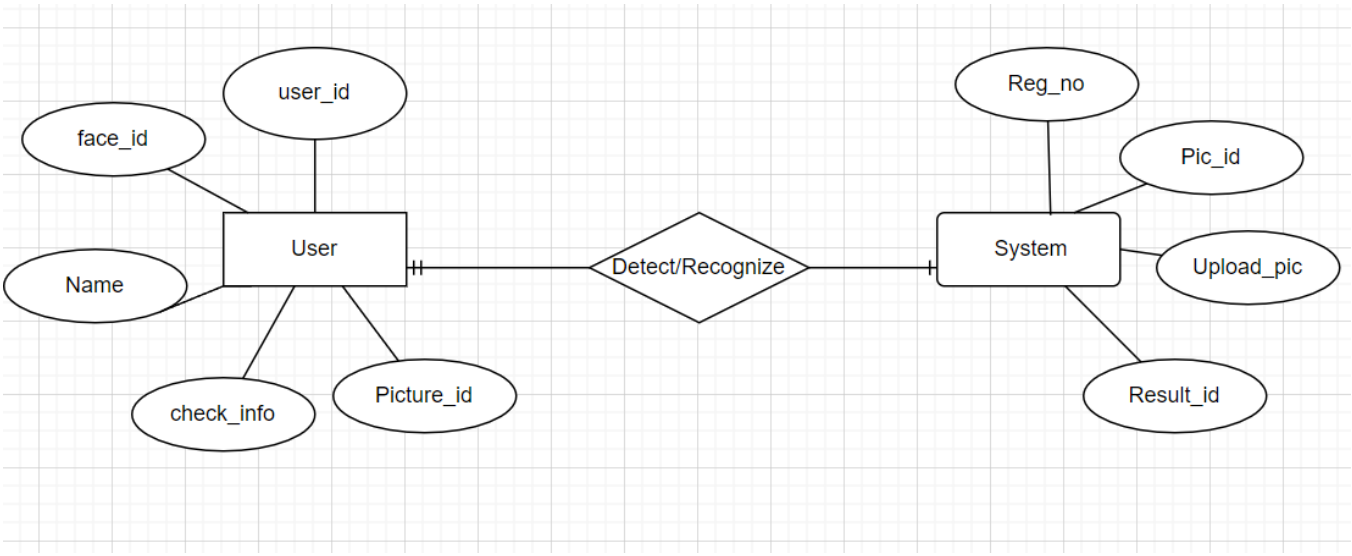
- One entity from entity set X can be associated with multiple entities of entity set Y, but an entity from entity set Y can be associated with at least one entity.

3. Many to One

- More than one entity from entity set X can be associated with at most one entity of entity set Y. However, an entity from entity set Y may or may not be associated with more than one entity from entity set X.

4.Many to Many:

- One entity from X can be associated with more than one entity from Y and vice versa.


user_id
face_id
User
Name
check_info
Picture_id
Detect/Recognize
Reg_no
Pic_id
System
Upload_pic
Result_id

CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

PURPOSE OF CLASS DIAGRAM

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

1. It analyses and designs a static view of an application.
2. It describes the major responsibilities of a system.
3. It is a base for component and deployment diagrams.
4. It incorporates forward and reverse engineering.

Benefits of class diagram

1. It can represent the object model for complex systems.
2. It reduces the maintenance time by providing an overview of how an application is structured before coding.
3. It provides a general schematic of an application for better understanding.
4. It represents a detailed chart by highlighting the desired code, which is to be programmed.
5. It is helpful for the stakeholders and the developers.

Basic components of a class diagram

The standard class diagram is composed of three sections:

- **Upper section:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
- **Middle section:** Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
- **Bottom section:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.

Member access modifiers

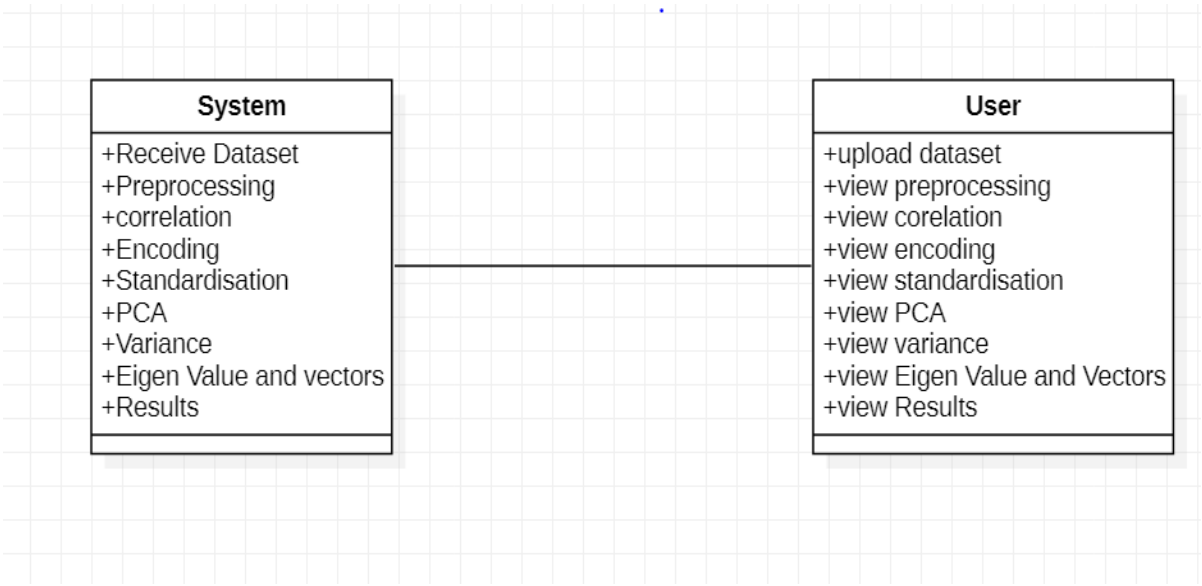
All classes have different access levels depending on the access modifier (visibility). Here are the access levels with their corresponding symbols:

- Public (+)
- Private (-)
- Protected (#)
- Package (~)
- Derived (/)
- Static (underlined)

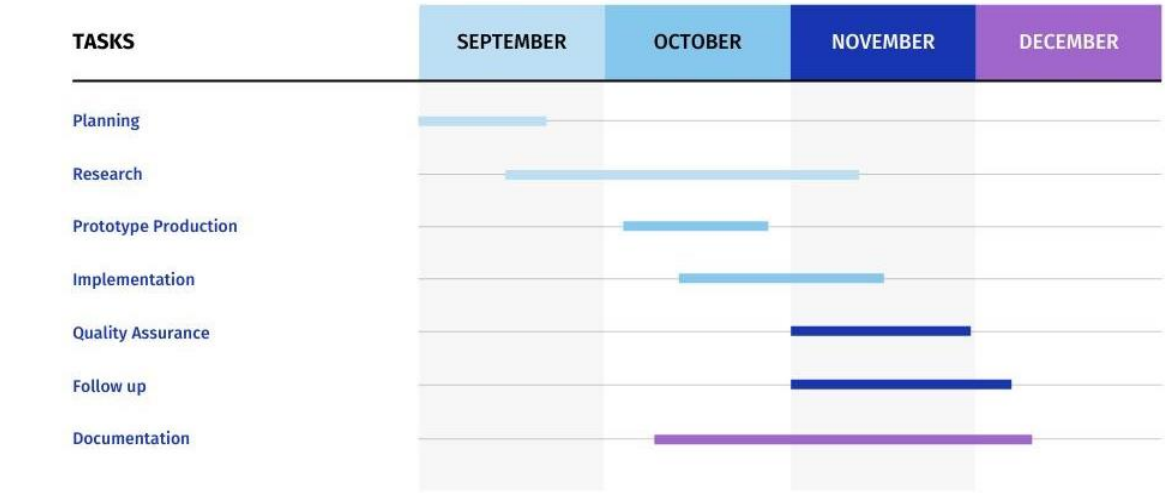
Member scopes

There are two scopes for members: classifiers and instances.

Classifiers are static members while instances are the specific instances of the class. If you are familiar with basic OO theory, this isn't anything groundbreaking.


System
+Receive Dataset
+Preprocessing
+correlation
+Encoding
+Standardisation
+PCA
+Variance
+Eigen Value and vectors
+Results
User
+upload dataset
+view preprocessing
+view corelation
+view encoding
+view standardisation
+view PCA
+view variance
+view Eigen Value and Vectors
+view Results

CHAPTER-7
GANTT CHART



CHAPTER-8
IMPLEMENTATION

• Implemented using pretrained open cv to detect the facial features of nose, eyes, and the corners of the mouth. The coordinates are returned and used to outline with a rectangle where a face is within the image.

• Next, a database of peoples' faces who a user wants to recognize is composed of a unique vector of numbers called a face embedding. When an image or video is passed through the algorithm, it breaks down the detected face into a new embedding and uses Euclidean distances to compare the new face to any faces in the database, looking for a minimum distance to be a match.

• If a match is found, the face is recognized, and the name is displayed on the image. This math was implemented using pretrained models in an API.

• This method is prone to spoofing (pictured below), where someone could hold an image of a person in front of the camera, and it will be recognized as the person standing there when they are not, which could cause a security breach • This is solved using 3D data capture of the Intel Realsense D435i. A point cloud is created of a user's face. This is then filtered and aligned to accommodate for different angles of capture, environments, and distance from the camera. All null points are then removed, and the image is cropped for unified formatting across the database.

• This is done on every user to create a database of registered users. When a new face is scanned, the Euclidean distances are compared to registered users, and if there is a match within a specified tolerance the output is that person's name. This process was used both to verify the 2D output, and to ensure a person was standing in real time in front of the camera.

CHAPTER-9
MODULE DESCRIPTION

In this work, a software for human face detection and recognition is prepared. Initial implementation of this work is service oriented. However, the analysis and design are done to promote the work as product based. The work is entitled as Face Detection and Recognition System. Face Detection is a application software to deal with human face. It has the provisions to collect image from the user so that they can detect the eyes, nose, mouth and whole face of human in the image.

Face Recognition:

STEP 1: FACE DETECTION

First, a camera will detect and recognize a human's face – one that can either be in a crowd or alone. It is most easily detected when the person is looking straight at the camera. However, modern technological advances allow face recognition software to still work if the person's face is angled slightly.

STEP 2: FACE ANALYSIS

After detection and recognition, a photo will capture the face and will then be analyzed. The majority of face recognition technology use 2D images instead of 3D. This is because 2D photos are more readily correlated with public photos or pictures in a database (these are typically 2D as well). During analysis, the face will be separated into distinguishable landmarks – we can call these nodal points. A human face has eight nodal points. Face recognition technology will analyze each of these points – for example, the distance between your eyebrows.

STEP 3: CONVERTING AN IMAGE INTO DATA

After analysis, each nodal point becomes a number in the application database. The entire numerical code is referred to as a faceprint. Just like how everybody has a unique thumbprint, everyone also has a unique faceprint.

STEP 4: MATCHING

The final step of the process is finding a match. Your faceprint is compared to a database of other facial codes. The number of faces that are compared depends on the database and how many databases the software has access to. For instance, the FBA has access to 21 state databases, with 641 million photos across them. The facial recognition technology then identifies a match for your exact facial features – it returns the user with the found match and other relevant information, such as an address and a name.

Firstly We have to install opencv and python
Install all the libraries that are needed
Import CV2 and face recognition
CV2.__Version__
OpenCv

- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.
- Face_Recognition

When you install face_recognition, you get a simple command-line program called face_recognition that you can use to recognize faces in photograph or folder full for photographs.

- Numpy

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working

with ndarray very easy.

Command: pip install numpy

- Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

pip install imutils

- VideoStream
- Time: This module provides various time-related functions. For related functionality, see also the datetime and calendar modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

- Os:

The OS module in Python provides a way of using operating system dependent functionality. The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux.

```
import os
os.getcwd()
```

- Image:

The Image module provides a class with the same name which is used to represent a PIL image. The module also provides a number of factory functions, including functions to load images from files, and to create new images.

- Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

matplotlib.pyplot

- Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software.

Spyder is extensible with first-party and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Features include:

- An editor with syntax highlighting, introspection, code completion
- Support for multiple IPython consoles
- The ability to explore and edit variables from a GUI
- A Help pane able to retrieve and render rich text documentation on functions, classes and methods automatically or on-demand
- A debugger linked to IPdb, for step-by-step execution
- Static code analysis, powered by Pylint
- A run-time Profiler, to benchmark code

CHAPTER-10
PURPOSE OF THE APPLICATION

Whenever we implement a new system it is developed to remove the shortcomings of the existing system. The computerized mechanism has the more edge than the manual system. The existing system is based on manual system which takes a lot of time to get performance of the work. The proposed system is a web application and maintains a centralized repository of all related information. The system allows one to easily access the software and detect what he wants.

GOALS OF PROPOSED SYSTEM:

a. Planned approach towards working: - The working in the organization will be well planned and organized. The data i.e. Image will be stored properly in database stores which will help in retrieval of information as well as its storage.

b. Accuracy: - The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

c. Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

d. No Redundancy: - In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

e. Immediate retrieval of information: - The main objective of proposed system is to provide for a quick and efficient detection of required information. Any type of detection would be available whenever the user requires. f. Immediate storage of information: - In manual system there are many problems to store the largest amount of information for processing.

g. Easy to Operate: - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the people.

CHAPTER-11
CASE STUDY OF FACE RECOGNITON

Face recognition is a technology capable of identifying or verifying a subject through an image, video, or any audiovisual element of his face. Generally, this identification is used to access an application, system, or service. It is a method of biometric identification that uses that body measures, in this case, face and head, to verify the identity of a person through its facial biometric pattern and data. The technology collects a set of unique biometric data of each person associated with their face and facial expression to identify, verify and/or authenticate a person. The procedure simply requires any device that has digital photographic technology to generate and obtain the images and data necessary to create and record the biometric facial pattern of the person that needs to be identified.

Unlike other identification solutions such as passwords, verification by email, selfies or images, or fingerprint identification, Biometric facial recognition uses unique mathematical and dynamic patterns that make this system one of the safest and most effective ones.

Challenges Faced by Facial Recognition System

Illumination

Illumination stands for light variations. The slight change in lighting conditions cause a significant challenge for automated face recognition and can have a significant impact on its results. If the illumination tends to vary, the same individual gets captured with the same sensor and with an almost identical facial expression and pose, the results that emerge may appear quite different.

Illumination changes the face appearance drastically. It has been found that the difference between two same faces with different illuminations is higher than two different faces taken under same illumination.

Pose

Facial Recognition Systems are highly sensitive to pose variations. The pose of a face varies when the head movement and viewing angle of the person changes. The movements of head or differing POV of a camera can invariably cause changes in face appearance and generate intra-class variations making automated face recognition rates drop drastically. It becomes a challenge to identify the real face when the rotation angle goes higher. It may result in faulty recognition or no recognition if the database only has the frontal view of the face.

Occlusion

Occlusion means blockage, and it occurs when one or other parts of the face are blocked and whole face is not available as an input image. Occlusion is considered one of the most critical challenges in face recognition system.

It occurs due to beard, moustache, accessories (goggle, cap, mask, etc.), and it is prevalent in real-world scenario. The presence of such components makes the subject diverse and hence making automated face recognition process a tough nut to crack.

Expressions

Face is one of the most crucial biometrics as its unique features play a crucial role in providing human identity and emotions. Varying situations cause different moods which result in showing various emotions and eventually change in facial expressions.

Different expressions of the same individual are another significant factor that needs to be taken into account. Human expressions are particularly macro-expressions which are happiness, sadness, anger, disgust, fear, surprise. Micro-expressions are the one which shows the rapid facial patterns and happen involuntarily.

Macro and micro expressions find their place on someone's face due to changes in one's emotional state and in the wake of such emotions- which are many- the efficient recognition becomes difficult.

More recently, the Department of Homeland Security unveiled its "Biometric Exit" plan, which aims to use facial recognition technology on nearly all air travel passengers by 2023, to identify compliance with visa status.

Perhaps surprisingly, 59% of Americans are actually in favour of implementing facial recognition technology, considering it acceptable for use in law enforcement according to a Pew Research survey. Yet, some cities such as San Francisco have pushed to ban surveillance, citing a stand against its potential abuse by the government.

Facial recognition technology can potentially come in handy after a natural disaster. After Hurricane Dorian hit in late summer of 2019, the Bahamas launched a blockchain-based missing persons database "FindMeBahamas" to identify thousands of displaced people.

Europe

Belgium and Luxembourg are two of only three governments in the world to officially oppose the use of facial recognition technology.

*"*The EU has been a haven for unlawful biometric experimentation and surveillance."

— European Digital Rights (EDRi)

In Russia, authorities have relied on facial recognition technology to check for breaches of quarantine rules by potential COVID-19 carriers. In Moscow alone, there are reportedly over 100,000 facial recognition enabled cameras in operation.

Conclusion

Face is the most essential part of the human body, and its unique features make it even more crucial in identifying humans. Various algorithms and technologies are used worldwide to make the face recognition process more accurate and reliable. The applications of this ever-growing technology are also expanding in healthcare, security, defence, forensic, and transportation, requiring more accuracy. However, some challenges are ubiquitous while developing face recognition technology such as pose, occlusion, expressions, ageing, etc, which have been discussed

above in the article.

CHAPTER-12

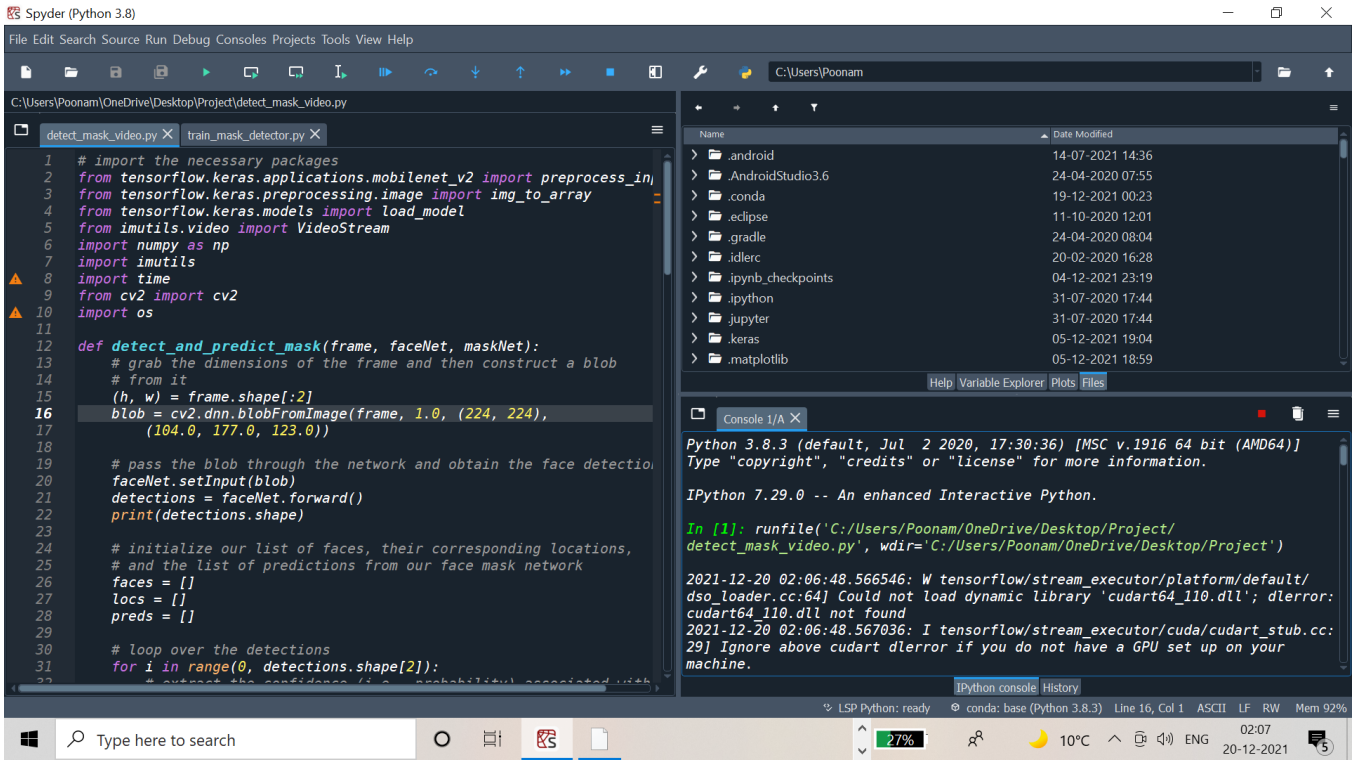
RESULTS AND DISCUSSION

In the end, integrating such new technology into an application was more difficult than initially considered. Most problems encountered were hard to find solutions to because the computer vision community was lacking experience with the concepts. However, through lots of research, some solutions were found as seen above. Starting with a base foundation of machine learning concepts and mathematical understanding was extremely beneficial in adapting to future concepts and problems. We were successful at implementing and understanding the theory of 2D facial detection and recognition but fell a bit short when processing 3D data. The concepts were well understood, but the coding implementation proved challenging given the circumstances, and a proper database of registered users was not able to be created. However, using the pointcloud method, Brendan was able to have his face recognized first in 2D and verified with 3D data as a registered user inconsistently. We successfully filtered input images and aligned them to a grid, but holes of imperfect image capture are still too frequently present, and the image cropping must be done manually rather than with code. Handling null points seems to be the biggest issue were we to move forward, and more time, testing, and users would be needed in the database to fully finish the system.

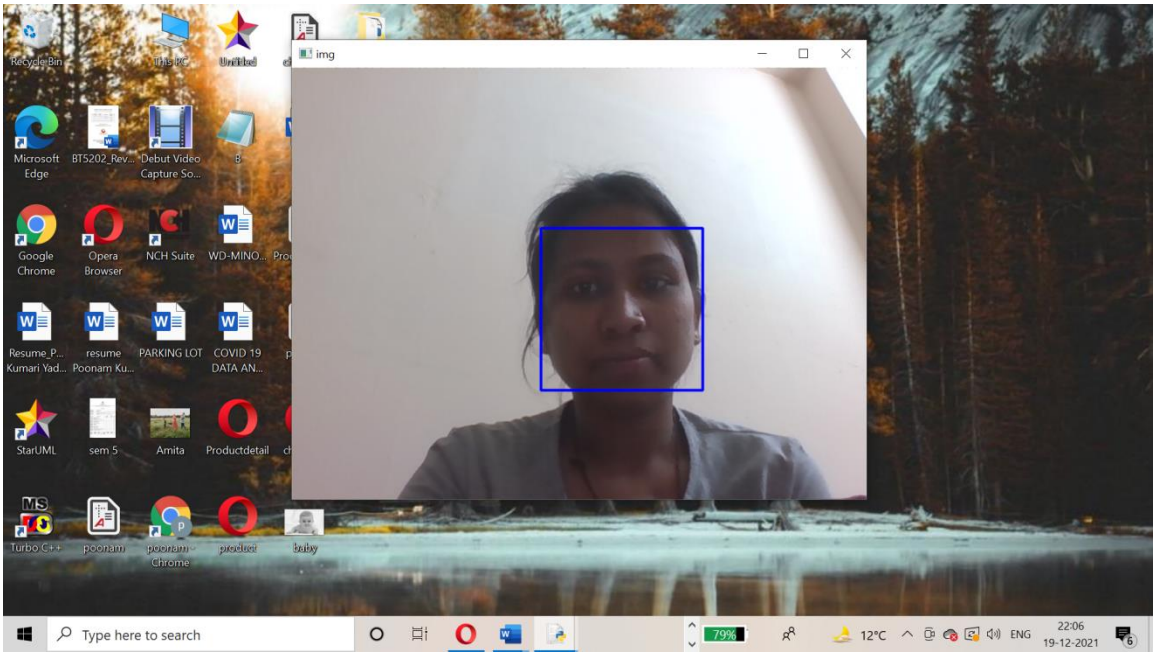
Code:

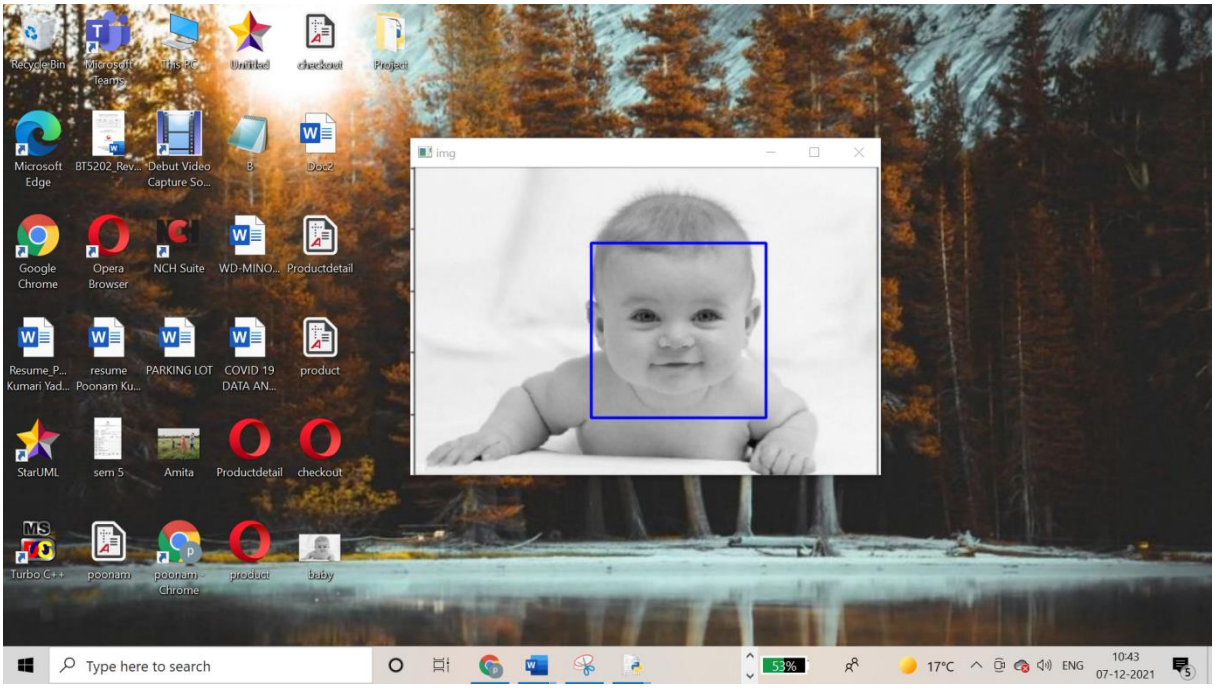
```
import cv2

# Load the cascade
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# Read the input image
img = cv2.imread('baby.jpg')
# Convert into grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Detect faces
```

```
faces = face_cascade.detectMultiScale(gray, 1.1, 4)
# Draw rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
# Display the output
cv2.imshow('img', img)
cv2.waitKey()
```



```
detect_face.py - C:\face recognition\detect_face.py (3.10.0)
File  Edit  Format  Run  Options  Window  Help
import cv2

# Load the cascade
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# Read the input image
img = cv2.imread('baby.jpg')
# Convert into grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Detect faces
faces = face_cascade.detectMultiScale(gray, 1.1, 4)
# Draw rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
# Display the output
cv2.imshow('img', img)
cv2.waitKey()
```




Recycle Bin
Microsoft Teams
This PC
Untitled
checkout
Project
Microsoft Edge
BT5202_Rev...
Debut Video Capture So...
8
Doc2
Google Chrome
Opera Browser
NCH Suite
WD-MINO...
Productdetail
Resume_P... Kumari Yad...
resume Poonam Ku...
PARKING LOT
COVID 19 DATA AN...
product
StarUML
sem 5
Amita
Productdetail
checkout
Turbo C++
poonam
poonam - Chrome
product
baby
img
Type here to search
53%
17°C
ENG
10:43
07-12-2021

CHAPTER-13

CONCLUSION

The computational models, which were implemented in this project, were chosen after extensive research, and the successful testing results confirm that the choices made by the researcher were reliable.The system with manual face detection and automatic face recognition did not have a recognition accuracy over 90%, due to the limited number of eigenfaces . The fully automated frontal view face detection system displayed virtually perfect accuracy .

The automated vision systems implemented in this thesis did not even approach the performance, nor were they as robust as a human's innate face recognition system. However, they give an insight into what the future may hold in computer vision.To build a reliable facial recognition tool, you will need access to facial databases and cameras with high resolution and vision capacities. You have to set up backend operations that will enable information exchange between the database and the software. You'll have to send a lot of confidential information back-and-forth, so security is a priority. We recommend using end-to-end data encryption and run security tests on each development stage.

Our vision of facial recognition tools is that they are powerful assets for enhancing global security and customer understanding. Paired with AI, biometrics, and Machine Learning, businesses can create smart detection solutions that get better with every analyzed face. It's a simple but universal technology that has the potential to impact just about all areas of daily life – for business owners and customers alike.

CHAPTER-14

FUTURE SCOPE

Face recognition is the technique in which is used to identity face that an be identified using ones individual face. Such kind of systems is used in photos, videos, or in the real time machines. With the help of such a technology one's can face can be easily detected by the help of dataset in similar matching appearance of a person. Due to the rising need for the systems that can help in the areas such as surveillance as well as security ,this kind of individual authentication can no longer be performed by using simple handmade methods hence there is a rising need of the automated systems that can easily rectify or correct the faults and process the human face recognition. When work is done by the machines tasks can be performed by it in very less duration of time and cuts off the major mistakes occurred by humans. A real time GUI based face recognition system built can make this work easy of face detection and can be achieved in various ways. And in the next coming years it will be of great use to restrict unauthorized access and to do everything in a secured like in when students give their exam so no body else can enter as well as seeing through the perspective of other things, so they will also be handled securely. The independent driving of construction vehicles could be a topic wherever little analysis has been performed. there's a substantial distinction between autonomous driving of the development vehicles at the construction web site and therefore the regular vehicles on the roads, because the surroundings is different, factors that have an effect on autonomous driving also will amendment and might be notably challenging. Therefore, future work may be done at the scaled construction site at PDRL laboratory, BTH., to develop autonomous vehicles mistreatment the results obtained from this research.

CHAPTER-15

REFERANCE

- Adelson, E. H., and Bergen, J. R. (1986) The Extraction of Spatio-Temporal Energy in
- Human and Machine Vision, Proceedings of Workshop on Motion: Representation and
- Analysis (pp. 151-155) Charleston, SC; May 7-9
- AAFPRS(1997). A newsletter from the American Academy of Facial Plastic and Reconstructive Surgery. Third Quarter 1997, Vol. 11, No. 3. Page 3.
- Baron, R. J. (1981). Mechanisms of human facial recognition. International Journal of Man Machine Studies, 15:137-178