

CHAPTER 1

INTRODUCTION

1.1 Introduction about face recognition

In this chapter, we review related works in classical and real-time face recognition. The purpose of this project is to develop an efficient face recognition system for real-time applications.

The face is our primary focus of attention in social life playing an important role in conveying identity and emotions. We can understand a number of faces found out at some point of our lifespan and pick out faces at a glance even after years of separation. This talent is pretty robust regardless of-of massive variations in visual stimulus due to converting circumstance, ageing and distractions inclusive of beard, glasses or modifications in coiffure.

Face popularity is an intrinsic part of the human visible perception and truly considered one of our core abilities. In the human brain, there are dedicated regions that offer us our awesome face popularity capabilities.

Face recognition is the technique in which the identity of a human being can be identified using one's individual face. Such kind of systems can be used in photos, videos, or in real time machines. The objective of this article is to provide a simpler and easy method in machine technology. With the help of such a technology one can easily detect the face by the help of dataset in similar matching appearance of a person. The method in which with the help of python and OpenCV in deep learning is the most efficient way to detect the face of the person. This method is useful in many fields such as the military, for security, schools, colleges and universities, airlines, banking, online web applications, gaming etc. this system uses powerful python algorithm through which the detection and recognition of face is very easy and efficient. The face recognition is a technique to identify or verify the face from the digital images or video frame. A human can quickly identify the faces without much effort. It is an effortless task for us, but it is a difficult task for a computer. There are various complexities, such as low resolution, occlusion, illumination variations, etc. These factors highly affect the accuracy of the computer to recognize the face more effectively. First, it is necessary to understand the difference between face detection and face recognition.

1.2 OpenCV

OpenCV (Open-Source Computer Vision) is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

When it is integrated with various libraries, such as NumPy, Python is capable of processing the OpenCV array structure for analysis. To identify image patterns and its various features, we use vector space and perform mathematical operations on these features.

1.3 Computer Vision

Computer vision is a process by which we can understand the images and videos, how they are stored, and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer Vision is playing a major role in self-driving cars, robotics, as well as in photo correction apps.

Face Detection: Face detection is generally considered as finding the faces (location and size) in an image and probably extracting them to be used by the face detection algorithm.

Face Recognition: The face recognition algorithm is used in finding features that are uniquely described in the image. The facial image is already extracted, cropped, resized, and usually converted to grayscale.

1.4 Objective:

The main aim of this report is to develop a system that will use the camera of the computer that would detect and recognize the person's face or the face of the individual using the tool in OpenCV called as the Open Face and Python programming language in the deep learning domain.

The objectives of Face Recognition for Real-Time Applications are given below:

- To enhance the Frame/sec for Face Recognition System, such that Recognition is done in Real Time.
- To achieve higher frames/sec or high-Resolution frames/sec

1.5 Features of Face Recognition:

- i. Unlocking Phones.
- ii. Law Enforcement.
- iii. Airport and Border Control.
- iv. Finding missing persons.
- v. Reducing crime rate.
- vi. Marketing.
- vii. Healthcare.

1.6 Motivation

The main problem of face recognition is its high dimension space, which is to be reduced by any dimension reduction techniques. The pattern recognition approach then tries to match the facial features, which are extracted from all the images present in the database. Therefore, there are two major problems one is feature extraction and then pattern recognition. Before this image, registration of all the faces is required to enhance the recognition rate of the whole system. So, these all motivates to search for a new method to solve all these problems and then integrate them to make a fully functional system with high accuracy.

1.7 Organization of Report

Chapter one contains the introduction of the Face Recognition which comprises information regarding the real time uses of Face Recognition and overview. This chapter also describes the motivation and scope of project.

Chapter two includes the literature review which refers to the study that has been carried out for face recognition applications over the previous years that have some relevance with existing models and various technologies which are used for proposed application. This chapter also specifies the aim and objectives.

Chapter three explains the tools and technologies which are used for development of face recognition.

Chapter four explains proposed approach and system architecture which includes the detail of how the proposed application has been developed with different components.

Chapter five discusses the implementation details of the application; this chapter explains the coding part developed for performing various functions.

Chapter six discusses the results that were generated from the proposed application and showcases every necessary output needed to describe the application precisely.

Chapter seven contains conclusion about the proposed application, and it also describes limitation of study and its future scope.

CHAPTER 2

LITERATURE REVIEW

Face recognition is one among the trending topics within the field of image processing and computer vision. Some examples include image retrieval, security and intelligence, OCR, medical imaging and agricultural monitoring. Traditionally, researchers used pattern recognition to predict faces supported prior face models. A break detection technology then was developed named as Viola Jones detector that was an optimized technique of using HAAR, digital image features utilized in visual perception.

However, it failed because it didn't perform well on faces in dark areas and non-frontal faces. There are various methods of object detection supported deep learning which are divided into two categories: one stage and two stages object detectors.

2.1 DNN

P. Gupta, N. Saxena, M. Sharma, J. Tripathi (2018) has published the paper on face recognition that introduces a new way of using a deep neural network (another type of deep network). In this proposed approach, only the extracted facial features are provided instead of providing raw pixel values as input. Facial features are being extracted with the help of Haar-Cascade and feeding these facial features rather than raw pixel values. As the number of redundant input features has been decreased, the complexity of the neural network-based recognition framework is also decreased. It also makes the process lighter and faster by using DNN instead of Convolutional Network. The proposed method does not compromise the accuracy of the framework, as average accuracy so obtained is 97.05%.

2.2 CNN

M. Rahman, S. Mahmud, J. Kim, Md. M. Manik, Md. M. Islam (2020) published a document aimed at developing a system for determining whether a person uses a mask or not and informing the relevant authority in the smart city network. It makes use of real-time filming of various public places of the city to capture the facial images. The facial images extracted from this video is being used to identify the masked faces. The convolutional neural network (CNN) learning algorithm is used to extract features from images, after which those features are learned through multiple hidden layers. Whenever the architecture identifies people without a mask, this information is passed through the city network to the appropriate authority in order to take the necessary actions. The proposed system assessed promising results based on data collected from various sources. In these documents, they also set out a system that can ensure proper law enforcement against people who do not follow basic health guidelines in this pandemic situation.

2.3 Viola Jones Face Detection Algorithm

Paul Viola and Michael Jones (2001) proposed the Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time. Even though it can be trained to detect a variety of object classes, it was motivated mainly by the problem of face detection. This face detection framework is capable of processing images extremely rapidly while achieving high detection rates. There are three keys of assistance.

The first is the introduction of a new image illustration called the Integral Image which allows the features used by our detector to be computed very quickly.

The second is an easy and efficient classifier which is built using the AdaBoost learning algorithm to select a small number of critical visual features from a very large set of potential features.

The third contribution is a process for combining classifiers in a cascade which allows background regions of the image to be quickly discarded while spending more computation on promising face-like regions.

It is the most admired algorithms for face detection in real time. The main advantage of this approach is uncompetitive detection speed while relatively high detection accuracy, comparable to much slower algorithms. High accuracy. Viola Jones gives accurate face detection. Constructing a cascade of classifiers which totally reduces computation time while improving detection accuracy. The Viola and Jones technique for face detection is an especially successful method as it has a very low false positive rate.

There are some disadvantages of the algorithm. Some of them are as follows:

- Extremely long training time.
- Limited head poses.
- Not detect black Faces.

2.4 YOLO

R. Bhuiyan, S. Khushbu, S. Islam (2020) have published one paper in which the proposed system aims for recognizing the masked and faces are rendered using the advanced YOLOv3 architecture. YOLO (You Only Look Once), uses the learning algorithm Convolution Neural Network (CNN). YOLO establishes a connection with CNN through hidden layers, through research, easy algorithm retrieval, and can detect and locate any type of image. Execution begins by taking 30 unique images of the dataset into the model after combining the results to derive action-level predictions. It gives excellent imaging results and also good detection results. This model is applied to a live video to check if the fps rate of the model inside the video and its detection performance with masked/unmasked two layers. Inside video, our model has impressive outputs with an average fps of This system is more efficient and faster than other methods using their own data set.

2.5 Deep Learning

Vinitha & Velantina (2020) published one article in which, using a deep learning algorithm and computer vision, they proposed a system that focuses on how to distinguish a person with a masked face in an image / video stream. Libraries like Tensor flow, Open CV, Keras and PyTorch are being used. The project is being implemented in two stages. The phase one consists of training a deep learning model followed by the second phase where mask detector is applied on live image/video stream. The framework used to do real-time face detection from a live stream via webcam is OpenCV.

According to Tech-Sci Research report “Global Facial Recognition Market By Component (Software & Services), By Software Type (2D, 3D & Thermal Face Recognition), By End Use Sector (Government & Transportation, Military & Defence, etc), By Region, Competition Forecast & Opportunities, 2023”, the global facial recognition market stood at around \$ 2.9 billion in 2017 and is expected to grow at a CAGR of 13% by 2023, on account of growing criminal activities and increasing need for enhanced monitoring and surveillance. Moreover, rising awareness about the benefits of facial recognition and growing implementation by the commercial sector is further expected to push the demand for facial recognition technology in the coming years. Government, transportation and military & defence are the largest contributors to the global facial recognition market as these end use sectors are adopting facial recognition for real time identification of criminals and to provide more secure environment. Additionally, huge National ID Programs like ‘Aadhaar’ in India are also boosting the face recognition market as they involve recording the facial biometrics of individuals.

2.6 Aim and Objectives

The main aim of this project is to provide or develop a system that will use the camera of the computer or the system that would detect and recognize the person’s face or the face of the individual using the tool in OpenCV called as the Open- Source Computer Vision and python programming language in deep learning domain.

Objectives

The objectives of Face Recognition for Real-Time Applications are given below:

- To enhance the Frame/sec for Face Recognition System, such that Recognition is done in Real Time.
- To achieve higher frames/sec or high-Resolution frames/sec.

CHAPTER 3

PROPOSED APPROACH AND SYSTEM ARCHITECTURE

3.1 Proposed Approach

In order to create this system first we will have to make the datasets. When the image quality becomes favorable different procedures will take place in the face recognition system the tasks are performed using the python queries “python camera_test.py”. The input will be taken from the webcam. Secondly a file detect_faces.py” will contain all the required methods and the techniques for the process of identification of the face of the person from the given image of the dataset. The given file will be executed by the python command “python recognize_faces_image.py”. We can resize or turn the image for approximately with the goal for getting the desired output. The present classifier along with OpenCV libraries will enhance the outcome or results in the face recognition system.

3.2 System Architecture

Our application is divided into three phases:

- Phase 1: Data Gathering
- Phase 2: Train the Recognizer
- Phase 3: Recognition

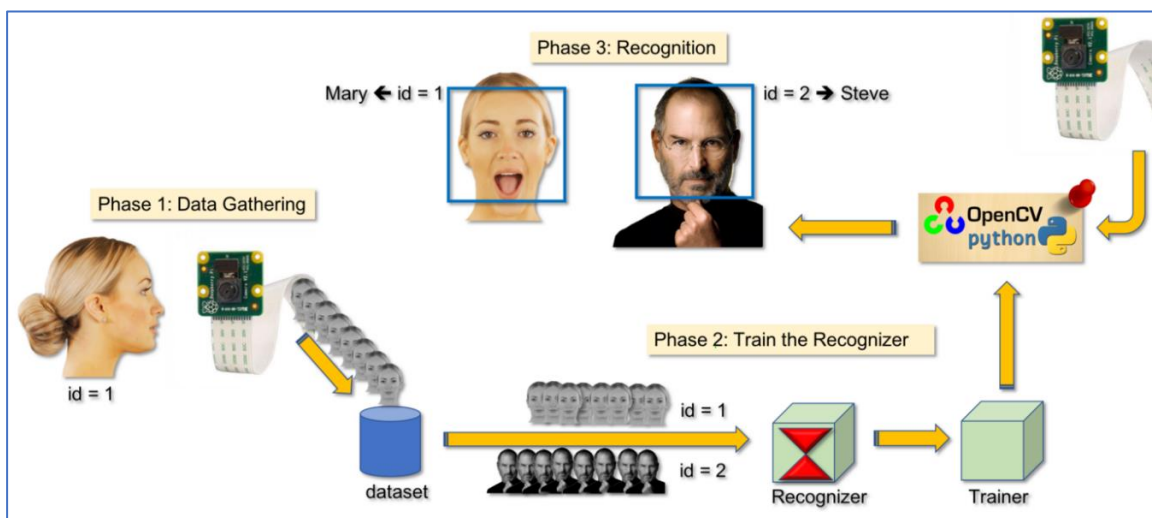


Fig. 3.2 System Architecture

3.2.1 Use Case Diagram

A use case is a description of how end-users will use a software code. It describes a task or a series of tasks that users will accomplish using the software and includes the responses of the software to user actions.

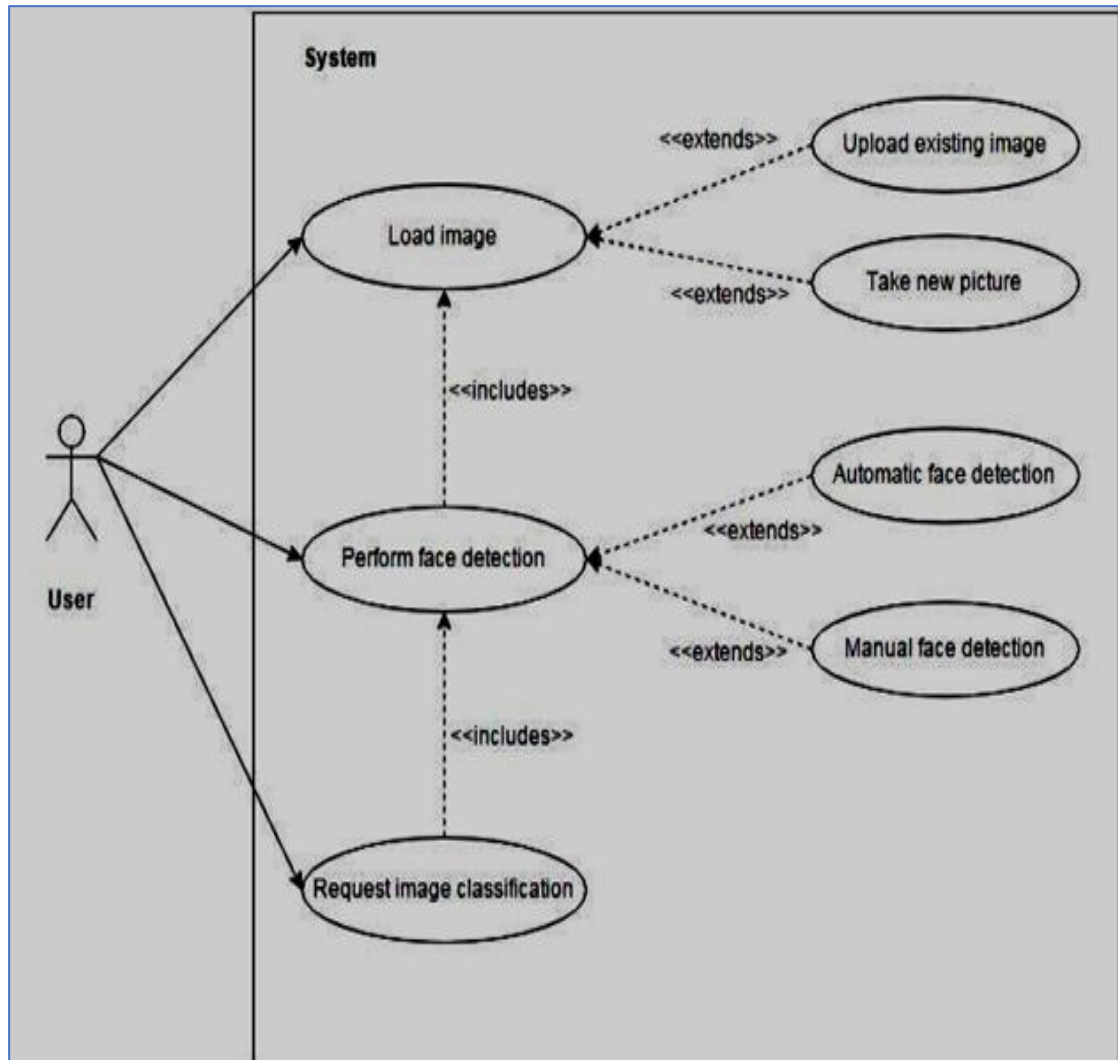


Fig: 3.2.1 Use Case Diagram for Face Recognition

3.2.2 Sequence Diagram:

The below diagram shows the sequence diagram for face recognition

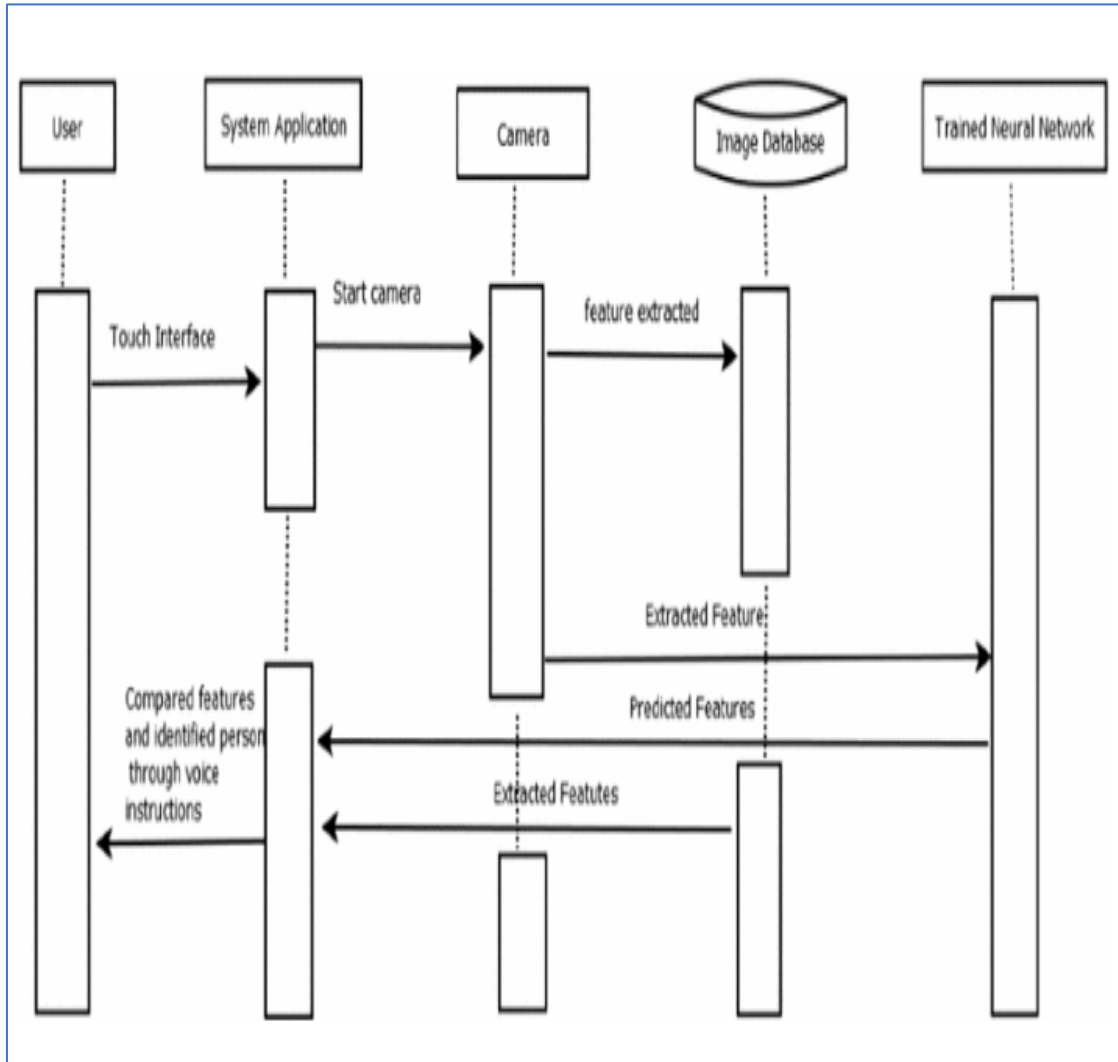


Fig: 3.2.2 Sequence Diagram for Face Recognition

3.3 SYSTEM REQUIREMENTS

This project requires the following hardware and software requirements

3.3.1 Hardware Requirements

In this section, we will describe the hardware requirement for construction and deploying the system, including personal computer, android smart phones having internet services.

Processor	Intel i3(PC)
RAM	1 GB minimum
Disc Space	4 GB minimum

Table 3.3.1: Hardware Requirements

3.3.2 Software Requirements

In this section, we will describe the software requirement for constructing and deploying this system.

Operating System	Windows 7/8/10/11
Frontend	Python
IDE	PyCharm (2021.3.3)

Table: 3.3.2 Software Requirements

CHAPTER 4

TOOLS AND TECHNOLOGIES

4.1 OPENCV

It is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including feature like face detection and object detection.

Currently Open CV supports a wide variety of programming languages like C++, Python, Java etc. and is available on different platforms including Windows, Linux, OS X, Android, iOS etc.

Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations. Open CV-Python is the Python API of Open CV. It combines the best qualities of Open CV C++ API and Python language.

OpenCV (Open-Source Computer Vision Library) is an opensource 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.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of -the-art computer vision and machine learning algorithms.

Algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

4.2 NUMPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications.

NumPy is opensource software and has many contributors. The Python programming language was not initially designed for numerical computing, but attracted the attention of the scientific and engineering community early on, so that a special interest group called matrix-sig was founded in 1995 with the aim of defining an array computing package. Among its members was Python designer and

maintainer Guido van Rossum, who implemented extensions to Python's syntax (in particular the indexing syntax) to make array computing easier. An implementation of a matrix package was completed by Jim Fulton, then generalized by Jim Hugunin to become Numeric also variously called Numerical Python extensions or NumPy. Hugunin, a graduate student at Massachusetts Institute of Technology (MIT) joined the Corporation for National Research Initiatives (CNRI) to work on J Python in 1997 leaving Paul Dubois of Lawrence Livermore National Laboratory (LLNL) to take over as maintainer. In early 2005, NumPy developer Travis Oliphant wanted to unify the community around a single array package and ported num-array's features to Numeric, releasing the result as NumPy 1.0 in 2006. This new project was part of SciPy. To avoid installing the large SciPy package just to get an array object, this new package was separated and called NumPy.

4.3 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCS), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also an educational version, as well as a Professional Edition with extra features.

Features of PyCharm:

1. Coding assistance and analysis, with code completion, syntax and error highlighting, linear integration, and quick fixes
2. Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
3. Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
4. Support for web frameworks: Django, web2py and Flask [professional edition only]
5. Integrated Python debugger
6. Integrated unit testing, with line-by-line code coverage
7. Google App Engine Python development [professional edition only]
8. Support for scientific tools like Matplotlib, NumPy and SciPy [professional edition only]

4.4 Notepad

Notepad is a basic text editor that you can use to create simple documents. It hasn't really changed much even though all other standard Windows tools were upgraded with each new version. The most common use for Notepad is to view or edit text(.txt) files, but many users find Notepad a simple tool for creating Web pages.

Windows Notepad is a simple text editor for Windows; it creates and edits plain text documents. First released in 1983 to commercialize the computer mouse in MS-DOS, Notepad has been part of every version of Windows ever since.

USES:

1. Create, open, and save text files with Notepad
2. Use Notepad to make simple text edits
3. Search and replace text in Notepad documents
4. Turn Word Wrap on or off
5. Zoom in and zoom out in Notepad (only in Windows 10)
6. Change the font of the text document
7. Print text files with Notepad

4.5 Python

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry.

Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

Below are some facts about Python Programming Language:

1. Python is currently the most widely used multi-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber etc

5. The biggest strength of Python is huge collection of standard Library which can be used for the following:

- Machine Learning
- GUI Applications (like Tkinter, etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing etc.

4.6 Windows Operating System

Microsoft Windows, commonly referred to as Windows, is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft. Each family caters to a certain sector of the computing industry. Active Microsoft Windows families include Windows NT and Windows IoT; these may encompass subfamilies, (e.g. Windows Server or Windows Embedded Compact) (Windows CE). Defunct Microsoft Windows families include Windows 9x, Windows Mobile and Windows Phone.

Microsoft introduced an operating environment named *Windows* on November 20, 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). Microsoft Windows came to dominate the world's personal computer (PC) market with over 90% market share, overtaking Mac OS, which had been introduced in 1984.

Early versions include:

1. Windows 1.0
2. Windows 2.0
3. Windows 3.0
4. Windows 9x
5. Windows 95
6. Windows 98

Latest versions include:

1. Windows XP
2. Windows 7
3. Windows 8
4. Windows 10
5. Windows 11

4.7 IDLE

IDLE (short for **I**ntegrated **D**evelopment and **L**earning **E**nvironment) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distribution. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl /Tk).

IDLE is intended to be a simple IDE and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

According to the included README, its main features are:

- Multi-window text editor with syntax highlighting, autocompletion, smart indent and other.
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility.

Python IDLE offers a full-fledged file editor, which gives us the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.

CHAPTER 5

IMPLEMENTATION

It represents the development stages of the proposed application. This chapter also describes the functionalities and testing methods for proposed applications.

5.1 Function to test camera

The below function captures the face and pops a dialog box and shows the user's face using webcam.

The video_capture captures the face using webcam.

Inside the while loop, video_capture.read() function reads the face.

imshow() function pops a dialog box and shows face frame by frame.

If we press 'q' key in our keyboard dialog box ends and the loop is broken.

```
import cv2

video_capture = cv2.VideoCapture(0)

while True:
    ret, frame = video_capture.read()

    cv2.imshow("video frame", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video_capture.release()
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Fig 5.1 Function to test camera

5.2 Function to detect faces

In the following function, the webcam detects the user's face and draws a rectangular shape around the face. We need to download *.xml* file for the code to run. The user sets the color and shape according to his choice and runs the code to show face is detected or not.

```
import numpy as np
import cv2

faceCascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)
cap.set(3, 640) # set Width
cap.set(4, 480) # set Height

while True:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.2,
        minNeighbors=5,
        minSize=(20, 20)
    )

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 255), 0)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

    cv2.imshow('video', img)
```

Fig 5.2 Function to detect faces

5.3 Function to recognize the face

This function is used to recognize the face. To run this code we need to download .yml file. The user sent the theme font and also user tells his name. Then webcam reads the face and tells his/her name with positive accuracy. If no face is recognized, the system shows unknown with negative accuracy.

```
import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'
faceCascade = cv2.CascadeClassifier(cascadePath)
font = cv2.FONT_HERSHEY_TRIPLEX
# indicate id counter
id = 0
# names related to id
names = ['None', 'Aneesh']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

# Define min window size to be recognized as a face
minW = 0.1 * cam.get(3)
minH = 0.1 * cam.get(4)
while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Fig 5.3 Function to recognize the face

CHAPTER 6

RESULT AND DISCUSSION

In this chapter, the various results of execution are shown. When the system runs the program, the following results are displayed.

6.1 Testing the camera

This function test the camera. When the user runs the program, the dialog box named video frame appears and shows the user face.

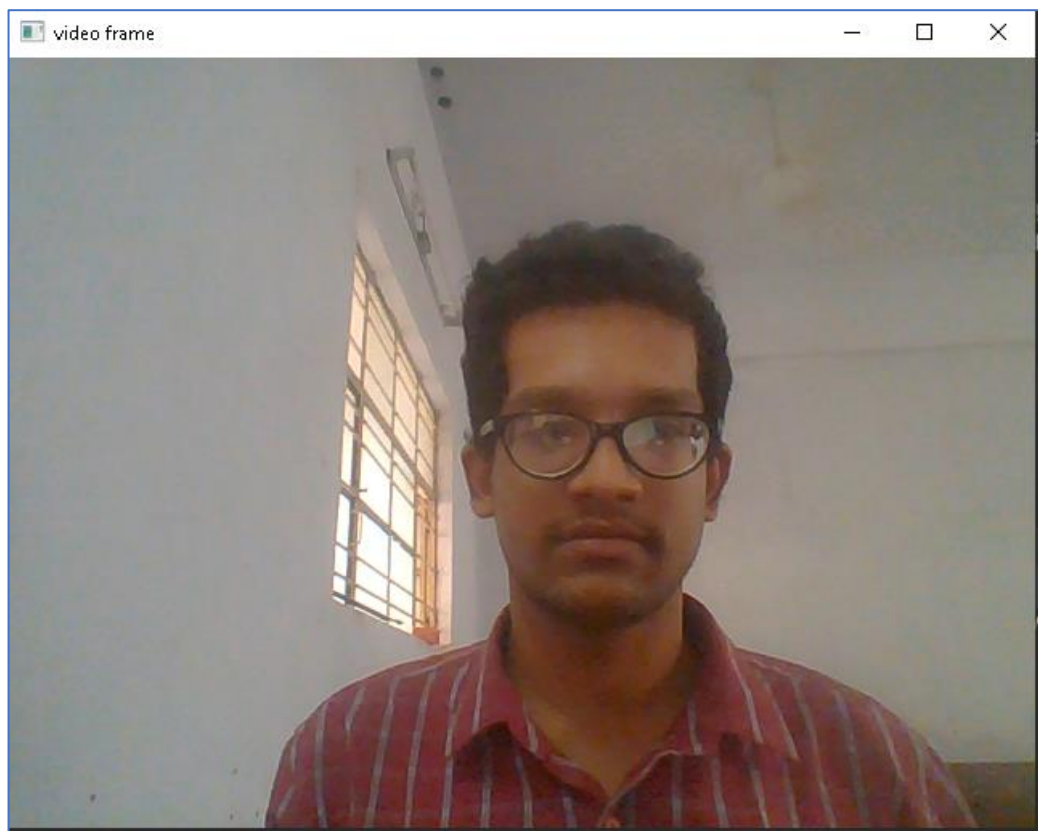


Fig 6.1 Testing the camera

6.2 Detect the face

This function is used to detect single or multiple faces. When the user runs the program, the dialog box appears named video and shows the users face surrounded by a red rectangular box around the face.

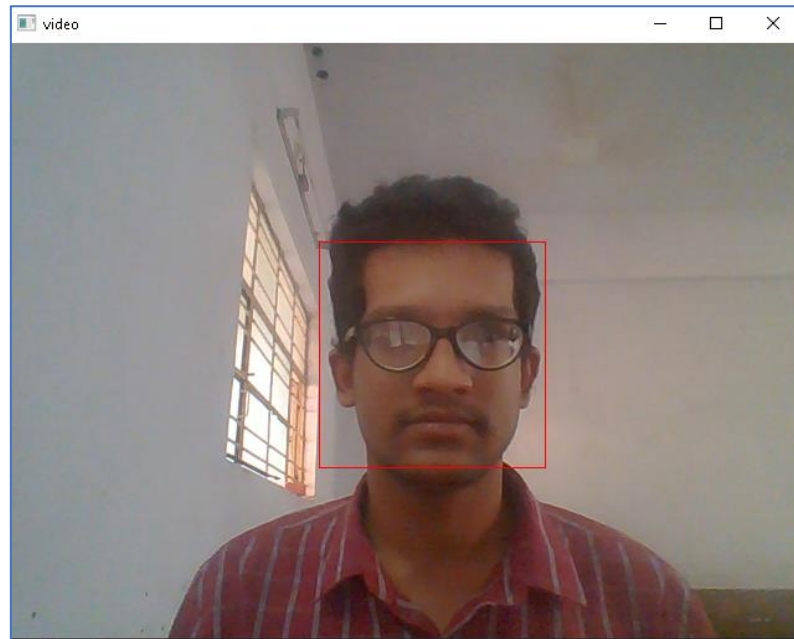


Fig.6.2 Detecting the face

6.3 Recognize the face

This function recognizes the face by showing the name and accuracy. When the user runs the program, the dialog box named camera appears and recognizes the face with the image stored in the database by telling name with positive accuracy.

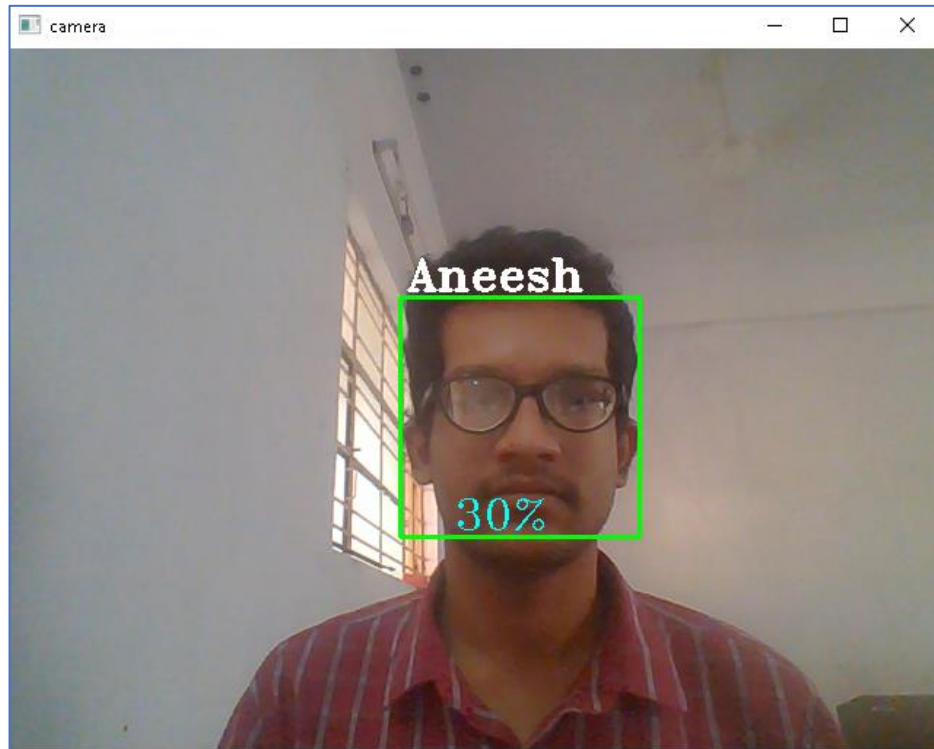


Fig 6.3 Recognize the face

6.4 Face is not recognized

In this function, we check the unknown image using our mobile. If the face is not recognized, it shown unknown with negative accuracy.

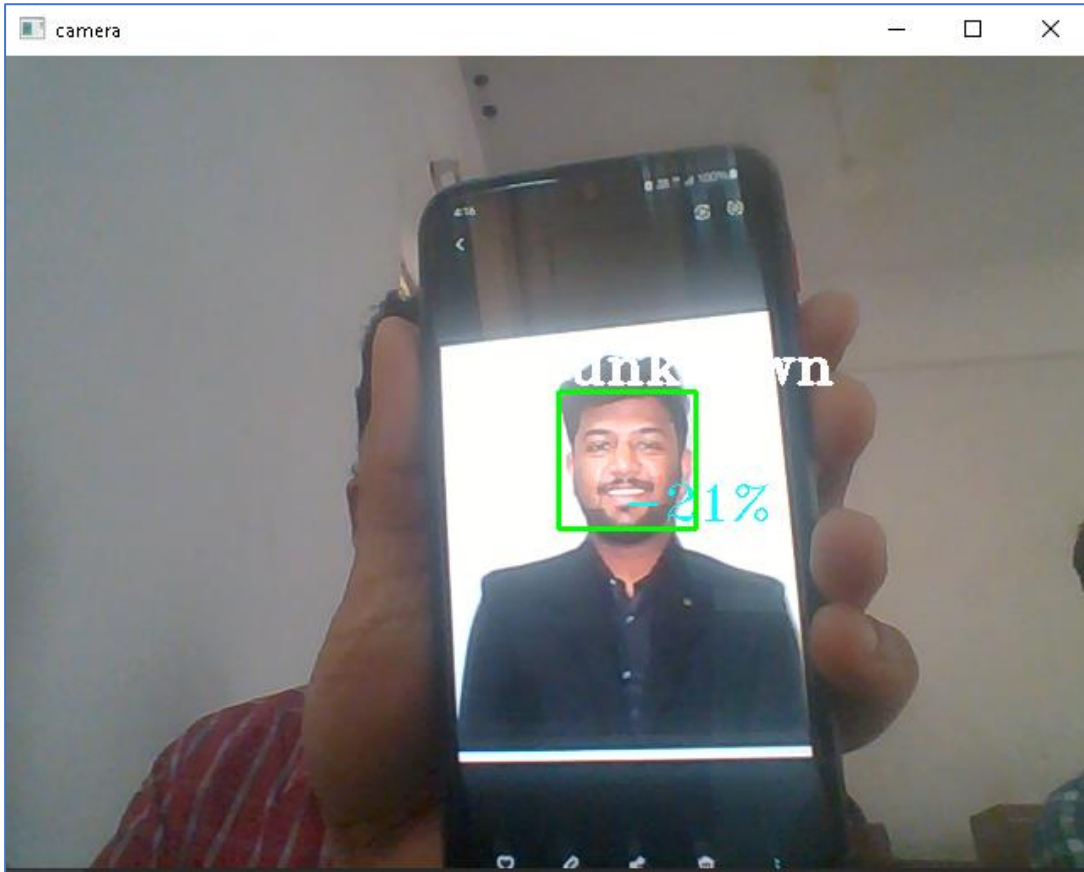


Fig. 6.4 If face not recognized

CHAPTER 7

CONCLUSION

In this work, a fast, efficient, secure and reliable facial recognition system is developed which replaces the manual, unreliable system. The system is time-saving and will reduce the work done by security agents in manually scanning surveillance videos. Furthermore, the need for specialized hardware for installing the system has been eliminated, as it uses a computer, a wireless IP camera and a wireless router. Since the system operates from live camera feed, the camera was tested for good image quality and performance in real-time. This ensured proper functioning of the system. The system can be deployed for permission-based scenarios and secure access authentication for access management, personal security, home (video) surveillance systems and crime control.

7.1 Limitations

- As with any technology, there are potential drawbacks to using facial recognition, such as
- threats to privacy,
- violations of rights and personal freedoms,
- potential data theft and other crimes.
- There's also the risk of errors due to flaws in the technology.

7.2 Future Scope

The future of facial recognition technology is bright. Forecasters says that this technology is expected to grow at a formidable rate and will generate huge revenues in the coming years. Security and surveillances are the major segments which will be deeply influenced. Other areas that are now welcoming it with open arms are private industries, public buildings, and schools. It is estimated that it will also be adopted by retailers and banking systems in coming years to keep fraud in debit/credit card purchases and payment especially the ones that are online. This technology would fill in the loopholes of largely prevalent inadequate password system. In the long run, robots using facial recognition technology may also come to foray. They can be helpful in completing the tasks that are impractical or difficult for human beings to complete.

References

1. HOG. N. J. Wang, S. C. Chang and P. J. Chou. (2012) “Face Detection in Real Time”, *International Symposium on Intelligent Signal Processing and Communications Systems*, pp. 333-337.
2. S.V.Viraktamath, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni (2013) “Face Detection and Tracking using OpenCV”, *The Standard International Journals (The SIJ)*, Vol. 1, pp. 45-50.
3. Face Recognition Homepage, <http://www.facerec.org/algorithms/>
4. Seeing with OpenCV, Article, http://www.cognotics.com/opencv/servo_2007_series/part_1/index.html
5. Python with deep learning, <https://www.tutorialspoint.c> Accessed on 4th April 2022 at 3:00 pm
6. <https://www.irjet.net/archives/V7/i10/IRJET-V7I10219.pdf> Accessed on 4th April 2022 at 3:00 pm
7. <https://www.youtube.com/watch?v=Id-jWnmqPoE> Accessed on 4th April 2022 at 3:00 pm
8. <https://www.geeksforgeeks.org/opencv-python-program-face-detection/> Accessed on 4th April 2022 at 3:00 pm