January 2022

ATTENDANCE WITH FACE_RECOGNITION

USING
MACHINE LEARNING

Prepared by: 20A91F0048

Machine learning - Project

Table of Content

Abstract

Face detection and face recognition are very important technologies these days, and attendance system of old practices are not quite efficient now a days for keeping a track. It is necessary to keep the effective system which records the attendance of person automatically. So decided to make a program that detects and recognize the face as a student attendance system and can be a substitute for the regular paper attendance system and finger print attendance system. This program is developed using python application that identifies and verifies person's identity with help of a camera. Once the recognized face matches with the stored image, the attendance is recorded by clicking "V" as verify option provided on the screen and the record will be stored in a excel sheet format of that person . Hence, the system is robust, cost-efficient and need less installation.
Components used in the project are
a computer installed with python , image data for training, excel sheet to save the record of attendance and a webcam
to scan the face id.

Introduction

The main objective of this project is to develop face recognition based automated student attendance system . In the old days, Facebook used to make you to tag your friends in photos by clicking on them and typing in their name. Now as soon as you upload a photo, Facebook tags everyone for you like magic This technology is called face recognition. In order to achieve better performance, the test and train images or frames are compared to gives the accuracy of 98% which is pretty much as good as humans can do and this is done by using machine learning

Problem Statement

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

Advantages

- **Automated Time Tracking System**

It is effortless to track time for employees with facial recognition.

- **Cost-Effective**

A facial recognition attendance system can save resources like paper , time tracking systems .

- **Touchless Sign In System**

Workplaces and multi-tenant environments can greatly reduce the frequency of contact between individuals

- **More Accurate and Better Worker Attendance**

With a face recognition attendance system, the entire environment is automated. You won't just take the attendance but also automatically record the entry-exit time

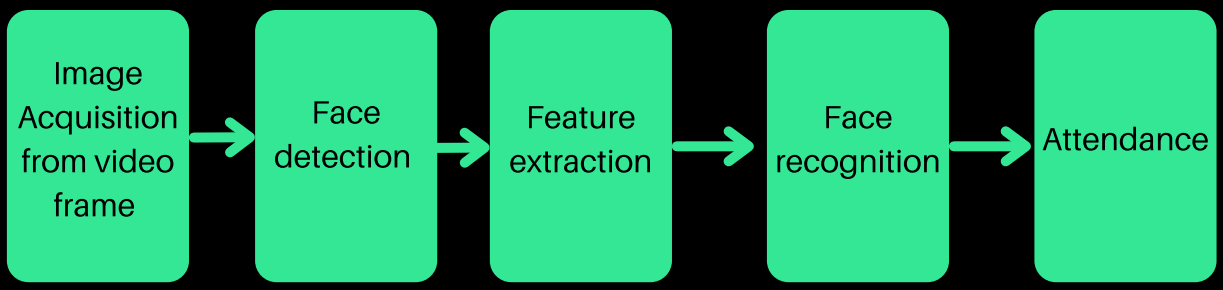
- **Easy To Manage**

As compared to manual attendance systems, AI-based attendance systems are highly automated.

Goals

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.



Face Detection

Difference between face detection and face recognition are often misunderstood. Face detection is to determine only the face segment or face region from image, whereas face recognition is to identify the owner of the facial image.

There are a few face detection methods that the previous researchers have worked on. However, most of them used frontal upright facial images which consist of only one face. The face region is fully exposed without obstacles and free from the spectacles. Akshara Jadhav et al. (2017) and by P. Arun Mozhi Devan et al. (2017) suggested Viola-Jones algorithm for face detection for student attendance system. They concluded that out of methods such as face geometry- based methods, Feature Invariant methods and Machine learning based methods, Viola-Jones algorithm is not 8 only fast and robust, but gives high detection rate and perform better in different lighting condition

S.Aanjanadevi et al. (2017) and Wei-Lun Chao (2007) presented a few factors which cause face detection and face recognition to encounter difficulties. These factors consist of background, illumination, pose, expression, occlusion, rotation, scaling and translation

Advantages & Disadvantages of Face Detection Methods

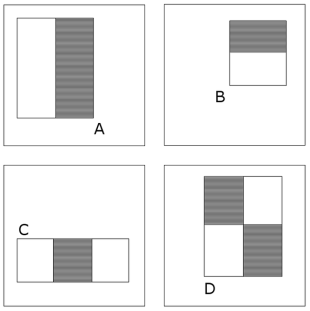
There are many face detection methods , mostly used algorithms are Viola jones algorithm , Local Binary pattern, AdaBoost algorithm (part of Viola jones algorithm) , SMQT Features and SNOW Classifier Method , Neural-Network

Face detection method	Advantages	Disadvantages
Viola jones algorithm	1. High detection speed 2. High accuracy	1. Long training time. 2. Limited head pose. 3. Not able to detect dark faces.
Local Binary pattern	1. Simple computation. 2. High tolerance against the monotonic illumination changes.	1. Only used for binary and grey images. 2. Overall performance is inaccurate compared to Viola-Jones algorithm.

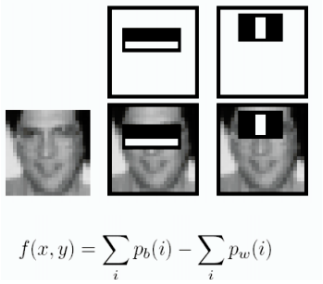
AdaBoost algorithm (part of Viola jones algorithm)	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	1. Capable to deal with lighting problem in object detection. 2. Efficient in computation	The region contain very similar to grey value regions will be misidentified as face
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm.

Viola-Jones Algorithm

Viola-Jones algorithm which was introduced by P. Viola, M. J. Jones (2001) is the most popular algorithm to localize the face segment from static images or video frame.
Basically the concept of Viola-Jones algorithm consists of four parts. The first part is known as Haar feature, second part is where integral image is created, followed by implementation of Adaboost on the third part and lastly cascading process.



Viola-Jones algorithm analyses a given image using Haar features consisting of multiple rectangles (Mekha Joseph et al., 2016). Figure 2.1 shows several types of Haar features. The features perform as window function mapping onto the image. A single value result, which representing each feature can be computed by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s) (Mekha Joseph

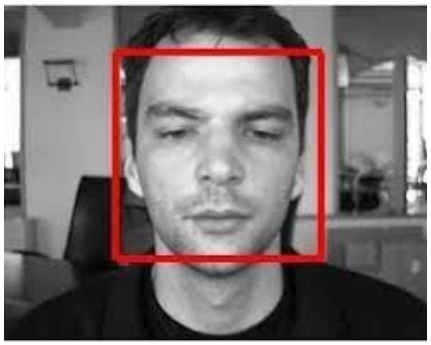


$$f(x,y) = \sum_i p_b(i) - \sum_i p_w(i)$$

The value of integrating image in a specific location is the sum of pixels on the left and the top of the respective location. In order to illustrate clearly, the value of the integral image at location 1 is the sum of the pixels in rectangle A. The values of integral image at the rest of the locations are cumulative. For instance, the value at location 2 is summation of A and B, (A + B), at location 3 is summation of A and C, (A + C), and at location 4 is summation of all the regions, (A + B + C + D) (Srushti Girhe et al., 2015). Therefore, the sum within the D region can be computed with only addition and subtraction of diagonal at location 4 + 1 − (2 + 3) to eliminate rectangles A, B and C.

Burak Ozen (2017) and Chris McCormick (2013), they have mentioned that Adaboost which is also known as 'Adaptive Boosting' is a famous boosting technique in which multiple "weak classifiers" are combined into a single "strong classifier". The training set is selected for each new classifier according to the results of the previous classifier and determines how much weight should be given to each classifier in order to make it significant.

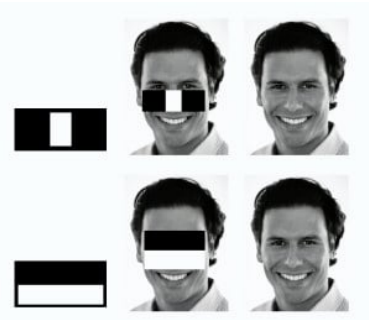
However, false detection may occur and it was required to remove manually based on human vision. Figure 2.3 shows an example of false face detection (circle with blue).



Pre-Processing

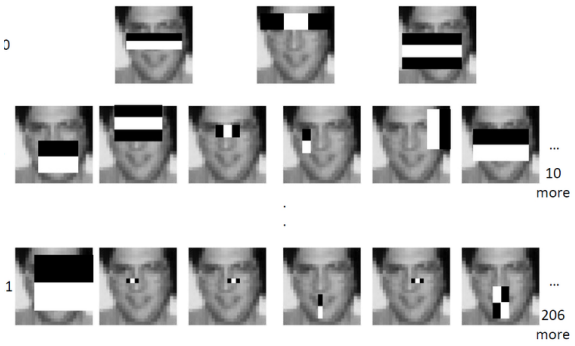
Subhi Singh et al. (2015) suggested cropping of detected face and colourimage was converted to grayscale for pre-processing. They also proposed affine transform to be applied to align the facial image based on coordinates in middle of the eyes and scaling of image to be performed. Arun Katara et al (2017), Akshara Jadhav et.al (2017),
Shireesha Chintalapati, and M.V. Raghunadh (2013), all of the 3 papers have proposed histogram equalization to be applied to facial image, and scaling of images was performed for pre-processing.

Pre-processing enhances the performance of the system. It plays an essential role to improve the accuracy of face recognition. Scaling is one of the important preprocessing steps to manipulate the size of the image. Scaling down of an image increases the processing speed by reducing the system computations since the number of pixels are reduced. The size and pixels of the image carry spatial information. Gonzalez, R. C. and Woods (2008) mentioned spatial information is a measure of the smallest discernible detail in an image. Hence, spatial information has to be manipulated carefully to avoid distortion of images to prevent checkerboard effect. The size should be same for all the images for normalization and standardization purposes. Subhi Singh et al (2015) proposed PCA (Principal Component Analysis) to extract features from facial images, same length and width of image is preferred, thus images were scaled to 120 × 120 pixels.



4 Images Show Checkerboard Effect Significantly Increasing from Left to Right

Besides scaling of images, colour image is usually converted to grayscale image for pre-processing. Grayscale images are believed to be less sensitive to illumination condition and take less computational time. Grayscale image is 8 bit image which the pixel range from 0 to 255 whereas colour image is 24 bit image which pixel can have 16 77 7216 values. Hence, colour image requires more storage space and more computational power compared to grayscale images. (Kanan and Cottrell, 2012). If colour image is not necessary in computation, then it is considered as noise. In addition, pre-processing is important to enhance the contrast of images. In the paper of Pratiksha M. Patel (2016), he mentioned that Histogram equalization is one of the methods of pre-processing in order to improve the contrast of the image. It provides uniform distribution of intensities over the intensity level axis, which is able to reduce uneven illumination effect at the same time.



Facial Images Were Converted To Grayscale, Histogram Equalization Was Applied and Images Were Resized to 100x100 (Shireesha Chintalapati and M.V. Raghunadh, 2013)

There are a few methods to improve the contrast of images other than Histogram Equalization. Neethu M. Sasi and V. K. Jayasree (2013) studied Histogram Equalization and Contrast Limited Adaptive Histogram Equalization (CLAHE) in order to enhance myocardial perfusion images. Aliaa A. A. Youssif (2006) studied contrast

enhancement together with illumination equalization methods to segment retinal vasculature. In addition, in paper by A., I. and E.Z., F. (2016) Image Contrast Enhancement Techniques and performance were studied. Unlike Histogram equalization, which operate on the data of the entire image, CLAHE operates on data of small regions throughout the image. Hence, the Contrast Limited Adaptive Histogram Equalization is believed to outperform the conventional Histogram Equalization.

Feature Extraction

The feature is a set of data that represents the information in an image. Extraction of facial feature is most essential for face recognition. However, selection of features could be an arduous task. Feature extraction algorithm has to be consistent and stable over a variety of changes in order to give high accuracy resul

There are a few feature extraction methods for face recognition. In the paper of Bhuvaneshwari et al. (2017), Abhishek Singh and Saurabh Kumar (2012) and Liton Chandra Paul and Abdulla Al Sumam (2012), they proposed PCA for the face recognition. D. Nithya (2015) also used PCA in face recognition based student attendance system. PCA is famous with its robust and high speed computation. Basically, PCA retains data variation and remove unnecessary existing correlations among the original features. PCA is basically a dimension reduction algorithm. It compresses each facial image which is represented by the matrix into single column vector. Furthermore, PCA removes average value from image to centralize the image data. The Principle Component of distribution of facial images is known as Eigen faces. Every single facial image from training set contributes to Eigen faces. As a result, Eigen face encodes best variation among known facial images. Training images and test images are then projected onto Eigen face space to obtain projected training images and projected test image respectively. Euclidean distance is computed by comparing the distance between projected training images and projected test image to perform the recognition. PCA feature extraction process includes all trained facial images. Hence, the extracted feature contains correlation between facial images in the training set and the result of recognition of PCA highly depends on training set image.

Face Recognition

Face Recognition is being able to uniquely identify and verify a person's face by comparing and analyzing a biometrics person's face. A face recognition system is an application that is used for identifying or verifying a person from a digital image.

History of Facial Recognition

Woody Bledsoe, Helen Chan Wolf, and Charles Bisson were the earliest pioneers of facial recognition. They began working to recognize the human face using a computer in 1964 and 1965. They marked various landmarks on the face such as eye centers, nose, mouth manually. They later used the computer to mathematically rotate to compensate for pose variation. The distances between the facial landmarks were computed automatically and compared with the image to match the identity. This was the dawn of facial recognition.

Sirovich and Kirby applied linear algebra to facial recognition and made it a viable biometric for business. They developed a system called "Eigenface" where less than one hundred values were required to code the facial image accurately. In 1991, the discovery of face detection within an image by Turk and Pentland led to the beginning of automated facial recognition. This paved the way for the advancement and development of facial recognition technology. FERET program was rolled out in the early 1990s by the DARPA and NIST for commercial facial recognition. They created a database for facial images, which included 2413 facial images that represented 856 people.

. In the early 2000s, to provide independent government evaluations of facial recognition system and its prototype technologies, FRVTs was designed. These evaluations provided the necessary information to deploy the facial recognition technology in the best way to the law enforcement agencies and government. Face Recognition Grand Challenge was launched in 2006 to evaluate the face recognition algorithms available. It used high-resolution images, 3D face scans, and iris images for the test. The test concluded that the new algorithm was 10 times more accurate than the algorithms of 2002 and more than 100 times more accurate than the algorithm of 1995. In recent years, Facebook has implemented facial recognition functionality to identify people featured in the user's daily updates. In 2017, Apple launched the iPhone-X, which was the first iPhone to implement facial recognition to unlock the phone.

Importance of Facial Recognition System

Applications using facial recognition systems are widespread. They are applied in security systems, authentication systems, verification systems, surveillance systems, etc. We are interacting with face recognition systems without even realizing it. Many Businesses are using facial recognition systems for authentication, verification, and security. There are diverse applications of this system. Countries such as United States, United Kingdom, and Australia are now installing facial recognition technologies in different public spaces such as airports, cafes, shopping areas, factory areas, and government buildings. A large retail company like Alibaba is working on the development of pay-by-face technology. Workspaces are using this technology to record the clock in and clock out tine of the employees. Law enforcement agencies are installing cameras with facial recognition systems to identify criminals and search for missing persons. As facial recognition technology and algorithms advance, we would see it being implemented more and more in our society

Challenges of Facial Recognition System

A facial recognition system can revolutionize how businesses and governments interact with people. However, if not used properly, there are potential pitfalls with this technology. Potential misuse of personal and sensitive information is very real. Businesses and Organizations need to make sure that there are proper checks and balances and proper security before implementing this technology. Every time this technology scans someone's face, the distinct biometrics of the person is stored in a database. Depending on who owns the database and security in place to protect the database, the information can be leaked, stolen, or misused without the consent of the person. Facial recognition systems are not perfect. Data collected by humans are used to train the algorithms. If there are a lack of data and a diverse array of data to train the algorithms, the system can misidentify the person. There have been many instances where the system incorrectly identified the gender or identity of people with darker skin tones. This happened because of a lack of data representing a diverse array of people.

With the advancement of new technology comes a new type of crime. Criminals could access the facial recognition data by hacking the database and track people's movement, location, and information without their consent. Criminals can cause significant damage with the aid of a facial recognition system. They can steal sensitive personal information or the identity of a person to commit a crime.

The application of facial recognition technology holds many promise. However, it needs to be handled carefully. Businesses that want to implement this technology need to implement the proper framework and facial data protection measures. If successfully managed to implement this technology, they can reap the benefits of this technology

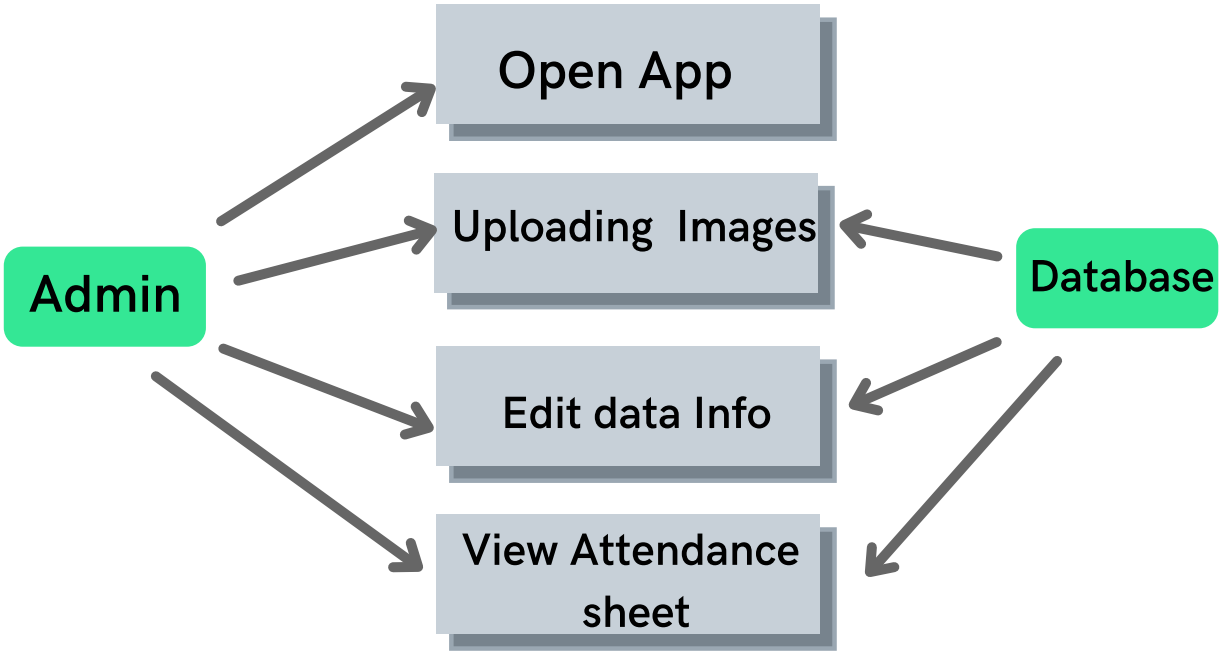
Requirement Analysis and Feasibility Study

Literature Review

Viola-Jones algorithm is used to detect the face. A camera is set up in the classroom that scans the facial structure of the students. The detected face is extracted for further processing. A image of that person in the database will be compared and return the best matched image in the data.

Admin Module

Admin has the highest privileges among all as admin is responsible to design the system. They are responsible to take images of the students and add them to the database. Admin can view and update the details of both admin and users. They can also view the attendance report.



Non-Functional Requirements

Non-Functional Requirements are the characteristics or attributes of the system that are necessary for the smooth operation of the system. Those requirements are listed below.

- The system should perform the process accurately and precisely to avoid problems
- The system should be easy to modify for any updates. Any errors or bugs that are identified should be easy to mend.
- The system should be secure and maintain the privacy of the students
- The system should be easy to understand and use.
- Execution of the operation should be fast.

Technical Feasibility

Technical feasibility is carried out to determine whether the project is feasible in terms of software, hardware, personnel, and expertise, to handle the completion of the project. It considers determining resources for the proposed system.

As the system is developed using python, it is platform independent. Therefore, the users of the system can have average processing capabilities, running on any platform. The technology is one of the latest hence the system is also technically feasible.

Method and Materials

This is the most important section of the thesis. This section describes the detailed workflow of the project and the necessary theoretical background. Tools and Technologies

Tools and Technologies

Tools and techniques used in the project are described in this section of the thesis. This project focused was mainly focused on Python Programming and its libraries.

Python

Python is a high-level object-oriented programming language. It was created by Guido van Rossum in 1991 as Python 0.9.0. It was created as the successor of the ABC programming language. Python 2.0 was released on 16 October 2000 and added many features like list comprehension and garbage collecting system. On 3 December 2008, Python 3.0 was released. Python is a very popular programming language and can be used for various purposes. It is widely used for web development, software development, mathematics and data analysis, system scripting, etc. Python is a multi-purpose programming language that works on different platforms like Windows, Linux, Mac, Raspberry Pie, etc. Python is popular than other programming languages because it has a simple syntax than other programming languages. Its syntax allows the programs to write code that is easier to understand and in fewer lines. It runs in an interpreter system. Hence, the code can be executed as soon as it is written

OpenCV

OpenCV is an open-source machine learning and computer vision library. OpenCV is a cross-platform library and is free to use. It was launched in 1999. Intel launched OpenCV to advance CPU-intensive applications. It was developed in C++. It provides bindings for Java and Python programming languages. It runs in different operating systems such as Linux, Windows, OSx, etc. It focuses mainly on video capturing, image processing, and analysis. It has face detection and objects detection features. OpenCV can be used to read and write images and capture and save videos. It can perform feature detection like faces, cars, images, etc. Many established companies like Yahoo, Google, Microsoft, Intel, and many others use the library.

Dlib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments.

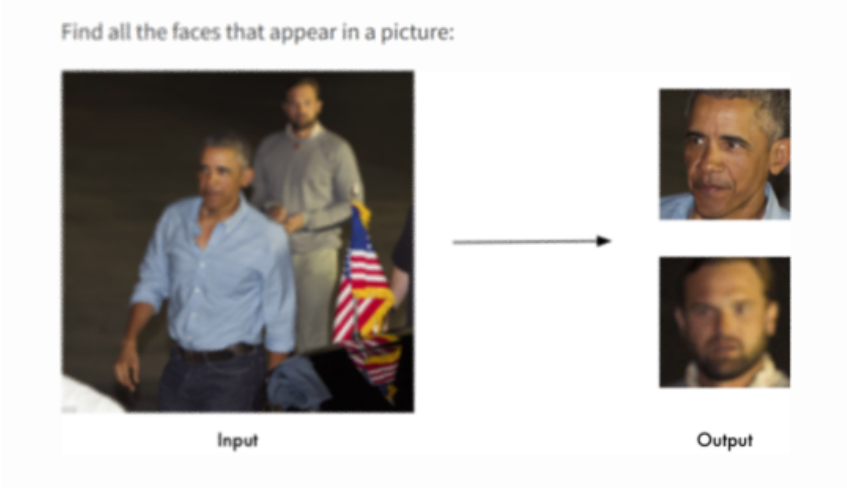
face_recognition

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38%

Approach

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

Find faces in pictures
Find all the faces that appear in a picture:



The approach follows the steps of loading of images as very first step , once the image is loaded it process the image by face_recogition library using face_recognition.face_loactions() method which reads the face. Also we can use face_recognition.face_landmark() to recognize the image this needs only one sinlge image to train the system and the accurate results were produced

Sign Up

It is nothing but adding a image into image data to recognize the person.

Importing of libraries program

OpenCV, numpy , face_recognition , OS

Loading of image data

Stored images of registered members

Converting to RGB2BGR /face detection/ Encoding

Three main steps to do

Comparing live image with images in image data

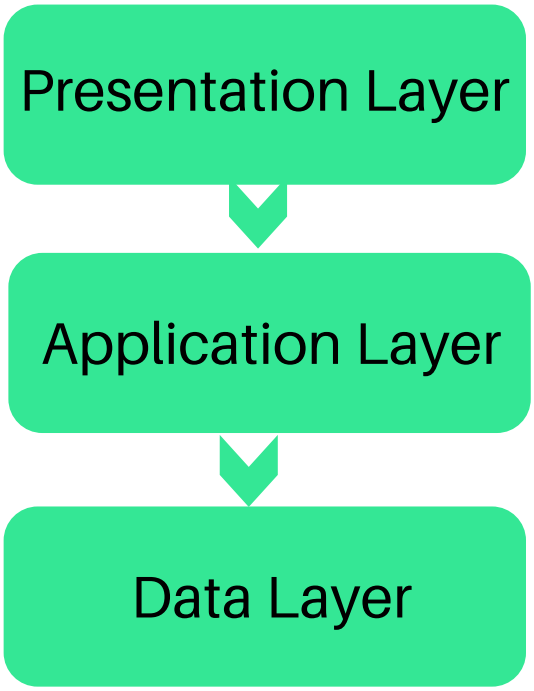
Repeating step four for live or snapshotted image and comparing their encodings .

Marking attendance and storing

If match found attendance will be recorded and stored.

System Design

The project follows three-layered architecture, which is described below
Presentation Layer



Presentation Layer

This layer is responsible for the user interface. All the components that users see and interact with within the application are in this layer.

Application Layer

Application layer controls the overall functionality of the system. Functionality such as logging into the system, facial detection, and recognition is all done in this layer.

Data Layer

In this layer, Data and Information are stored and retrieved in the database. The names, images of students as datasets, teaches are stored in the database. Once the face is matched, marking of attendance in the database

Application

Importing libraries

```
import cv2
import face_recognition
import os
import numpy as np
from datetime import datetime
```

Loading image data

```
def findEncoding(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```

Encoding the image

```python
def findEncoding(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```

Capturing live video

```python
while True:
    check, img = cap.read()
    imge = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imge = cv2.cvtColor(imge, cv2.COLOR_BGR2RGB)

    faceCur = face_recognition.face_locations(imge)
    encd = face_recognition.face_encodings(imge, faceCur)
```

Comparing encoding to recognize image

```
for encodeface, facloc in zip(encd, faceCur):
    matches = face_recognition.compare_faces(encodeListknown, encodeface)
    facedis = face_recognition.face_distance(encodeListknown, encodeface)
    matchIndex = np.argmin(facedis)

    if matches[matchIndex]:
        name = classnames[matchIndex].upper()
        print(name)
        y1, x2, y2, x1 = facloc
        y1, x2, y2, x1 = y1 * 4, x2 * 4, y1 * 4, x1 * 4
        cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0),
        cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
        cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX,1, (255, 255, 255), 2)
```

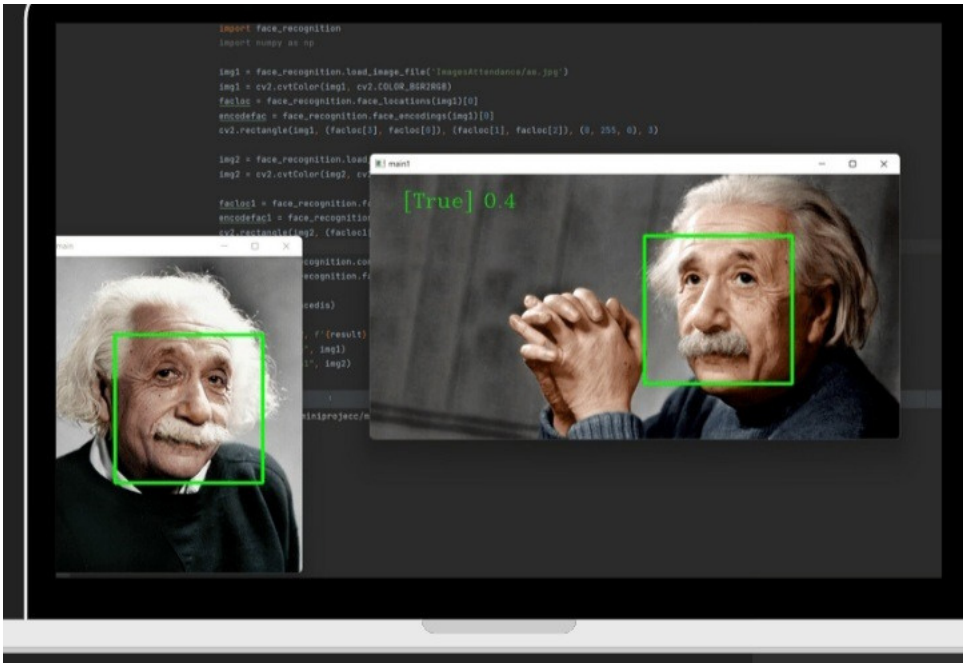
Recording attendance

```
def Attendance(name):
    with open('Attendance_Register.csv','r+') as f:
        DataList = f.readlines()
        names = []
        for data in DataList:
            ent = data.split(',')
            names.append(ent[0])
        if name not in names:
            curr = datetime.now()
            dt = curr.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dt}')
encodeKnown = DbEncodings(images)
```

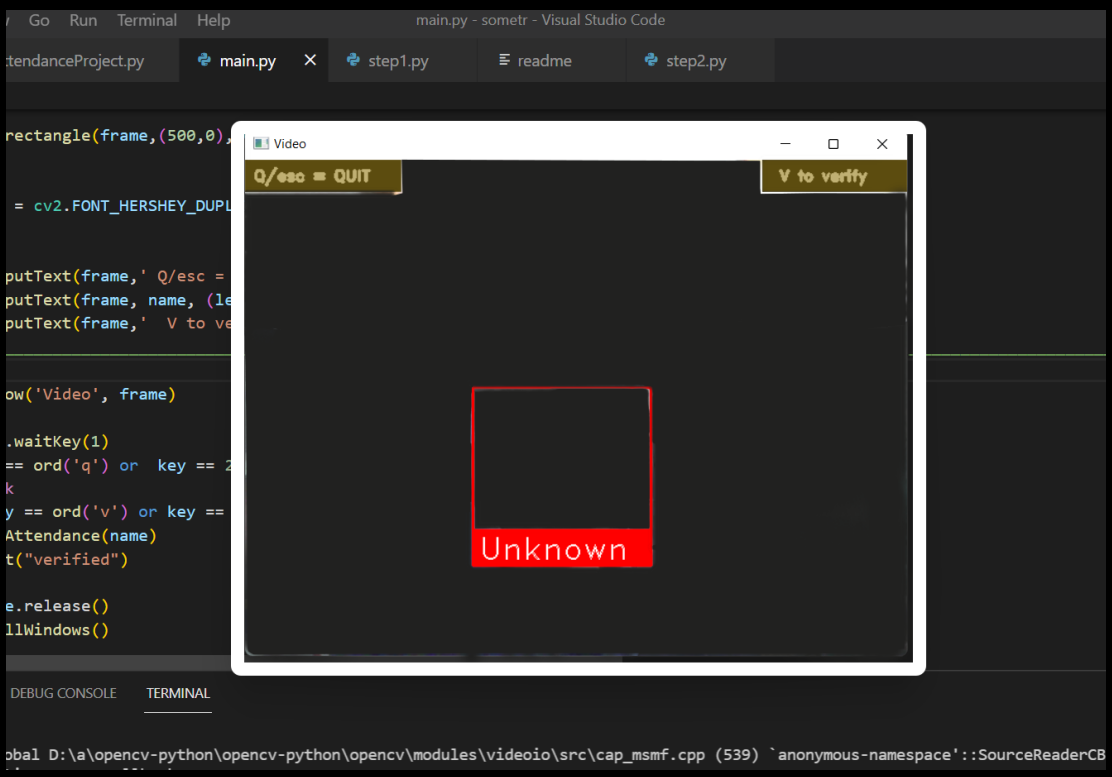
Sample program

```python
import cv2
import face_recognition
import numpy as np

img1 = face_recognition.load_image_file('ImagesAttendance/ae.jpg')
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
facloc = face_recognition.face_locations(img1)[0]
encodefac = face_recognition.face_encodings(img1)[0]
cv2.rectangle(img1, (facloc[3], facloc[0]), (facloc[1], facloc[2]), (0, 255, 0), 3)

img2 = face_recognition.load_image_file('ImagesAttendance/tst.png')
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

facloc1 = face_recognition.face_locations(img2)[0]
encodefac1 = face_recognition.face_encodings(img2)[0]
cv2.rectangle(img2, (facloc1[3], facloc1[0]), (facloc1[1], facloc1[2]), (0, 255, 0), 3)

result = face_recognition.compare_faces([encodefac], encodefac1)
facedis = face_recognition.face_distance([encodefac], encodefac1)

print(result, facedis)

cv2.putText(img2, f'{result} {round(facedis[0], 1)}', (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 1)
cv2.imshow('main', img1)
cv2.imshow('main1', img2)
```

Output



Output window for program



E n D