Face Recognition Using Python

B.Tech Project Report

by

Manish, Jatin

COMPUTER SCIENCE ENGINEERING DEPARTMENT

OM STERLING GLOBAL UNIVERSITY

HISAR, 125001 INDIA

May, 2023

FACE RECOGNITION USING PYTHON

A Project Report submitted

in partial fulfillment of the requirements

for the Degree of

Bachelor's in Technology

by

Manish, Jatin

(191SET01010001, 191SET01010006)

Under the guidance of

Ms. Neha

(APPROVED BY UGC, MINISTRY OF EDUCATION, GOVT. OF INDIA)

to the

COMPUTER SCIENCE ENGINEERING DEPARTMENT

OM STERLING GLOBAL UNIVERSITY

HISAR, 125001 INDIA

May, 2023

CERTIFICATE

This is to certify that the project titled **"FACE RECOGNITION USING PYTHON"** submitted by them in the partial fulfillment of the requirements for the degree of the Bachelor of Technology in (Computer Science & Engineering) during the academic year of 2019-2023, is an authentic work carried out by them under my guidance. The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this university or any other university/Institute for the fulfilment of the requirement of any course of study.

This is further certified that they have completed all the requirements of ordinance for submission of the project that has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

SUPERVISOR	**HEAD OF DEPARTMENT**
(Ms. Neha)	**(Dr. Rajinder Singh Sodhi)**

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report. We are privileged to express our sense of gratitude to our respected teacher **Ms. Neha** whose unparalleled knowledge, moral fiber and judgment along with his know-how, was an immense support in completing the project. We are also grateful to **Dr. Rajinder Singh Sodhi** the Head of Department, Information Technology, for the brainwave and encouragement given. We take this opportunity also to thank our friends and contemporaries for their co- operation and compliance.

Manish

Jatin

ABSTRACT

The face is one of the easiest ways to distinguish the individual identity of each other. 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 recognizes 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 patterns, namely, the 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 Eigenfaces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project's face detection system with face recognition is Image processing. The software requirements for this project is python software.

Keywords: face detection, Eigen face, PCA, python

Extension: There are vast number of applications from this face detection project, this project can be extended so that the various parts in the face can be detected which are in various directions and shapes.

INDEX

CONTENTS Page No.

Chapter-1
Introduction

Face recognition is the task of identifying an already detected object as a known or unknown face. Often the problem of face recognition is confused with the problem of face detection. Face Recognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

1.1 FACE RECOGNITION:

DIFFERENT APPROACHES OF FACE RECOGNITION:

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 the 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.

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.

1.2 FACE DETECTION:

Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin colour and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height). The face detection system can be divided into the following steps:-

a) **Pre-Processing:** To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples, that is the face images are obtained by cropping images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.

b) **Classification:** Neural networks are implemented to classify the images as faces or non-faces by training on these examples. We use implementation of the neural network toolbox for this task. Different network configurations are experimented with to optimize the results.

c) **Localization:** The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has done on:- Position Scale Orientation Illumination

1.3 What Is A Facial Recognition Attendance System?

A facial recognition attendance system uses facial recognition technology to identify and verify a person using the person's facial features and automatically mark attendance.

The software can be used for different groups of people such as employees, students, etc. The system records and stores the data in real-time.

1.4 What's the hype all about?

A facial recognition attendance system is a contactless way to manage visitors and employees in an organization.

Unlike other types of biometric systems, such as fingerprints that capture identity through touch, a facial recognition system is a touchless way to manage employees and visitors. In times of the COVID-19 pandemic, a touchless system is an effective preventive measure. It helps manage the inflow and outflow of people in buildings and premises in a safe and efficient manner.

1.5 A Touchless Attendance System - Ensure COVID-safe practices

The face identification software identifies a person's face and records attendance in the system in less than a second. No contact is involved.

Face recognition attendance system is a preventive measure that organizations can take to stop the corona virus from spreading. While ensuring efficiency and accuracy in time and attendance management.

If your organization has been using a fingerprint biometrics attendance system, it's time to switch over to a facial recognition biometric system for touchless attendance. If not, you certainly need to consider digitizing your daily office operations.

1.6 How Facial Recognition Works?

A facial recognition software captures and compares patterns on a person's face and analyses the details to identify and verify the individual. While the underlying system is complex, the whole technology can be broken down into three steps:

1. **Face Detection:** An essential step is locating human faces in real-time
2. **Transform Data:** Once captured, the analogue facial information is transformed into a set of data or vectors based on a person's facial features
3. **Face Match:** The system matches the data above with the one in the database for verification

Almost every big tech company including Amazon, Google, Microsoft, and Cisco is leading the effort to make face recognition more mainstream. There are many reasons to adopt the technology.

Chapter-2
Literature survey

Face detection is a computer technology that determines the location and size of a human face in an 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 parts in the given image i.e. feature base and image base approach. Feature based approach tries to extract features of the image and match it against the knowledge of the face features. The image base approach tries to get the best match between training and testing images.

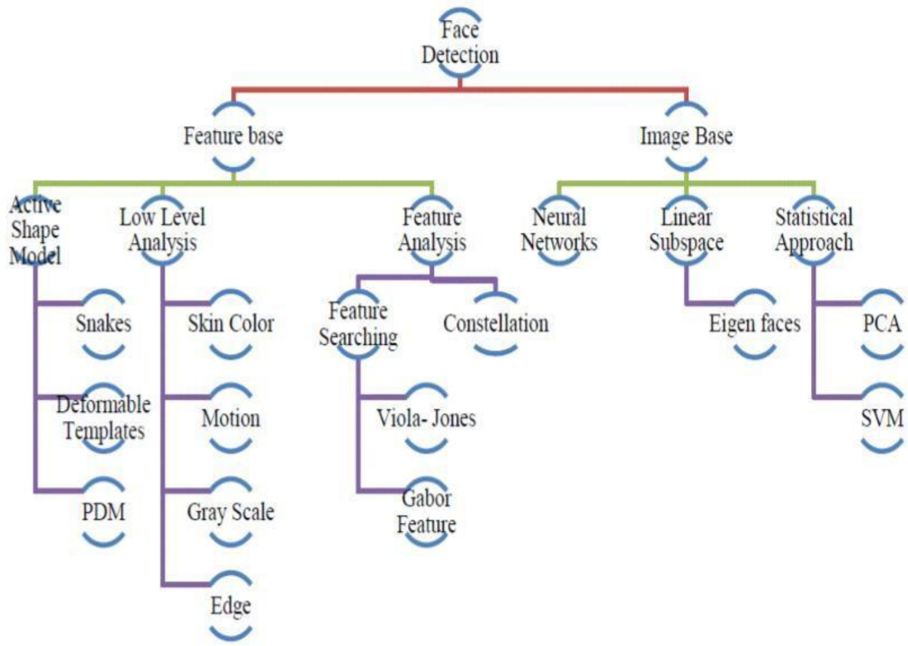


Fig 2.1 detection methods

2.1 FEATURE BASED APPROACH:

Active Shape Model Active 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 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

b. Snakes: The first type uses a generic active contour called snakes, first introduced by Kass et 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 assumes the shape of the head. The evolution of a snake is achieved by minimizing an energy function, Esnake (analogy with physical systems), denoted as snake = Internal + External Where Internal and External 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 equilibrium. Two main considerations for forming snakes i.e. selection of energy terms and energy minimization. Elastic energy is used commonly as internal energy. Internal energy varies 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 the other side external energy relies on image features. Energy minimization process is done by optimization techniques such as the steepest gradient descent. Which needs the 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.

2.2 LOW LEVEL ANALYSIS:

Based on low level visual features like color, intensity, edges, motion etc. Skin Color BaseColor is a vital 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.

(1) Classify the skin region in the color space,
(2) Apply threshold to mask the skin region and
(3) Draw a 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 regions like legs, arms, etc. Cahi and Ngan [27] suggested a skin color classification algorithm with YCbCr color space. Research found that pixels belonging to skin regions have similar Cb and Cr values. So that the thresholds are 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 predicate in HSV color space to separate skin regions from background. Skin color classification in HSI color space is the same as YCbCr color space but here the responsible values are hue (H) and saturation (S). Similar to above the threshold be chosen as [H1, S1] and [H2, S2], and a pixel is classified to have skin tone if the values [H,S] fall

within the threshold and this distribution gives the localized face image. Similar to the above two algorithms, this algorithm is also having the same constraint.

2.3 MOTION BASE:

When use of video sequence is available, motion in formation can be used to locate moving objects. Moving silhouettes like face and body parts can be extracted by simply thresholding accumulated frame differences . Besides face regions, facial features can be located by frame differences .

2.4 FEATURE ANALYSIS :

These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. These methods are designed mainly for face localization

2.4.1 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 over- complete 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.

2.5 CONSTELLATION METHOD

All methods discussed so far are able to track faces but still some issues like locating faces of various poses in complex backgrounds is truly difficult. To reduce this difficulty investigators 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.

2.6 LINEAR SUBSPACE METHOD

Eigenfaces 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 by HOTELLING 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. A reasonable likeness (i.e., capturing 95 percent of the variance)

2.7 OBJECTIVE:

Whenever we implement a new system it is developed to remove the shortcomings of the existing system. The computerized mechanism has more edge than the manual system. The existing system is based on a 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.

2.8. 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 operations 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 the 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 systems there are many problems with storing 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 user.

Chapter-3
Design flow/Process

3.1 Feature/characteristics identification

Planning is the strong key to make the project more effective and well utilization of resources to achieve the goal .Projects are not homogeneous. Each project is different in itself. The distinctive characteristics of a project are as follows:

a) **Single entity:** A good team is the first requirement for this project so that the system should have good user experience. We worked together as a team to bring this project to fruition as a single entity. We have a team of three members mentioned above. All of us worked hard on this project, still working to make our project user-friendly and attractive as much as we can.

b) **Life Span:** This project has a life span of 4 months from February to May. The project is divided in two steps. First we will develop the user interface using different modules and libraries of python. In the next step the project is tested and reviewed and hosting is to be done.

c) **Life Cycle:** Since this is a Face Recognition system that combines facial expression technologies to come up with a real-time intelligent system we have seen the need to add collaborative methodology to it because it isn't enough for the face recognition to work alone in a collaborative system. This system is developed in a way that it will be able to detect the person's facial expression.
To create a complete project on Face Recognition, we work on 3 very distinct phases:
    - Face Detection and Data Gathering
    - Train the Recognizer
    - Face Recognition

d) **Team Spirit:** We as a team had team spirit from the beginning, from project selection to learning to implementation. We put our faith in each other to complete the task at hand, we aided each other as needed. When working on a project as part of a group, team spirit, trust and passion are all important factors to consider.

e) **Risk and Uncertainty:** Our project is well defined, but still there remains a chance of error or uncertainty. Like there may be errors in the face detection and can lead to wrong results. There may be some malfunctioning in the equipment used resulting in error in results. There may be something's which lead to errors in the result. But still, even after hard work and hours of thinking, errors may remain.

f) **Flexibility:** We believe in "everything is constant except change" philosophy, our ideas are constantly evolving to make the best ever product. We are not rigid regarding our ideas and think the only way to move forward is through making changes.

3.2 Constraints Identification

a. Cost :

What is the cost of developing the face recognition software from scratch? And the answer is that it depends on the project requirements. There are many variables that comprise the total price of the bespoke face recognition software developed.

To offer the client the best solution to meet challenges, we enters such projects with the Investigation Stage aimed at the following:

a. Business objectives analysis
b. Data analysis and identification of the scope of work
c. Definition of high-level architecture and appropriate tools and technologies selection
d. Risks identification
e. Building proof of concept
f. Definition of the expected results as well as costs and resources estimation.

The investigation stage allows validating an idea at an early stage, minimizing risks of a project failure, and saving a substantial amount of money.

b. Risk :

Facial recognition technology also carries other risks, such as errors, which could, for example, lead to users being charged incorrectly. And as with any AI system, we should be concerned about whether the algorithms and data sets are free from bias, and have clean, complete and representative training data.

Importantly, employing facial recognition technologies also goes some way to normalizing the surveillance. It’s possible the knowledge they are being tracked in this way could impact someone's wellbeing.

c. Benefit :

Facial recognition is a technology that can benefit society, including increasing safety and security, preventing crimes, and reducing human interaction. Here are some pros of facial recognition:

a. Helps find missing people
b. Protects businesses against theft
c. Improves medical treatment
d. Strengthens security measures
e. Makes shopping more efficient
f. Reduces the number of touch-points
g. Improves photo organization

d. Quality :

In ideal conditions, facial recognition systems can have near-perfect accuracy. Verification algorithms used to match subjects to clear reference images (like a passport photo or mugshot) can achieve accuracy scores as high as 99.97% on standard assessments like NIST’s Facial Recognition Vendor Test (FRVT). This is comparable to the best results of iris scanners. This kind of face verification has become so reliable that even banks feel comfortable relying on it to log users into their accounts.

e. Scope :

a. In order to prevent the frauds of ATM in India, it is recommended to prepare the database of all ATM customers with the banks in India & deployment of high resolution camera and face recognition software at all ATMs. So, whenever a user enters an ATM his photograph will be taken to permit the access after it is being matched with stored photo from the database.

b. Duplicate voters are being reported in India. To prevent this, a database of all voters, of course, of all constituencies, is recommended to be prepared. Then at the time of voting the resolution camera and face recognition equipment of the voting site will accept a subject face 100% and generate the recognition for voting if a match is found.

c. Passport and visa verification can also be done using face recognition technology as explained above.

d. Driving license verification can also be exercised through face recognition technology as mentioned earlier.

e. To identify and verify terrorists at airports, railway stations and malls the face recognition technology will be the best choice in India as compared with other biometric technologies since other technologies cannot be helpful in crowded places.

f. Time :

The project's completion, or final due date for deliverables is May 2022. The project includes 3 phases: first phase i.e. requirement gathering includes the collection of data that can be done , second phase i.e. planning phase. Writing the main code, implementation and removal of bugs.

3.3 Analysis of features and finalization subject to constraints

The project required a great deal of reasoning on how and what highlights ought to be carried out. Keeping every one of the essential important prerequisites a venture ought to have as well as drawing out a few less executed ideas, we have summed up certain highlights the application contains as follows:

a. **CNN :** The convolutional neural network is a specialized kind of neural network model designed for working with two-dimensional image data.

b. **Open CV :** Open CV stands for open-source computer vision. It's a group of libraries in Python. it's a tool by which we will be able to manipulate the pictures, like image scaling, etc. Installation of all the necessary libraries.

c. **Tensorflow :** Tensor flow is a high-performance numerical computation open source software package. Its adaptable design allows for easy computing deployment across a variety of platforms (CPUs, GPUs, TPUs), from PCs to clusters of servers to mobile and edge devices.

d. **Training the dataset :** The data set is typically trained with the gathering of knowledge and available information, collection of images and files. Several different images of each face are used to train the data set. The thing to be kept in mind is that the images should be properly labeled for clear understanding of the data set. The training is done in three phases: training, validation and testing to determine the accuracy and performance of the system. More are the number of images provided to the data set, more is the accuracy.

e. **Debugging :** The subsequent errors that arise are to be dealt with if needed.

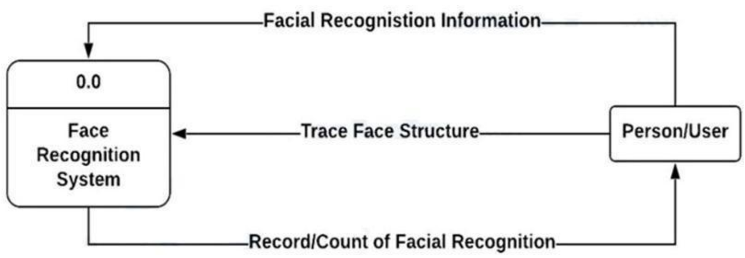
Feature based face recognition algorithms are computationally efficient compared to model based approaches. These algorithms have proved themselves for face identification under variations in poses. However, the literature lacks direct and detailed investigation of these algorithms in completely equal working conditions. This motivates us to carry out an independent performance analysis of well known feature based face identification algorithms for different poses with mug-shot face database situations. The analysis focuses on variations in performance of feature based algorithms in terms of identification rates due to variation in poses. The analysis is carried out in a face identification scenario using large amounts of images from the standard face databases such as AT&T, Georgian Face database and Head Pose Image database. We analyzed state-of-the art feature based algorithms such as PCA, log Gabor, DCT and FPLBP and found that log Gabor outperforms for larger degree of pose variation with an average identification rate 82.47% with three training images for Head Pose Image database.

3.4 Design selection

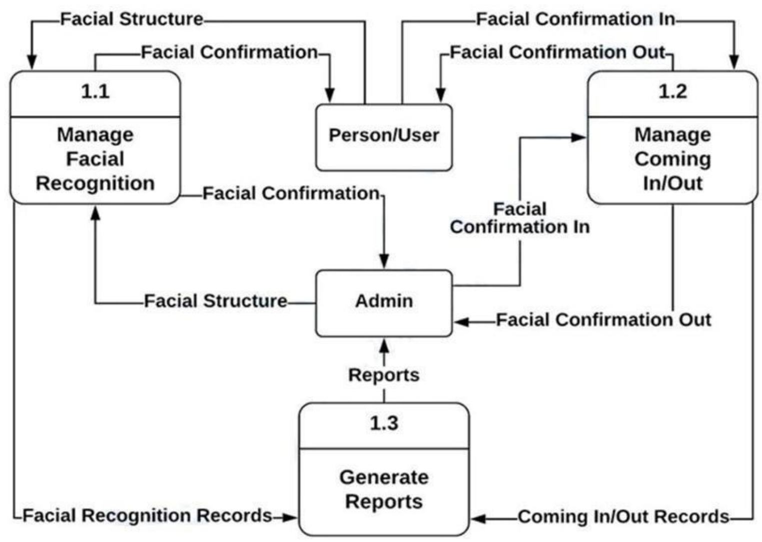
The following are the data included in Face Recognition System DFD:

- Facial Records
- Records of Unfamiliar Faces in Real-Time
- Records of Familiar Faces in Real-Time
- Number of Real-Time Records
- Count of Individuals that comes in and out of the Establishment

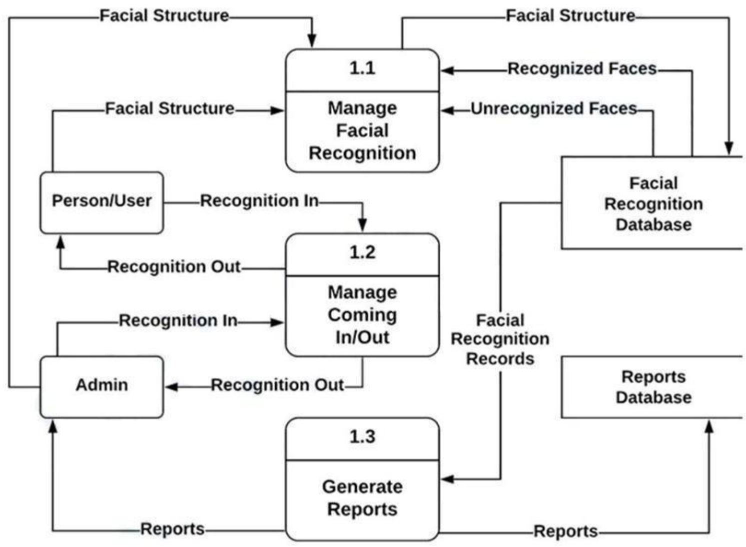
DFD
Level-0



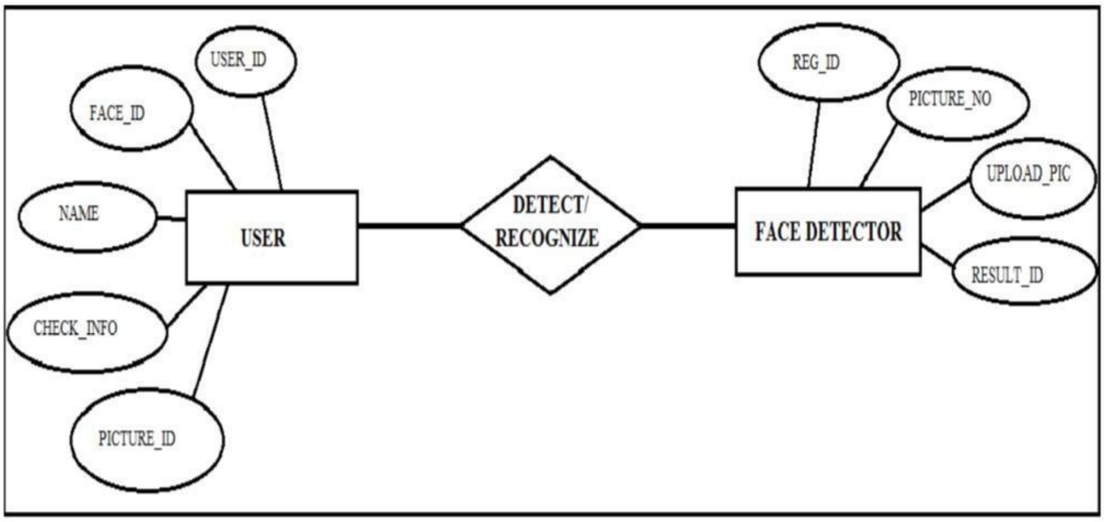
DFD Level-1



DFD Level-2

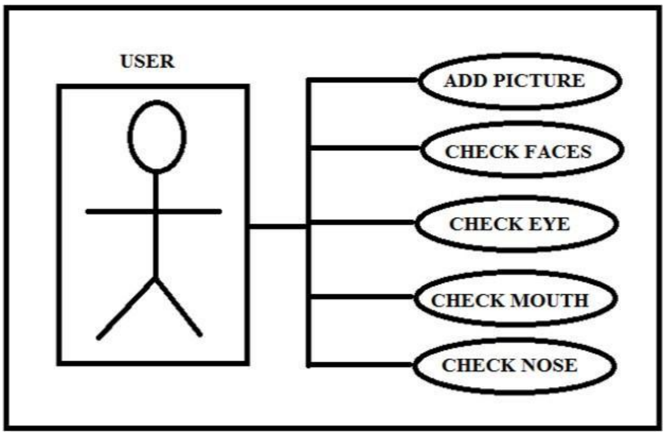


Entity Relationship Diagram:

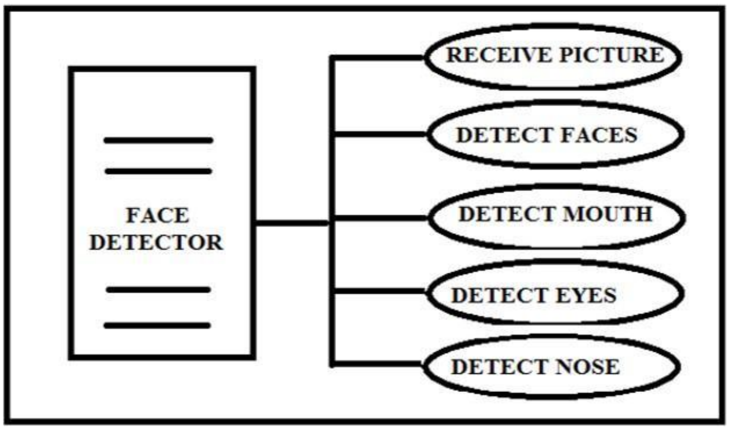


Use Case Diagram:

- **User Module:**



- **Software Module:**



Chapter-4
Results analysis and validation

This report has considered technical merits of Face Recognition System, particularly as they function in real-world settings in relation to specific goals. Although certain barriers to performance might be overcome by technical breakthroughs or mitigated by policies and guidelines, there remains a class of issues deserving attention not centered on functional efficiency but on moral and political concerns. These concerns may be grouped under general headings of privacy, fairness, freedom and autonomy, and security.

While some of these are characteristically connected to facial recognition and other biometric and surveillance systems, generally, others are exacerbated, or mitigated, by details of the context, installation, and deployment policies.

Privacy

Privacy is one of the most prominent concerns raised by critics of FRS. This is not surprising because, at root, FRS disrupts the flow of information by connecting facial images with identity, in turn connecting this with whatever other information is held in a system's database. Although this need not in itself be morally problematic, it is important to ascertain, for any given installation, whether these new connections constitute morally unacceptable disruptions of entrenched flows (often regarded as violations of privacy) or whether they can be justified by the needs of the surrounding context.

Fairness

The question of fairness is whether the risks of FRS are borne disproportionately by, or the benefits flow disproportionately to, any individual subjects, or groups of subjects. For example, in the evaluations discussed above, noting that certain systems achieve systematically higher recognition rates for certain groups over others—older people over youth and Asians, African-Americans, and other racial minorities over whites—raises the politically charged suggestion that such systems do not belong in societies with aspirations of egalitarianism. If, as a result of performance biases, historically affected racial groups are subjected to disproportionate scrutiny, particularly if thresholds are set so as to generate high rates of false positives, We are confronted with racial bias similar to problematic practices such as racial profiling.

Freedom and Autonomy

In asking how facial recognition technology affects freedom and autonomy, the concern is constraints it may impose on people's capacity to act and make decisions ("agency"), as well as to determine their actions and decisions according to their own values and beliefs. It is important to stress that the question is posed against a backdrop of existing expectations and standards of freedom and autonomy, which recognize that freedom and autonomy of any person is legitimately circumscribed by the rights of others, including their freedom, autonomy, and security.

Security

Acceptance of facial recognition and other biometric identification systems has generally been driven by security concerns and the belief that these technologies offer solutions. Yet, less salient are the security threats posed by these very systems, particularly threats of harm posed by lax practices dealing with system databases. Recent incidents in the UK and US suggest that institutions still do not deserve full public trust in how they safeguard personal information. In the case of biometric data, this fear is magnified many times over since it is generally assumed to be a non-falsifiable anchor of identity. If the biometric template of my face or fingerprint is used to gain access to a location, it will be difficult for me to argue that it was not me, given general, if problematic, faith in the claim that "the body never lies." Once my face or fingerprint has been digitally encoded, however, it can potentially be used to act "as if " it were me and, thus, the security of biometric data is a pressing matter, usefully considered on a par with DNA data and evidence. A similar level of caution and security needs to be established.

Chapter-5
Conclusion and future work

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 that were used for the PCA transform. This system was tested under very robust conditions in this experimental study and it is envisaged that real-world performance will be far more accurate. The fully automated frontal view face detection system displayed virtually perfect accuracy and in the researcher's opinion further work need not be conducted in this area.

The fully automated face detection and recognition system was not robust enough to achieve a high recognition accuracy. The only reason for this was the face recognition subsystem did not display even a slight degree of invariance to scale, rotation or shift errors of the segmented face image. However, if some sort of further processing, such as an eye detection technique, was implemented to further normalize the segmented face image, performance will increase to levels comparable to the manual face detection and recognition system. Implementing an eye detection technique would be a minor extension to the implemented system and would not require a great deal of additional research. All other implemented systems displayed commendable results and reflect well on the deformable template and Principal Component Analysis strategies. The most suitable real-world applications for face detection and recognition systems are for mugshot matching and surveillance. There are better techniques such as iris or retina recognition and face recognition using the thermal spectrum for user access and user verification applications since these need a very high degree of accuracy. The real-time automated pose invariant face detection and recognition system proposed in chapter seven would be ideal for crowd surveillance applications. If such a system were widely implemented its potential for locating and tracking suspects for law enforcement agencies is immense.

The implemented fully automated face detection and recognition system (with an eye detection system) could be used for simple surveillance applications such as ATM user security, while the implemented manual face detection and automated recognition system is ideal for mugshot matching. Since controlled conditions are present when mugshots are gathered, the frontal view face recognition scheme should display a recognition accuracy far better than the results, which were obtained in this study, which was conducted under adverse conditions.

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.

REFERENCE

- 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

- Beymer, D. and Poggio, T. (1995) Face Recognition From One Example View, A.I. Memo No. 1536, C.B.C.L. Paper No. 121. MIT

- Bichsel, M. (1991). Strategies of Robust Objects Recognition for Automatic Identification of Human Faces. PhD thesis, , Eidgenössische Technische Hochschule, Zurich.

- Brennan, S. E. (1982) The caricature generator. M.S. Thesis. MIT.

- Brunelli, R. and Poggio, T. (1993), Face Recognition: Features versus Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10):1042-1052

- Craw, I., Ellis, H., and Lishman, J.R. (1987). Automatic extraction of face features. Pattern Recognition Letters, 5:183-187, February.

- Deffenbacher K.A., Johanson J., and O'Toole A.J. (1998) Facial ageing, attractiveness, and distinctiveness. Perception. 27(10):1233-1243

- Dunteman, G.H. (1989) Principal Component Analysis. Sage Publications.

- Frank, H. and Althoen, S. (1994). Statistics: Concepts and applications. Cambridge University Press. p.110

- Gauthier, I., Behrmann, M. and Tarr, M. (1999). Can face recognition really be dissociated from object recognition? Journal of Cognitive Neuroscience, in press.

- Goldstein, A.J., Harmon, L.D., and Lesk, A.B. (1971). Identification of human faces. In Proc. IEEE, Vol. 59, page 748

- de Haan, M., Johnson, M.H. and Maurer D. (1998) Recognition of individual faces and average face prototypes by 1- and 3- month-old infants. Centre for Brain and Cognitive

- Development, Department of Psychology, Birkbeck College.

- Hadamard, J. (1923) Lectures on the Cauchy Problem in Linear Partial Differential Equations , Yale University Press

- Haralick, R.M. and Shapiro, L.G.. (1992) Computer and Robot Vision, Volume I. Addison-Wesley

- Haxby, J.V., Ungerleider, L.G., Horwitz, B., Maisog, J.M., Rapoport, S.I., and Grady, C.L. (1996). Face encoding and recognition in the human brain. Proc. Nat. Acad. Sci. 93: 922 - 927.

APPENDIX

main.py

```
import os # accessing the os functions
import check_camera
import Capture_Image
import Train_Image
import Recognize

# creating the title bar function
def title_bar():
    os.system('cls')  # for windows

    # title of the program

    print("\t**********************************************")
    print("\t***** Face Recognition Attendance System *****")
    print("\t**********************************************")

# creating the user main menu function
def mainMenu():
    title_bar()
    print()
    print(10 * "*", "WELCOME MENU", 10 * "*")
    print("[1] Check Camera")
    print("[2] Capture Faces")
    print("[3] Train Images")
    print("[4] Recognize & Attendance")
    print("[5] Auto Mail")
    print("[6] Quit")

    while True:
        try:
            choice = int(input("Enter Choice: "))

            if choice == 1:
                checkCamera()
                break
            elif choice == 2:
                CaptureFaces()
                break
            elif  choice  ==
                3:
                Trainimages(
                ) break
            elif choice == 4:
                RecognizeFaces()
                break
            elif choice == 5:
                os.system("py automail.py")
                break
                mainMenu()
```

```
            elif choice == 6:
                print("Thank You")
                break
            else:
                print("Invalid Choice. Enter 1-4")
                mainMenu()
        except ValueError:
            print("Invalid Choice. Enter 1-4\n Try Again")
    exit


#
# calling the camera test function from check camera.py file

def checkCamera():
    check_camera.camer()
    key = input("Enter any key to return main menu")
    mainMenu()



#
# calling the take image function form capture image.py file

def CaptureFaces():
    Capture_Image.takeImages()
    key = input("Enter any key to return main menu")
    mainMenu()



#
# calling the train images from train_images.py file

def Trainimages():
    Train_Image.TrainImages()
    key = input("Enter any key to return main menu")
    mainMenu()



#
# calling the recognize_attendance from recognize.py file

define RecognizeFaces():
    Recognize.recognize_attendence()
    key = input("Enter any key to return main menu")
    mainMenu()



#                                             main driver
mainMenu()
```

check_camera.py

```
def camera():
    import cv2

    # Load the cascade
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

    # To capture video from a webcam.
    cap = cv2.VideoCapture(0)

    while True:
        # Read the frame
        _, img = cap.read()

        # Convert to grayscale
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        # Detect the faces
            faces = face_cascade.detectMultiScale(gray, 1.3, 5, minSize=(30,
30),flags = cv2.CASCADE_SCALE_IMAGE)

        # Draw the rectangle around each face
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (10,159,255), 2)



        # Display
        cv2.imshow('Webcam Check', img)

        # Stop if escape key is pressed
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # Release the VideoCapture object
    cap.release()
    cv2.destroyAllWindows()
```

Capture_Image.py

```
import csv

import cv2
import os


# counting the numbers


def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass

    return False




# Take image function

def takeImages():


    Id = input("Enter Your Id: ")
    name = input("Enter Your Name: ")

    if(is_number(Id) and name.isalpha()):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector = cv2.CascadeClassifier(haar cascade Path)
        sampleNum = 0

        while(True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                        faces   =   detector.detectMultiScale(gray,   1.3,   5,
minSize=(30,30),flags = cv2.CASCADE_SCALE_IMAGE)
            for(x,y,w,h) in faces:
                cv2.rectangle(img, (x, y), (x+w, y+h), (10, 159, 255), 2)
                #incrementing sample number
                sample eNum = sampleNum+1
```

```
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage" + os.sep +name + "."+Id + '.' +
                            str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])
                #display the frame
                cv2.imshow('frame', img)
            #wait for 100 milliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is more than 100
            elif sampleNum > 100:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for ID : " + Id + " Name : " + name
        row = [Id, name]
            with open("StudentDetails"+os.sep"StudentDetails.csv", 'a+') as
csvFile:                                        writer =
                                                csv.writer(csvF
                                                ile)
                                                writer.writerow
                                                (row)
        csvFile.close()
    else:
        if(is_number(Id)):
            print("Enter Alphabetical Name")
        if(name.isalpha()):
            print("Enter Numeric ID")
```

Train_Image.py

```
import os
import time
import cv2
import numpy as np
from PIL import Image
from threading import Thread


#                 image labels

def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # print(imagePaths)

    # create empty face list
    faces = []
    # create empty ID list
    Ids = []
     # now looping through all the image paths and loading the Ids and the
images
    for imagePath in imagePaths:
        # loading the image and converting it to grayscale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces, Ids



#              train images function
def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, Id = getImagesAndLabels("TrainingImage")
    Thread(target = recognizer.train(faces, np.array(Id))).start()
    # Below line is optional for a visual counter effect
    Thread(target = counter_img("TrainingImage")).start()
    recognizer.save("TrainingImageLabel"+os.sep+"Trainner.yml")
    print("All Images")

# Optional, adds a counter for images trained (You can remove it)
def counter_img(path):
    imgcounter = 1
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    for imagePath in imagePaths:
        print(str(imgcounter) + " Images Trained", end="\r")
```

```
        time.sleep(0.008)
        imgcounter += 1
```

Recognize.py

```
import datetime
import os
import time

import cv2
import pandas as pd


#
def recognize_attendence():
              recognizer      =      cv2.face.LBPHFaceRecognizer_create()       #
cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel"+os.sep+"Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath)
    df = pd.read_csv("StudentDetails"+os.sep+"StudentDetails.csv")
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', 'Name', 'Date', 'Time']
    attendance = pd.DataFrame(columns=col_names)

    # Initialize and start realtime video
    capture cam = cv2.VideoCapture(0,
    cv2.CAP_DSHOW) 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, im = cam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.2, 5,minSize = (int(minW),
int(minH)),flags = cv2.CASCADE_SCALE_IMAGE)
        for(x, y, w, h) in faces:
            cv2.rectangle(im, (x, y), (x+w, y+h), (10, 159, 255), 2)
            Id, conf = recognizer.predict(gray[y:y+h, x:x+w])

            if conf < 100:

                aa = df.loc[df['Id'] == Id]['Name'].values
                confstr = "  {0}%".format(round(100 - conf))
                tt = str(Id)+"-"+aa



            else:
                Id = ' Unknown '
                tt = str(Id)
                confstr = "  {0}%".format(round(100 - conf))

            if (100-conf) > 67:
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
```

```
                                                            timeStamp       =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa = str(aa)[2:-2]
                attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]

            tt = str(tt)[2:-2]
            if(100-conf) > 67:
                tt = tt + " [Pass]"
                 cv2.putText(im, str(tt), (x+5,y-5), font, 1, (255, 255, 255),
2)
                            else:                       cv2.putText(im, str(tt), (x +
                                                        5, y - 5), font, 1, (255, 255,
255), 2)


                                        if (100-conf) > 67:
                                             cv2.putText(im, str(confstr), (x + 5, y
255, 0),1 )                                  + h - 5), font,1, (0,

                                        elif (100-conf) > 50:
                                            cv2.putText(im, str(confstr), (x + 5, y
                                            + h - 5), font, 1, (0,
255, 255), 1)
            else:
                cv2.putText(im, str(confstr), (x + 5, y + h - 5), font, 1, (0,
0, 255), 1)



        attendance = attendance.drop_duplicates(subset=['Id'], keep='first')
        cv2.imshow('Attendance', im)
        if (cv2.waitKey(1) == ord('q')):
            break
    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour, Minute, Second = timeStamp.split(":")
      fileName = "Attendance"+os.sep+"Attendance_"+date+"_"+Hour+"-"+Minute+"-
"+Second+".csv"
    attendance.to_csv(fileName, index=False)
    print("Attendance Successful")
    cam.release()
    cv2.destroyAllWindows()
```

automail.py

```
import os
import yagmail

receiver = "reviever@gmail.com" # receiver email address
body = "Attendence File" # email body
filename = "Attendance"+os.sep+"Attendance_2022-05-17_10-52-02.csv"    # attach
the file

# mail information
yag = yagmail.SMTP("your gmail_@gmail.com", "your password")

# sent the mail
yag.send(
    to=receiver,
    subject="Attendance Report", # email subject
    contents=body, # email body
    attachments=filename, # file attached
)
```