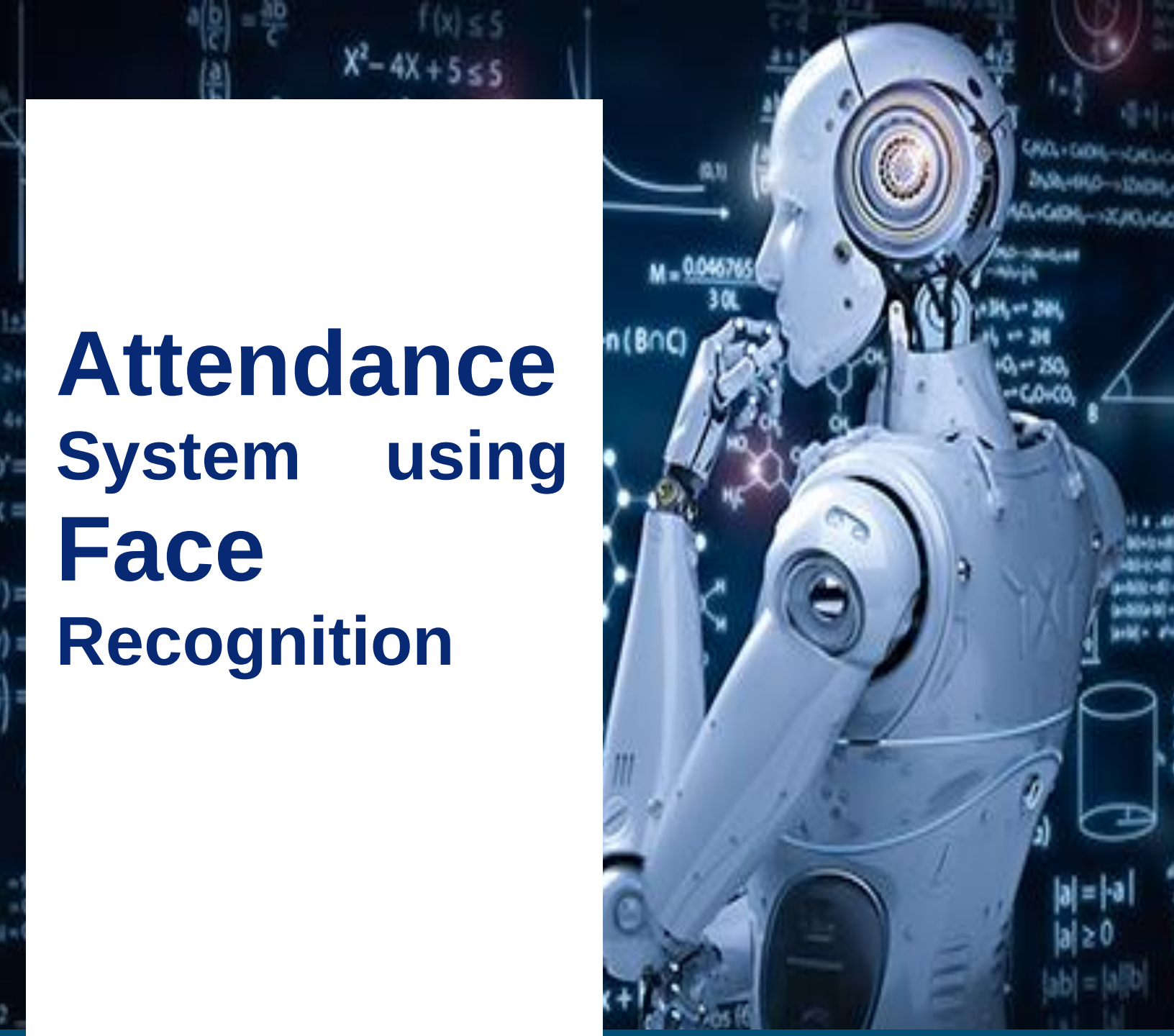

Attendance System using Face Recognition

JULY 6

Data Science Intern

Submitted by: Saifullah Rahimi

Solution Analysts

Analysing needs, delivering solutions

CONTENTS

FIGURE INDEX

1. SYNOPSIS

A facial recognition attendance system is a contactless technology that provides freedom from any physical interaction between the man and the machine. It is much easier to understand how attendance systems with face recognition can make buildings and premises safer and more efficient if we know how the technology works. This project involves building an attendance system that utilizes facial recognition to mark the presence, time-in, and time-out of employees. It covers areas such as facial detection, alignment, and recognition, along with the development of a web application to cater to various use cases of the system such as registration of new employees, the addition of photos to the training dataset, viewing attendance reports, etc.

Recognition of the human face is an active issue for authentication purposes specifically in the context of attendance of employees. An attendance system using face recognition is a procedure of recognizing employees by using face biostatistics based on high-definition monitoring and other computer technologies. The report explains the libraries and machine learning-based models and algorithms that have been used for facial detection and recognition. Explanation and use of Django, along with an SQLite database for web application development and database management have been provided. This project intends to serve as an efficient substitute for traditional manual attendance systems. It can be used in corporate offices, schools, and organizations where security is essential. The report also includes chapters covering project planning, a methodology adopted, and failures.

The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records. Present strategies for taking attendance are tedious and time-consuming. Attendance records can be easily manipulated by manual recording. The traditional process of making attendance and presenting biometric systems is vulnerable to proxies. This paper is therefore proposed to tackle all these problems, the modern face recognition attendance system is much more efficient and safer than its conventional counterparts. There are a few attendance systems that are based on face recognition technology and each of them has extremely varying features for the best suitability in different environments. it provides a facility for the central data management system. It can capture multiple images of faces at the same time and process them individually. It can tolerate certain facial movements up to an extent. That is, from frontal to a different direction. It is the safest and fastest method of security.

2. PREAMBLE

2.1 Introduction:

The human face plays an important role in our day-to-day life mostly for the identification of a person. Face recognition is a part of biometric identification that extracts the facial features of a face and then stores it as a unique face print to uniquely recognize a person. Biometric face recognition technology has gained the attention of many researchers because of its wide application. Face recognition technology is better than other biometric-based recognition techniques like a fingerprint, palm print, and iris because of its non-contact process. Recognition techniques using face recognition can also recognize a person from a distance, without any contact or interaction with the person. The face recognition techniques are currently implemented on social media websites like Facebook, at airports, and at railway stations. Then, in criminal investigations. Face recognition technology can also be used in crime reports, the captured photo can be stored in a database, and can be used to identify a person. Facebook uses the facial recognition technique for automating the process of tagging people. For face recognition, we require a large dataset and complex features to identify a person in all conditions like the change of illumination, age, pose, etc. Recent researches show there is a betterment in facial recognition systems. In the last ten years there is a huge development in recognition techniques Face Recognition is a biometric method of identifying an individual by comparing live capture or digital image data with the stored record for that person.

Smart Attendance System is marking of attendance based on this technology, which provides an automated attendance system that is practical, and reliable and eliminates disturbance and time loss of traditional attendance systems.

2.2 Statement of Problems:

Attendance is an important part of daily office evaluation. Face recognition-based attendance system is a problem of recognizing faces for taking attendance by using face recognition technology based on high-definition monitor video and other information technology. The concept of face recognition is to give a computer system the ability to find and recognize human faces fast and precisely in images or videos. Numerous algorithms and techniques have been developed for improving the performance of face recognition. Recently Deep learning has been highly explored for computer vision applications. The human brain can automatically and instantly detect and recognize multiple faces. But when it comes to computers, it is very difficult to do all the challenging tasks on the level of the human brain. Face recognition is an integral part of biometrics. In biometrics, the basic traits of humans are matched to the existing data. Facial features are extracted and implemented through algorithms, which are efficient, and some modifications are done to improve the existing algorithm models. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification, security systems, identity verification, etc. The face recognition system generally involves two stages:

- Face Detection – where the input image is searched to find any face, then image processing cleans up the facial image for easier recognition.
- Face Recognition – where the detected and processed face is compared to the database of known faces to decide who that person is.

Traditionally, employee attendance is taken manually by using an attendance sheet which is a time-consuming event. Moreover, it is very difficult to verify one by one employees in a large organization with distributed branches whether the authenticated employee is actually responding or not.

A smart Attendance system (SAS) using face recognition will present a system that can accurately evaluate employees' performance depending on their recorded attendance rate.

2.3 Objective and Scope of the Study:

Primarily the software will be based locally on the system to extract the features locally and then proceed with the identification of the face using those features on the server-side. The primary constraint of the software is correctly detecting and identifying the faces satisfying the detection threshold.

Furthermore, this project aims to automate the traditional attendance system where the attendance is marked manually. It also enables an organization to maintain its records like in-time, out-time, break-time, and attendance digitally. Digitization of the system would also help in better visualization of the data using graphs to display the no. of employees present today, the total work hours of each employee, and their break time. Its added features serve as an efficient upgrade and replacement over the traditional attendance system.

Facial recognition is becoming more prominent in our society. It has made major progress in the field of security. It is a very effective tool that can help low enforcers to recognize criminals and software companies are leveraging the technology to help users access the technology. This technology can be further developed to be used in other avenues such as ATMs, accessing confidential files, or other sensitive materials.

Using this system any corporate office, school and organization can replace their traditional way of maintaining attendance of the employees and can also generate their availability(presence) report throughout the month.

2.4 Module Description with Functionality:

2.4.1 Modules:

The features of the system are mainly divided into 3 modules.

1. Registration and Login Module

This module mainly deals with the functionalities related to the registration of any new employee to the organization, Logging into the system, and managing employee's profile details. Using features provided by this module admin can register new employees to the system and the admin/employee can log into the system using their credentials.

2. Manage Attendance Details

This module mainly deals with the features related to the employee's attendance. Using this employees can mark their presence, time-in, and time-out in the system. Admin can see the availability report of each employee, employee can see his/her attendance report along with some possible filters such as filter by employee and filter by date.

3. Manage Employee Details

This module mainly deals with the features related to the employee's profile. Using this admin can add a photo of the newly registered employee during registration. Admin can also command the system explicitly to train the model and the system will make necessary calculations and will generate some data which will be used internally to identify each employee uniquely.

2.4.2 Functional Requirements:

Manage Registration and Login:

Register new employee

Description: Admin can register new

Input: Employee Details

Output: success message displaying the user has been created.

Log-In to the system

Input: User credentials

Output: If the credentials are correct, the user will be redirected to the dashboard of the system

Exception Flow: If the entered credentials are incorrect then the user will be redirected to the login page again displaying an error message.

Manage Attendance Details:

Mark your attendance-in

Input: The user will scan his/her face using the external web camera.

Output: the system will identify the user uniquely and will mark his/her in-time to the database. The same success message will be transmitted to the user.

Mark your attendance-out

Input: The user will scan his/her face using the external web camera.

Output: the system will identify the user uniquely and will mark his/her out-time to the database. The same success message will be transmitted to the user.

View my attendance report

Description: Employees may often need to see his/ her attendance record throughout the month or year. Using this feature one can see his / her attendance record to the date.

Input: User selection

Output: Statistical analysis of the particular employee who is currently logged into the system will be displayed.

View employee's attendance report

Description: This feature is for admin. Admin can monitor the availability of each employee till the date. i.e., how many employees are present today out of total employees etc. can be monitored.
Input: user selection
Output: Attendance record of each employee including how many employees are present today out of total along with the availability graph.

Manage Employee Details:

Add photo of the employee

Description: Admin only can access this feature. Admin can add a photo of an employee during the registration process.
Input: Username of an employee
Output: Success message record has been added.
Process: The system will process an image and will generate necessary system data to identify each employee uniquely.

Train the system

Input: user selection

Output: the system will process all the available records of the employees and will generate necessary system data to identify each employee uniquely.

2.5 Methodology:

The face recognition system analyses the video feed frame by frame for the detection of faces. Then a predictor is used to identify the facial landmarks in the faces detected using a predefined template.

Using this template, the faces are aligned to increase the accuracy of the model. Now the unique embeddings of these aligned faces are generated. These embeddings are biometrics of the face and hence unique. These embeddings are classified using the classifier. This classifier is trained on the images collected from the users beforehand to ensure the smooth workflow of the program.

Significance of the Project

This project serves as a foundation for future projects based on facial detection and recognition.

The report covers the libraries incorporated in this project for testing and optimization of machine learning algorithms. It also compares algorithms with similar functionality. This project also covers web development and database management for a user-friendly UI.

We have used the Histogram of Oriented Gradient for face detection and deep learning techniques to calculate and compare 128-d face features for face recognition. Once faces are detected and recognized with the existing database, the system calculates attendance for the recognized employees.

Division of the report

The report has been divided into 4 sections.

The first section lays the foundation for Image Processing using OpenCV in live video streams.

The second section talks about different algorithms used for facial detection. It also covers facial landmarks used for face alignment to increase the accuracy of recognition. The third section discusses facial recognition models tested in this project. The fourth section marks the beginning of web development. It talks about the architecture, and project structure including the information on the management of databases and development of the front-end. Data visualization using graph plotting libraries and their examples are covered in the fifth section of the report.

2.5.1 Image Processing with OpenCV:

OpenCV can be installed by building from source which can be downloaded from OpenCV's website. Once it is downloaded, it can be imported using the command: import cv2

In order to load an image in OpenCV, the imread() method is used, in which the path of the image file is provided as a parameter. An image, once loaded, can be displayed using the imshow() method. Images are loaded as numpy arrays, and hence their dimensions can be retrieved using the shape attribute of numpy arrays. This provides another advantage - cropping images and extracting regions of interest is reduced to array slicing. Images can be resized using the resize() method. Rectangles, circles and text can be drawn onto an image using the rectangle(), putText() and circle() methods.

OpenCV loads images in BGR format which is not always desirable. Colorspace conversions to Grayscale and RGB formats is achieved easily using the cvtColor() method. All of these methods are used throughout the project for image manipulation.

Working with Video Streams

OpenCV provides VideoCapture and VideoWriter classes to capture videos and save them on disk, respectively. In this project, VideoCapture objects have been used to work with webcam streams. The VideoCapture() constructor takes one argument, which can either be a number, specifying the index of the camera device, or the path to a video file. Once a VideoCapture object is created, it can be processed frame by frame, allowing us to apply the same image processing techniques that we use for static images. The video capture can be released with the release() method.

2.5.2 Facial Detection and Alignment:

Face detection is the technology of recognizing human faces in a photo, video, or live frame.

Face detection can be said as a specific case of object-class detection. Face detection is the first step in the process of facial recognition. Face alignment, which can be thought of as data normalization, is a part of image pre-processing, wherein, after detecting the face, the image is cropped, rotated, scaled, and adjusted so as to provide a normalized input to the face classifier.

Haar and LBP Classifier

A classifier is defined as a program that can discriminate between a positive image and a negative image. Classifiers are trained on many positive (face containing) and negative images, so as to classify a new image correctly. OpenCV2 provides two pre-trained classifiers

1) Haar Classifier 2) LBP Classifier

1) **Haar Classifier**

Haar classifier works by extracting Haar features from the input image. The following figure illustrates the three kinds of Haar features and how they are used to extract features

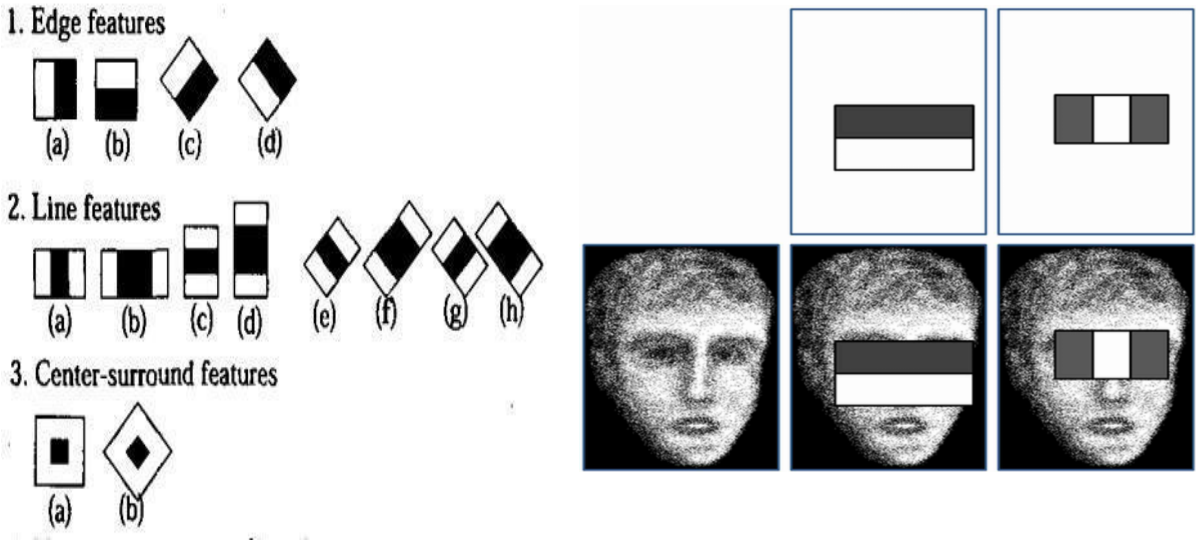

Figure 2.1: Placing of haar features on image to calculate features

Each of these windows is placed on the input image to calculate a single feature. A feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle. A large number of features are thus calculated for different positions and sizes of kernels on the entire image. Only relevant features are kept and irrelevant features are discarded through a technique called AdaBoost4.

2) **LBP Cascade Classifier**

LBP (Local Binary Patterns) is a visual/texture descriptor. It works by dividing the training image into blocks and analyzing 3*3-pixel areas for each block. The center pixel's value is compared with neighboring pixels in the 3*3 window. The value of neighbor pixels whose current value is greater than the center pixel's value is set to 1. Others are set to 0. Updated pixel values are then read in a clockwise manner and converted to a binary number. This number is converted to its decimal equivalent. The following figure illustrates this process.

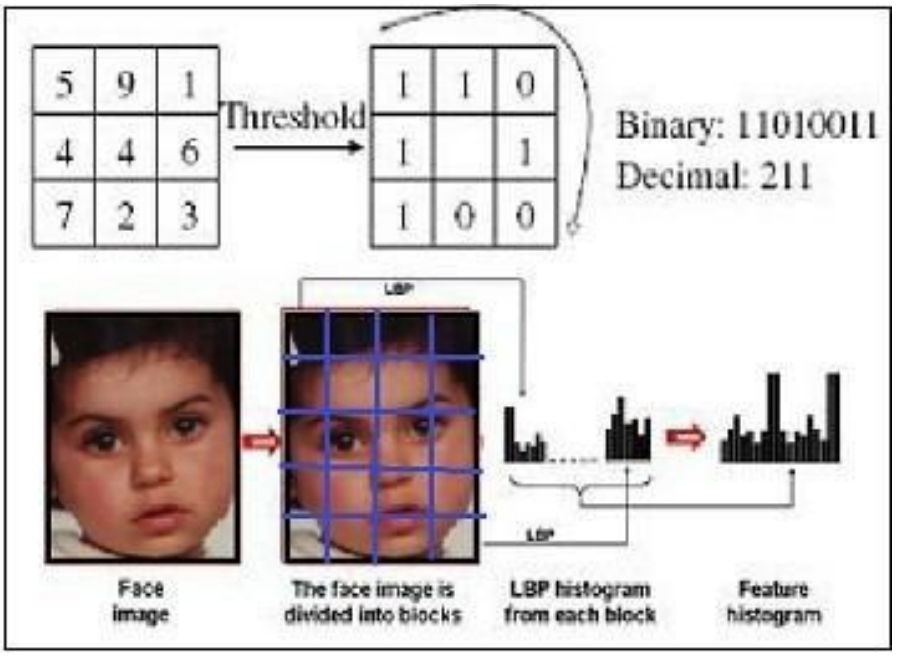

Figure 2.2: LBP Conversion from binary to decimal

The block values so obtained are converted to a histogram. Finally, these block histograms are concatenated to form one feature vector for one image.

Dlib's Detector

Dlib is a general-purpose cross-platform software library written in C++. Its libraries are available for use in different programming languages. The Object detection task can be achieved using Dlib's detectors such as Histogram of Oriented Gradients (HOG), or Convolutional Neural Network (CNN) based face detector.

Histogram of Oriented Gradients (HOG) detector

HOG is a feature descriptor used in computer vision and image processing for the purpose of object detection. It works on the concept of finding gradients along the x and y-axis. These gradients are added vectorially and information such as the magnitude and direction of the gradients are extracted forming edges in the image. Now the whole gradient image is divided into an 8*8- dimension matrix i.e. carrying 64 gradient vectors. Each matrix contains 9 bins and each bin has a range of 40 degrees to allocate the gradient vector lying in that range. After each gradient vector has been allocated a bin, a histogram of this matrix is formed which determines the overall direction and magnitude of the gradient. Luminosity plays a great role in the proper detection of gradients. To overcome this failure the value of the histogram is obtained in the red, green, and blue frame and then normalized making it lighting invariant. Finally, the normalized histogram is compared with its default histogram for faces to identify a face in an image.

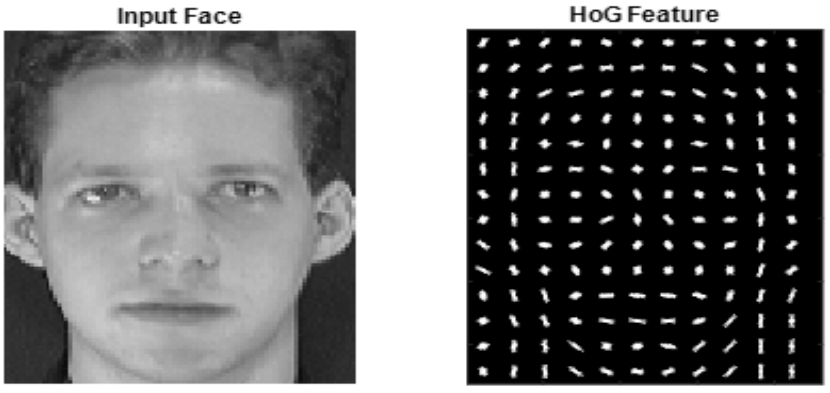

Figure 2.3: HOG Face detector

The following code is used to initialize the HOG based face detector:

```
hog_face_detector = dlib.get_frontal_face_detector()
```

2.5.3 Facial Alignment using Landmark Predictors:

Face Alignment is the process of identifying the geometric structure of faces in digital images and then attempting to obtain an alignment of the face based on rotation, scale, and translation. There are many methods to imply facial alignment. In this project, face alignment is achieved with the help of facial landmarks.

Facial detection is an important step before facial recognition as most face recognition algorithms can benefit highly from applying facial alignment and normalizing the input. Higher accuracy is obtained when face alignment is used.

Dlib has a 5 point and 68 points facial landmark predictor. It extracts important landmarks on the face like left eye start point, the center of the nose, etc. The facial landmark points help us to perform operations using the point coordinates. The operations performed for facial alignment are: -

1) Rotation, such that the eyes lie on a horizontal line

2) Scaling, such that the size of faces is approximately identical

3) Translation, such that the image is centred

In this project, 68 points facial landmark predictor is used. The results can be seen in the following figure :-

Figure 2.4: Facial Alignment results

Code to initialize a 68-point landmark predictor:

```
predictor = dlib.shape_predictor('./shape_predictor_68_face_landmarks.dat')
```

2.5.4 Facial Recognition:

A facial recognition system typically works by extracting face encodings from faces and comparing them with an existing database of known faces to find a match. Given an unknown image, the process of facial recognition comprises detecting faces in that image, aligning them, extracting feature vectors from aligned faces, and classifying each feature vector as belonging to one class from a set of classes.

In order to extract feature vectors, or face encodings, the team used Convolutional Neural Network-based models. Three different approaches were tested by the team:

1. Training our own CNN-based object classifier (face classifier)

2. Face_recognition library by Adam Geitgey

1) Training Our Own CNN-based Object Classifier

Using Keras to build the CNN

Keras is a high-level neural networks API, written in python, that runs on the Tensorflow backend. The core data structure of Keras is a model which is a way to organize layers of a neural network. For this project, the sequential model available in Keras was used. Convolution2D, MaxPooling2D, Flatten, and Dense layers were also used.

Data Augmentation

Data augmentation is a way to artificially expand the size of training data by creating modified versions of the images in the dataset. Data augmentation helps to increase the performance of the CNN as it improves with the amount of data available. Transforms include a range of operations from the field of image manipulation like shifts, zooms, flips, etc. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.

Training and Testing

Once the CNN is built, it is trained in a number of classes, each having several training images. Once trained with images and associated labels, the CNN is expected to predict the associated label for unknown image input. The training time of the CNN depends on the number of training images, the size of the batch, the number of times epochs, etc. In order to obtain high accuracy, it is necessary to train the CNN on a large dataset, which is computationally expensive and takes a lot of time. When trained and tested on a small dataset, the team was unable to get accurate results. Hence, for this project, the team decided to use pre-trained CNNs to extract encodings from images.

2) Face Recognition library

Face Recognition library works on deep metric learning10 tools to identify, train and classify test data. It is the world's simplest library for the recognition and manipulation of faces. This model showed an accuracy of 99.38% on the Labeled Faces in the Wild benchmark indicated by the python software foundation. The deep neural network used by it represents the face on a 128- dimensional unit hypersphere. The embedding is a generic representation of anybody's face.

Unlike other face representations, this embedding has the nice property that a larger distance between two face embeddings means that the faces are likely not of the same person. This property makes clustering, similarity detection, and classification tasks easier than other face recognition techniques where the Euclidean distance between features is not meaningful. This project has incorporated this library to extract the face encodings and face location from the images.

Code for Generating embeddings using Face Recognition library:

```
X_face_locations = face_recognition.face_locations(face_aligned)

Faces_encodings = face_recognition.face_encodings(face_aligned,
known_face_locations=X_face_locations)
```

3. REVIEW OF LITERATURE

This project involves building an attendance system which utilizes facial recognition to mark the presence, time-in, and time-out of employees. It covers areas such as facial detection, alignment, and recognition, along with the development of a web application to cater to various use cases of the system such as registration of new employees, the addition of photos to the training dataset, viewing attendance reports, etc. This project intends to serve as an efficient substitute for traditional manual attendance systems. It can be used in corporate offices, schools, and organizations where security is essential.

Web Application Development and Database Management

The project involved the creation of a web application that would work as an attendance system that uses the facial recognition model described in previous sections. For the development of this application, Django was used. Django is a high-level Python-based, open-source web framework that allows the creation of websites with ease. It follows the Model-Template-View (MTV) Architecture. Along with being secure, portable, scalable, and versatile, it takes care of the intricacies and hassle of web development, so that the developer can focus solely on writing the application.

3.1 Project structure:

The command Django-admin start project my_project followed by Django manage.py startup my_app is used to initialize a Django project named ‘my_project’ consisting of one application named ‘my_app’.

The views, models, and templates are thus separated, allowing us to harness the power of MTV architecture. The models.py file is where we define our database models. The templates contain HTML files that create the UI layer. The views.py file is used to communicate between the models and the templates. It consists of different functions, or views, which take in web requests and return web responses. As a result, the HTML contents of the associated template, along with linked CSS and JS files, are displayed as a web page, at the url mapped to that view.

This URL mapping is specified in the urls.py file. Django views are linked with their corresponding contents using the render function, and data to be displayed is passed using the Django templating language. The project thus has the following structure:

myproject/

manage.py

myproject/

__init__.py

settings.py

urls.py

Wsgi.py

myapp/

__init__.py

admin.py

models.py

tests.py

views.py

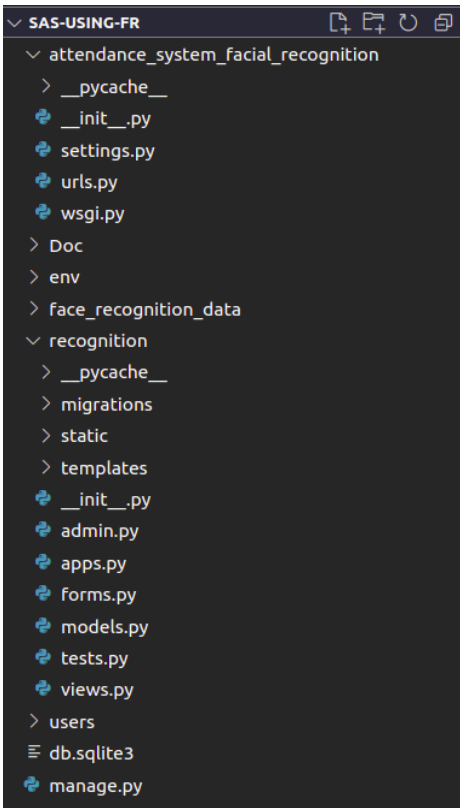

Figure 3.1: Project thus structure

3.2 Database and Model:

This project uses the built-in User model provided by Django, which stores the information of all employees. This model can be imported using the following command:

from django.contrib.auth.models import User

This project also uses two other model classes 'Present' and 'Time' that store the Attendance (present/absent) and time-in, and time-out, respectively of all employees. The following code is used to create these two models:

1. class Present(models.Model)

```
user=models.ForeignKey(User,on_delete=models.CASCADE)

date = models.DateField(default=datetime.date.today)

present=models.BooleanField(default=False)
```

2. class Time(models.Model):

```
user=models.ForeignKey(User,on_delete=models.CASCADE)

date = models.DateField(default=datetime.date.today)

time=models.DateTimeField(null=True,blank=True)

out=models.BooleanField(default=False)
```

The user field acts as a foreign key to the User model. The date field stores the current date.
The present field is a Boolean field that stores True for the present, and False for absent. The time field stores the time of entry or exit, depending upon the value of the outfield, which stores True for time-out, and False for time-in.

3.3 Front end with Bootstrap and CSS:

Django is a very powerful framework, but it can't be used for front-end development. In order to develop the website's UI and UX, Bootstrap and CSS styling were used. These CSS files were stored in a separate static folder within the Django project. CSS- Cascading Style sheets are used for describing the presentation of a document written in a markup language such as HTML.

It allows the addition of styles to a webpage with the use of classes and IDs. Bootstrap is a front-end framework that simplifies the process of adding styles, by providing built-in CSS classes that can be added to HTML elements, allowing the developer to forgo the process of writing complicated CSS.

For this project, Bootstrap was added via the Bootstrap CDN. The URL of Bootstrap CSS files is simply added within the <link> tag of the HTML file, and Bootstrap classes can then be used throughout the file. The Bootstrap typography, navbar, table, button, card, container, and alert classes were used extensively throughout the project.

Django Template Language

Django Template Language is used to send data from the view to the templates. It provides the developer with tags that function similarly to some programming constructs - such as the 'if' tag for boolean tests and the 'for' tag for looping. The templating language has been used throughout the project for interaction between the view and the templates.

3.5 Tabular Representation of Attendance Data:

The application provides the admin with the feature to view attendance reports, where in attendance data is displayed as tables. This is achieved via Django's ORM and Templating Language.

Data is fetched from the attendance database in the form of query sets, making use of Django's ORM. These query sets are then passed to the template (HTML) files using Django's templating language. These query sets can be displayed as a table by using the {% for tag %} of the templating language for displaying rows and outputting each attribute of the object in a different table cell (column). The table formed using query set and its attributes are displayed in the following figure table).

Date	Employee	Present	Time in	Time out	Hours	Break Hours
April 27, 2022	saifullah_rahimi	P	April 27, 2022, 9:11 a.m.	April 27, 2022, 9:44 a.m.	0 hrs 33 mins	0 hrs 0 mins
April 26, 2022	saifullah_rahimi	P	April 26, 2022, 12:13 p.m.	April 26, 2022, 5:56 p.m.	5 hrs 43 mins	0 hrs 0 mins

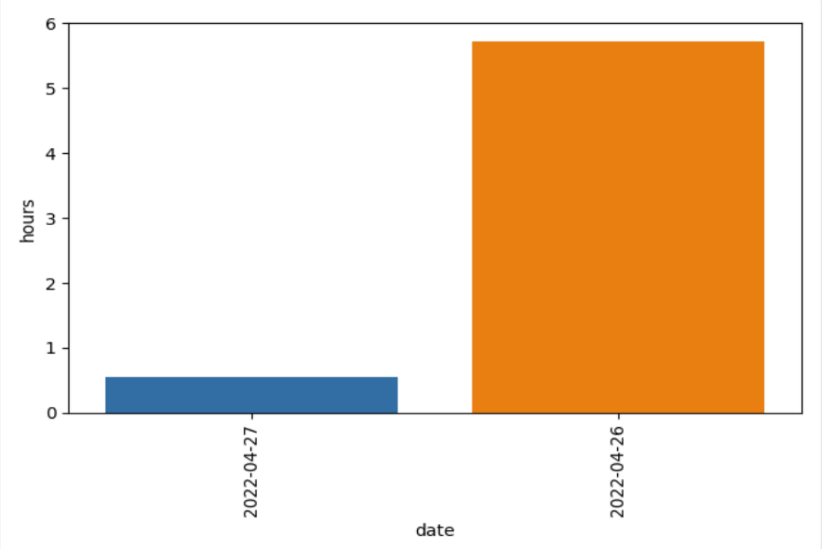

Figure 3.2: Tabular Representation of attendance data

Data Visualization

One of the features of this application is that it allows the admin to view attendance data in the form of graphs and plots. For this purpose, Matplotlib and Seaborn libraries have been used. Data is first fetched from the database in the form of query sets, using Django. Query sets can be converted to Pandas data frame objects using the django_pandas library. These data frames are then used as direct inputs to Seaborn graphing functions.

Matplotlib and Seaborn

Matplotlib is a 2D Python 2D plotting library that can be used to generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc. from datasets. Seaborn is a graphical library built on top of Matplotlib. Seaborn provides a high-level interface to draw statistical graphics. Seaborn functions operate on data in the form of Pandas dataframes.

Both Matplotlib and Seaborn are pip installable.

pip install matplotlib seaborn

Examples

1. A Seaborn bar plot demonstrating the number of hours worked by each employee on a particular date is displayed in the following figure.

Date	Employee	Present	Time in	Time out	Hours	Break Hours
April 26, 2022	anupam_kumar	P	April 26, 2022, 12:18 p.m.	April 26, 2022, 5:56 p.m.	5 hrs 38 mins	5 hrs 37 mins
April 26, 2022	saifullah_rahimi	P	April 26, 2022, 12:13 p.m.	April 26, 2022, 5:56 p.m.	5 hrs 43 mins	0 hrs 0 mins

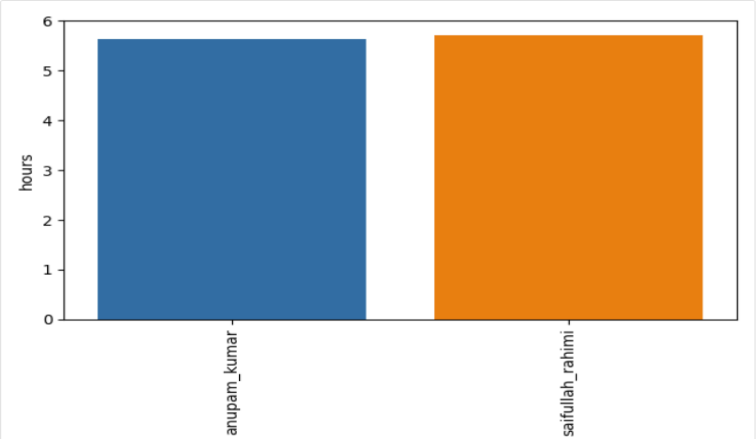

Figure 3.3: Hours worked vs. username for a particular date

2. A Seaborn line plot demonstrating the number of hours worked by a particular employee for a range of dates is displayed in the following figure. Note that it allows easy visualization of the weekly minimum.

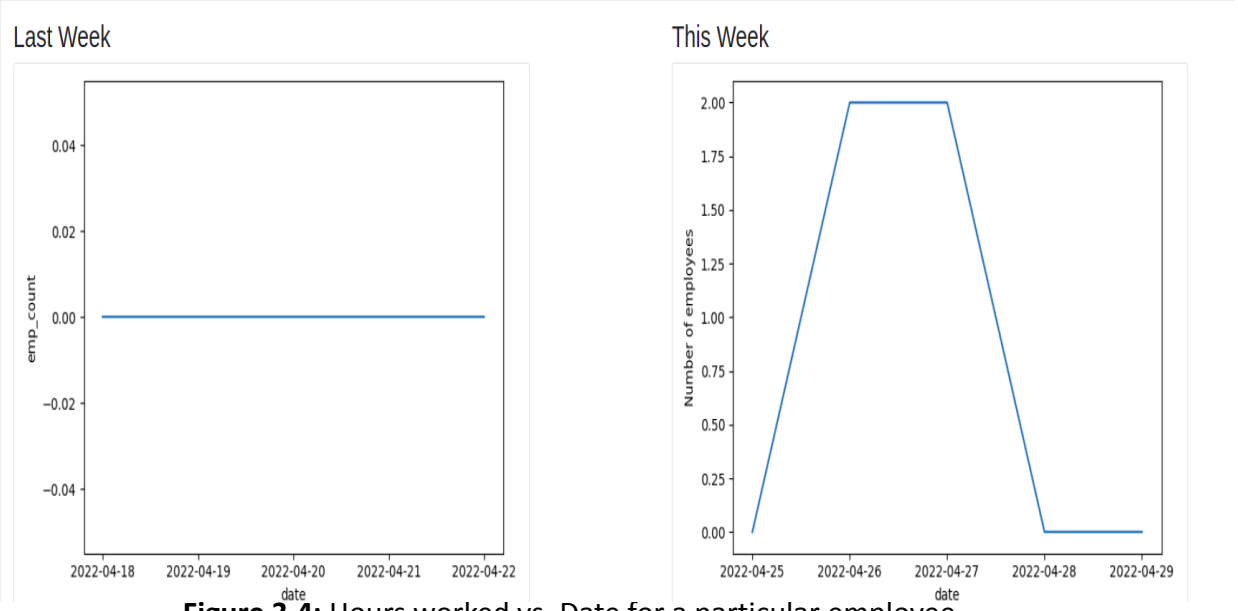

Figure 3.4: Hours worked vs. Date for a particular employee

4. CONCLUSION

Smart Attendance System using Face Recognition is designed to solve the issues of existing manual systems. I have used the face recognition concept to mark the attendance of employees and make the system better. The system performs satisfactorily in different poses and variations. In the future this system need to be improved because these systems sometimes fail to recognize employees from some distance, also we have some processing limitation, working with a system of high processing may result in an even better performance of this system. There may be some error or some defect in the work. The system has taken enough care to make the project user-friendly and more interactive.

This paper has come up with some great features. As the system used the Haar Cascade model for feature extraction. This model is quite good for feature detection as it used integral images and boosting algorithm that makes the detection process faster and helps to choose the best features. So that, no other extra things come to the frame when the machine is being trained. Haar Cascade with LBPH can recognize a face from any angle even if an employee is not looking at the camera. The system can recognize faces when an employee has glasses on. It can also take attendance for multiple employees at the same time. This system is very efficient for any organization. I believe that organizations will be benefited from using this application because of its easy access to this application.

5. LEARNING DURING PROJECT WORK

Smart Attendance System Using Face Recognition was quite challenging for me as this project include much enough functionalities and components of AI Concepts. So this project enabled me to learn new concepts in various parts which I will point out some of them here:

Learnt to demonstrate skill and knowledge of current information and technological tools and techniques specific to the professional field of study.

Learnt to identify, analyse and solve problems creatively through sustained critical investigation.

Learnt to integrate information from multiple sources.

I realized that practice the skills, diligence, and commitment to excellence needed to engage in lifelong learning.

6. FUTURE ENHANCEMENTS

I think that not a single project is ever considered complete forever because our mind is always thinking new and our necessities also are growing. if you see at the first glance that you find it to be complete but I want to make it still mature and fully automatic. The system is modified in the future as per the owner's requirement.

1. A feature that can give intruder alerts can be included in the system. Furthermore, the images of unknown people can be saved in an efficient manner and displayed in the system for better security.

2. The number of training images can be reduced so that less storage is required. This can be done by removing duplicate images of the same person, or images with similar embedding.

3. The training time can be reduced by retraining the classifier only for the newly added images.

4. A feature can be added where an employee is automatically sent a warning if his attendance or working hours are below the threshold.

5. Wrongly classified images can be added to the training dataset with the correct label so as to increase the accuracy of the recognition model.

7. BIBLIOGRAPHY

Bibliography means that I have referred while doing a project, which sites we have gone through and which books I have referred to, or which magazines I have seen to implement this project. There are many books and websites that can help us with the proper guild line to implement my system in the right direction.

10.1 Online References

1. King, Davis. “High Quality Face Recognition with Deep Metric Learning.” Dlib, blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html.

2. Krasser, Martin. Deep Face Recognition with Keras, Dlib and OpenCV. krasserm.github.io/2018/02/07/deep-face-recognition/.

3. Kumar, Satyam. “Face Recognition Using OpenFace.” Medium, medium.com/@krsatyam1996/face-recognition-using-openface-92f02045ca2a. Link to Bootstrap website - for CDN add later

5. Mallick, Satya. “Histogram of Oriented Gradients.” Learn OpenCV, www.learnopencv.com/histogram-of-oriented-gradients/.

6. Pupale, Rushikesh. “Support Vector Machines(SVM) — An Overview.” Towards Data Science, towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989.

7. Rosebrock, Adrian. “Face Alignment with OpenCV and Python.” Pyimageserach, www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/.

8. Symanovich, Steve. “How Does Facial Recognition Work?” Norton by Symantec, us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html.

9. ujjwalkarn. “An Intuitive Explanation of Convolutional Neural Networks.” The Data Science Blog, ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

10.2 Offline References

References Books:

Face Detection and Recognition: Theory and Practice

Book by Asit Kumar Datta, Madhura Datta, and Pradipta Kumar Banerjee