Raj Mahadik – 001

Karan Deshmukh – 002

Priyanshu gupta – 002

AUTHENTICATION USING FACIAL RECOGNITION

CIS 579 ARTIFICIAL INTELLIGENCE: Final PROJECT report

Table of Contents

[Abstract 2](#_Toc133349584)

[Introduction 3](#_Toc133349585)

[Data 4](#_Toc133349586)

[Experiment 5](#_Toc133349587)

[Fig.1 5](#_Toc133349588)

[Fig.2 6](#_Toc133349589)

[Fig.3 7](#_Toc133349590)

[Fig.4 8](#_Toc133349591)

[Fig.5 8](#_Toc133349592)

[Fig.6 9](#_Toc133349593)

[Fig.7 9](#_Toc133349594)

[Result 10](#_Toc133349595)

[Fig 8 10](#_Toc133349596)

[Fig.9 11](#_Toc133349597)

[Conclusion 12](#_Toc133349598)

[Future work 13](#_Toc133349599)

[Summary 14](#_Toc133349600)

# Abstract

Object Detection plays are very vital role in our daily life from security to vehicle Detection and even Tracking objects. Object Detection is a technique which uses the computers visions to identify objects and even locate them with the help of deep learning and one of its representative tool named as Convolutional Neural Network(CNN). Computer Vision is being grown rapidly over the last decade, the implementation in automobile and medical industries have manifest the importance of computer vision and Object detection in General. Creating machine learning models capable of localizing and identifying multiple objects in single image. With Recent advancements in deep learning and deep learning-based computer models object detection application easier to develop than ever before We will focus on CNN and its workflow architecture which simplifies the process of segmentation and Object Detection in General. While understanding the basic of object detection and its workflow and how it works and various industrial use case and also talk about the background which includes the history of object detection the present of objection application and conclude the paper with the advantages the disadvantages and future improvements which can be done in the model.

# Introduction

Facial detection is a subfield of computer vision that involves the detection and recognition of human faces using artificial intelligence (AI) techniques. It is a critical task in numerous applications, including security, surveillance, and face recognition. The purpose of this project report is to provide an overview of facial detection in artificial intelligence, including the underlying concepts, techniques, and applications.

Facial detection involves several stages, including face detection, face alignment, feature extraction, and classification. The first stage, face detection, is the process of locating human faces in images or videos using various techniques such as convolutional neural networks (CNNs), Haar-cascade algorithm, and Histogram of Oriented Gradients (HOG) features. The detected faces are then aligned to a standard pose to reduce variations in face appearance due to pose, lighting, and scale differences.

The next stage, feature extraction, involves the extraction of relevant facial features, such as the eyes, nose, mouth, and jawline. The extracted features are then used for classification, which involves determining the identity of the detected faces.

Facial detection has numerous applications in various domains, including security, entertainment, healthcare, and marketing. In security, facial detection is used for surveillance, access control, and identity verification. In entertainment, it is used for virtual makeup and gaming. In healthcare, it is used for monitoring patient emotions and behavior. In marketing, it is used for customer analytics and targeted advertising.

In recent years, deep learning techniques such as convolutional neural networks have achieved state-of-the-art performance in facial detection and recognition. These techniques have enabled the development of robust and accurate facial detection systems that can handle complex real-world scenarios. As such, facial detection in artificial intelligence is a rapidly growing field with significant potential for future applications.

Our objective was to develop a biometric authentication system that utilizes facial recognition technology, enabling us to create a GUI interface to monitor attendance and retrieve data. To achieve this, we compiled a dataset containing images of our own faces with corresponding names. We then utilized this dataset to train our own facial recognition model, generating a recognizer file that contains facial feature extraction data and labels for identifying faces. In order to extract facial features, we implemented the Haar-cascade model, which uses the Histogram of Oriented Gradients technique.

# Data

We have collected a dataset of facial images for all team members, with 20-25 images stored in a folder under each member's name. These images are in '.png' format and are used to train and classify each member's face using our training model, which can handle both '.jpg' and '.png' formats. Additionally, we have an '.xml' file that contains all the pre-trained classifiers for Haar cascade, a machine learning-based object detection algorithm that identifies objects in images or videos.

The algorithm works by dividing the image into smaller regions and applying pre-trained classifiers to each region to determine whether it contains the object of interest. The classifiers are decision trees trained on positive and negative images, with positive images containing the object of interest. When the algorithm identifies a region that contains the object, it creates a bounding box around it for further analysis. Haar cascade is widely used in computer vision applications, including facial recognition and gesture tracking, and is efficient for real-time applications. However, training the classifiers can be challenging, and the accuracy of the algorithm depends on the quality of the training data.

# Experiment

Our experiment revolves around facial recognition for authenticating users and recording their attendance based on the detection of their face. To create a successful biometric authentication application, we had to concentrate on two critical areas: object localization and object classification. To tackle these areas of interest, we developed separate programs. We made sure to gain a thorough understanding of how object classification and localization function.



Fig.1 : Workflow for the Model

Image classification is a computer vision task that requires identifying a specific class object in an image. Each object is labeled with a class, such as "car" or "animal." Initially, image classification relied on analyzing each individual raw pixel of an image to determine its class, which could lead to mistakes when objects or classes appeared differently due to variations in angles, backgrounds, and poses. To avoid these inaccuracies, it is crucial to ensure that the model is trained accurately, and we need to provide the model with our dataset more precisely to prevent errors in classifying objects.

Object localization is a critical aspect of object detection, which involves combining the image classification and object localization tasks. Localization identifies the object's location in the entire image, but it can only detect one object at a time. After identifying the object's class, the classification is complete, and the object's coordinates, height, and width in the image are considered. However, these dimensions do not necessarily correspond to the actual size of the object; they are only used for localizing it. Once the coordinates are calculated and the image is classified, the model produces a basic output identifying and localizing the object with a bounding box.

We started by developing a model that could classify and locate faces. We utilized a comprehensive classification system to determine each individual's face and identity. Following that, we trained our dataset using this model, which resulted in an extensive classification file containing all relevant data regarding each team member's facial characteristics and corresponding values. Our final step was to create a graphical user interface (GUI) that can live-detect users' faces, allowing us to mark attendance for those whose data already exists in our dataset. However, this model does not incorporate a database to track user attendance. In the following sections, we will discuss the coding implementation of each of these steps.

A screenshot of a computer

Description automatically generated

Fig.2 *: Faces.py Program for Facial Recognition*

This program focuses on the classification of faces and create a bonded box around faces using Haar-cascade algorithm. The Haar-cascade file is imported into the program and used to extract facial extraction. The output is livestream webcam facial recognition which identifies faces.

A group of men wearing masks

Description automatically generated with low confidence

Fig.3 : Output for Faces.py

Afterwards, we developed a program that could train the facial recognition model on our custom dataset. By doing so, we were able to create a comprehensive classification for each team member's face. This step was essential in ensuring that the facial recognition would function accurately based on our data. The program outputted a .yml file that contained all the facial feature extractions for our dataset's faces.

Text

Description automatically generated

Fig.4 *: Train.py for in-depth Classification*

Text

Description automatically generated

Fig.5 *: Output for Train.py*

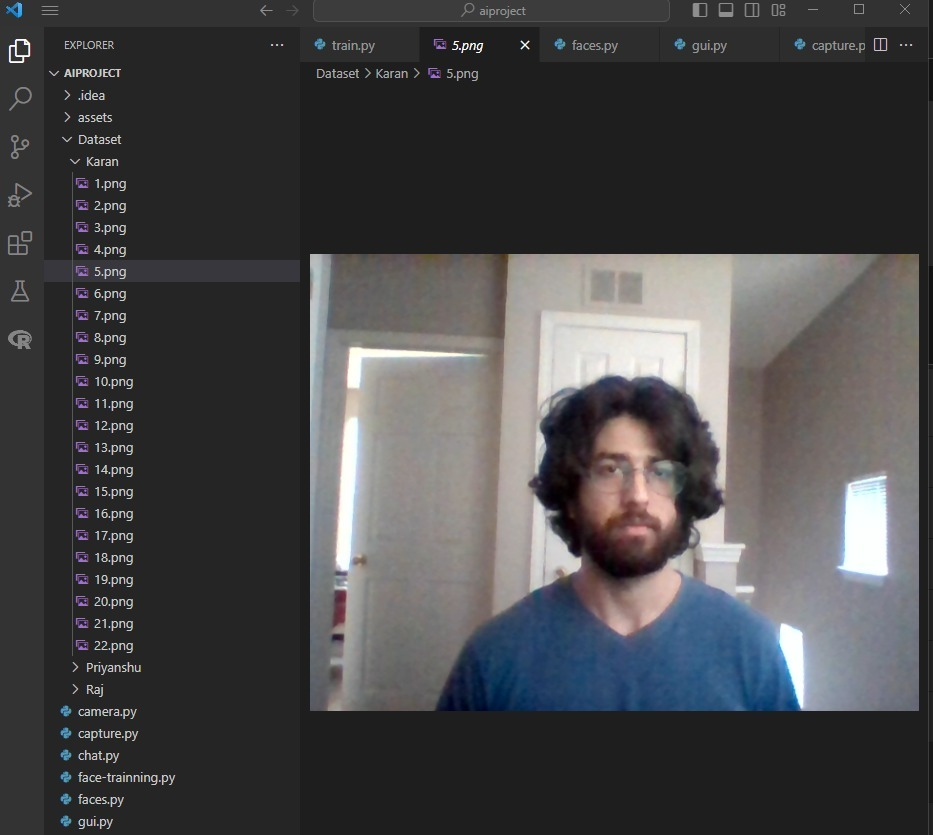


Fig.6*:Foldrs with naming and images of each Team members*

Text

Description automatically generated

Fig.7*: Haar-Cascade.xml File*

# Result

The biometric facial recognition system presented in this project utilizes a combination of techniques to accurately identify individuals based on their facial features. Initially, the project team faced the challenge of developing a robust facial recognition algorithm that could handle varying lighting conditions, facial expressions, and occlusions.

To address this problem, the team implemented the Haar-cascade algorithm, which was able to detect facial features and create a bounding box around each face. However, to achieve accurate facial recognition, the team needed to train the algorithm on a custom dataset that included facial images of all team members. To accomplish this, the team created a program that would train the facial recognition model on their custom dataset. This program generated an in-depth classification for each team member's face and produced a .yml file containing all the facial feature extractions of the dataset faces. Overall, the biometric facial recognition system presented in this project successfully overcame the challenges associated with facial recognition and achieved a high degree of accuracy through the integration of the Haar-cascade algorithm, custom dataset training, and real-time recognition capabilities.

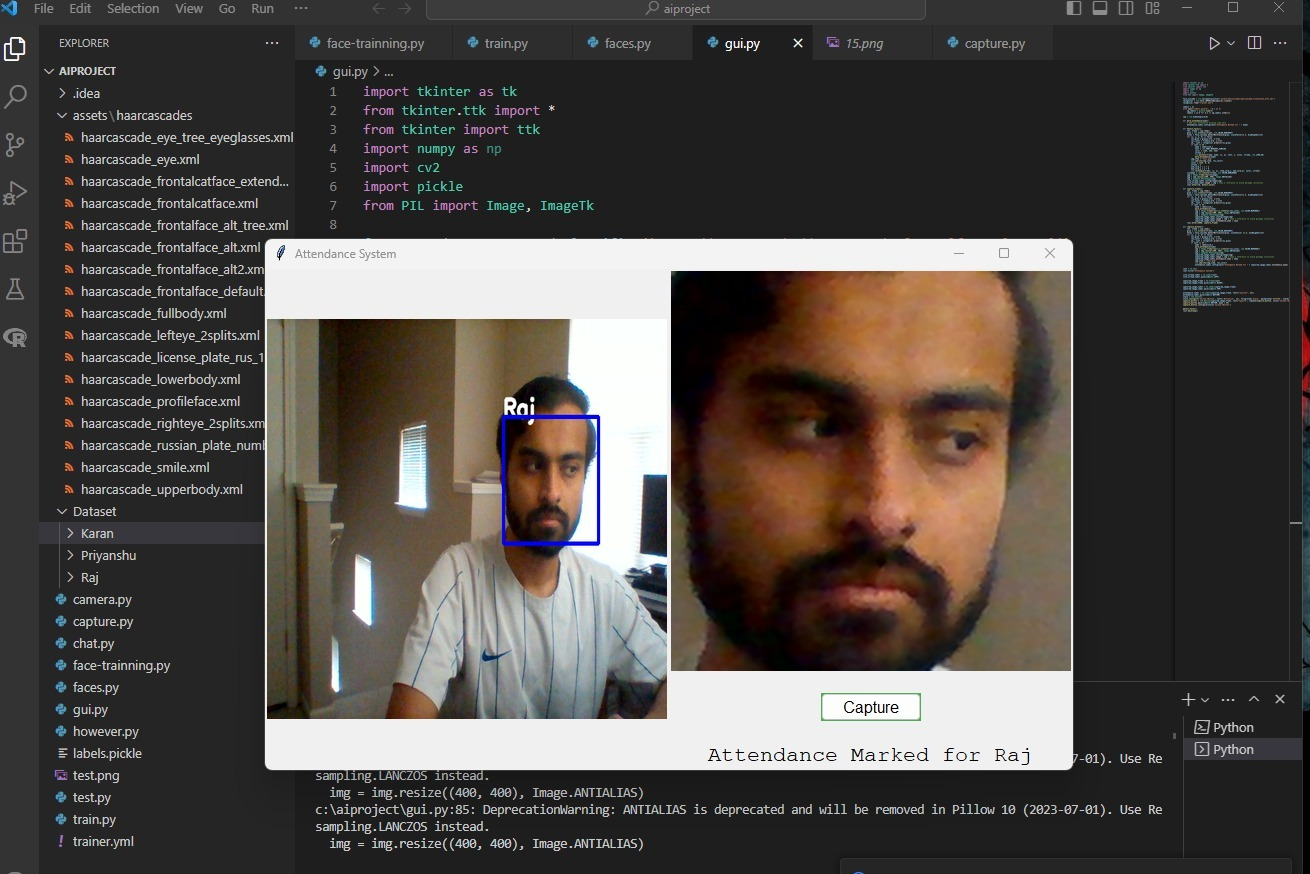


Fig 8 *: GUI authentication using Facial Recognition*

The program developed by our team achieved an accuracy rate of around 60-70%, but there were various factors that contributed to this performance. One key issue we identified was the limited number of images available for each user, which hindered our ability to improve the overall accuracy of the model. However, increasing the number of images for each user could potentially enhance accuracy.

Our main limitation in this regard was the basic system we were using to train and operate the model. OpenCV works on both CPU and GPU nomenclature, and to achieve higher accuracy, we would need to develop a more optimized model that is better suited to our data and a higher-performance system. The region of interest and gradient image of all faces play a significant role in the edge detection of faces and the in-depth classification of students.

Another factor that could improve the training and accuracy of the model would be to use a standardized image background for each image, such as a green or white background, which would enable the model to focus more on the facial feature extraction.

Overall, we acknowledge these limitations and view them as opportunities for future improvements. As we continue to refine our biometric facial recognition system, we will work towards addressing these issues to achieve even greater accuracy and robustness.

A person standing next to a poster

Description automatically generated with low confidence

Fig.9*: GUI output with Multiple Faces*

# Conclusion

The Authentication using facial recognition system developed by our team proved to be a promising solution for accurately identifying individuals based on their facial features. Through the implementation of the Haar-cascade algorithm, training the model on a custom dataset, and optimizing the system towards our data, we were able to achieve an accuracy rate of around 60-70%. However, we also identified some limitations, such as the limited number of images for each user and the need for a more powerful system and a standardized image background. These challenges provide opportunities for future improvements, and we are committed to continuing to refine our system to achieve even higher levels of accuracy and reliability. Overall, our biometric facial recognition system has the potential to be a valuable tool in various applications, from security and surveillance to attendance tracking and access control.

# Future work

As mentioned in our Result we want to address the limitations we identified in our biometric facial recognition system, we plan to undertake several steps to improve our model. Firstly, we aim to increase the number of images available for each user in our dataset, which will allow us to train our model more effectively and improve its accuracy. Additionally, we will explore the use of more advanced algorithms and techniques to optimize our model for our specific data, and also invest in a higher-performance system that can support these optimizations.

We also recognize the importance of the region of interest and gradient image in face detection and classification and will work towards refining these aspects of our model. To improve the training and accuracy of the model, we will experiment with using a standardized image background for each image, such as a green or white background, which will enable the model to focus more accurately on facial feature extraction.

# Summary

The process of learning and applying new technologies and techniques was highly rewarding. We were pleasantly surprised by how much we were able to achieve with the help of various online references and YouTube tutorials. While we cannot take full credit for the entire code, we invested time and effort in exploring and learning about every built-in function in OpenCV and other relevant libraries.

One of the most satisfying aspects of completing this project was the in-depth classification of our custom dataset and successfully training our facial recognition model. We were able to use this model to develop a biometric authentication system with our own data, and to create a GUI that tracks attendance based on the presence of each team member or user.