**PUNE INSTITUTE OF COMPUTER TECHNOLOGY**

**Department of Computer Engineering**

**(2022-2023)**

Subject - LP- V

Class: BE-1

Batch: R1

**Human Face Recognition**

Group members:

41146 – Sanket Kulkarni

41147 – Sourabh Kumbhar

41154 – Pranav Mohril

Guided by: Prof. Rutuja A. Kulkarni

**Table of Contents: -**

|  |  |  |
| --- | --- | --- |
| Sr.no | Title | Page no. |
| 1 | Problem Statement | 3 |
| 2 | Motivation | 3 |
| 3 | Scope | 3 |
| 4 | Objective | 3 |
| 5 | Outcomes | 4 |
| 6 | Software and Hardware  Requirements | 4 |
| 7 | Theory | 4 |
| 8 | Procedure | 5 |
| 9 | Code | 6 |
| 10 | Conclusion | 5 |

1. **Problem Statement:**

To develop a system that can accurately recognize an individual using their facial features extracted from an image.

1. **Motivation:**

The main objective of face recognition is to precisely determine and confirm the identity of a person by analyzing their facial characteristics. This has numerous practical uses, such as ensuring security and surveillance, managing access control, facilitating human-computer interaction, and providing personalized services. Utilizing face recognition technology provides a trustworthy and non-intrusive approach to identifying individuals, resulting in several advantages such as enhancing public safety, deterring fraudulent activities, improving user experiences, and minimizing the chances of identity theft. Additionally, the motivation behind face recognition involves the aspiration to create more advanced and intelligent systems capable of operating effectively and accurately in real-life situations.

1. **Problem Scope:**

Facial recognition, a discipline within computer vision and artificial intelligence, involves the identification and authentication of individuals by analyzing their facial characteristics. This technology finds extensive utility in various domains such as security, surveillance, access management, and interaction between humans and computers.

1. **Objectives:**

The objective of this report is to provide an overview of human face recognition, including its theory, hardware and software requirements, outcomes, and prospects. The report aims to explain how the technology works, what are its limitations and challenges, and how it can be applied to real-world problems.

1. **Outcomes:**

The expected outcomes of this report are:

1. An understanding of the fundamental principles of human face recognition.
2. An awareness of the hardware and software requirements for implementing a face recognition system.
3. An appreciation of the advantages and limitations of face recognition technology.
4. An insight into the ethical and social implications of using face recognition in different contexts.
5. **Hardware and Software Requirements**
6. **Hardware Requirements**

* CPU
* Windows 11, 64-bit
* GPU (Nvidia)
* 8GB RAM

1. **Software Requirements**

* Python 3.0
* PyCharm IDE
* Facebook’s DeepFace
* Histogram of Gradients

1. **Theory:**

Human face recognition is based on the idea that each person has unique facial features that can be used to distinguish them from others. These features can include the shape and size of the eyes, nose, mouth, and jawline, as well as the skin texture and color. Face recognition algorithms typically work in three main steps: face detection, face alignment, and face encoding.

1. **Face detection**

It is the process of identifying the presence and location of a face in an image or a video stream. This is typically done using machine learning models that can classify whether a given region of the image contains a face or not.

1. **Face alignment**

It is the process of normalizing the position and orientation of the detected face to a standard pose. This is necessary to ensure that the facial features are aligned and consistent across different images.

1. **Face encoding**

It is the process of extracting a set of numerical features that represent the unique characteristics of the face. This is typically done using deep learning models that can learn to encode the face features in a high-dimensional space.

1. **Procedure**
2. In this project, we use HOG or Histogram of Gradients to detect faces in an image.
3. We then use Open CV to preprocess the image and crop the image such that the new image only has the person’s face in it.
4. After cropping the face, we use Facebook’s DeepFace model to extract face embeddings from the cropped image.
5. We then compare these face embeddings to the ones in our database using cosine similarity and Euclidean distance.
6. Then we assume that the face with the highest cosine similarity is the person given in the input.
7. **Code**

Face detection function:

def detect\_faces(grp\_img):  
 face\_locations = face\_recognition.face\_locations(grp\_img)  
 images\_array = []  
 for face\_location in face\_locations:  
 top, right, bottom, left = face\_location  
 face\_array = grp\_img[top-100:bottom+100, left-100:right+100]  
 images\_array.append(face\_array)  
 return images\_array

Face recognition:

def find\_embeds(image\_arr):  
 embed\_arr = []  
 for i in range(len(image\_arr)):  
 result = DeepFace.represent(img\_path=image\_arr[i], model\_name="Dlib", enforce\_detection=False)  
 embed\_arr.append(result)  
 return embed\_arr

Cross-referencing faces:

def is\_same\_person(img1, img2):  
 em1 = np.array(img1)  
 em2 = np.array(img2)  
 a = np.matmul(np.transpose(em1), em2)  
 b = np.sum(np.multiply(em1, em1))  
 c = np.sum(np.multiply(em2, em2))  
 score = 1 - (a / (np.sqrt(b) \* np.sqrt(c)))  
 return score

1. **Conclusion**

The realm of human face identification is experiencing rapid advancements and holds numerous practical uses and noteworthy societal consequences. Despite considerable progress in recent times, there remain various obstacles and restrictions that require attention, including worries over privacy, algorithmic prejudice, and the ability to adapt to diverse lighting and positioning conditions. However, face recognition possesses the capability to profoundly transform numerous facets of our existence, enhancing our safety, security, and overall welfare.