**Mini Project Report on**



**Face Recognition using Machine Learning**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by-**

**Student Name: Chandni Mehta**  **University Roll No.: 2021676**

***Under the Mentorship of***

**Rishi Kumar**

**Assistant Professor, CSE**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Face Recognition using OpenCV”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Rishi Kumar**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: Chandni Mehta University Roll no: 2021676

**Table of Contents**

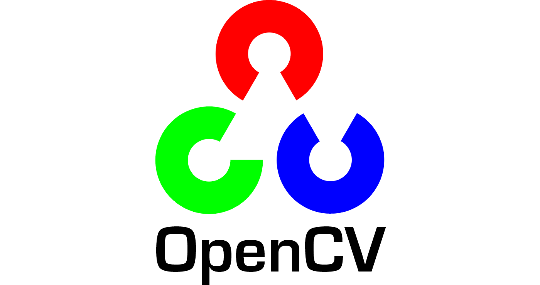
|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction  1.1.Introduction..................................................1.2.Problem Statement………………………..............  1.3.Objectives..................................................... | **1** |
| Chapter 2 | Literature Survey  2.1. Face Detection Techniques..........................  2.2. Face Recognition Models……………………..…..  2.3.Data Augmentation……..……………………..…….  2.4.Model…………………..……..……………………..……. | **2** |
| Chapter 3 | Methodology  3.1. Training In OpenCV......................................  3.2. Training The Classifiers................................  3.3. Code ............................................................  3.4. Real Time Face Recognition Implementation………………………………………………  3.5. LBPH………...........................................………. | **4** |
| Chapter 4 | Result and Discussion | **10** |
| Chapter 5 | Conclusion and Future Work  5.1. Future Scope.............................................. 5.2. Limitations.................................................  5.3. Conclusion.................................................. | **11** |
|  | References | **12** |

**Chapter 1**

**Introduction**

* 1. **Project Introduction**

Face recognition systems have become an increasingly important technology in a wide range of applications, from security and surveillance to user authentication and social media. These systems leverage advanced computer vision and machine learning algorithms to automatically identify and verify individuals based on their facial features.

OpenCV (Open Source Computer Vision Library) is a powerful and widely-used open-source library for computer vision and machine learning. It provides a comprehensive set of tools and algorithms for tasks such as image and video processing, object detection, and feature extraction. In the context of face recognition, OpenCV offers a variety of pre-built functions and models that make it easier to develop effective face detection and recognition systems.

* 1. **Problem Statement**

“Facial Recognition using open source Computer Vision Library (OpenCV)”. There are various scripts illustrated throughout the project that will have functionalities like detecting faces in static images, detecting faces in live feed using a webcam, capturing face images and storing them in the dataset, training of classifier for recognition and finally recognition of the trained faces.

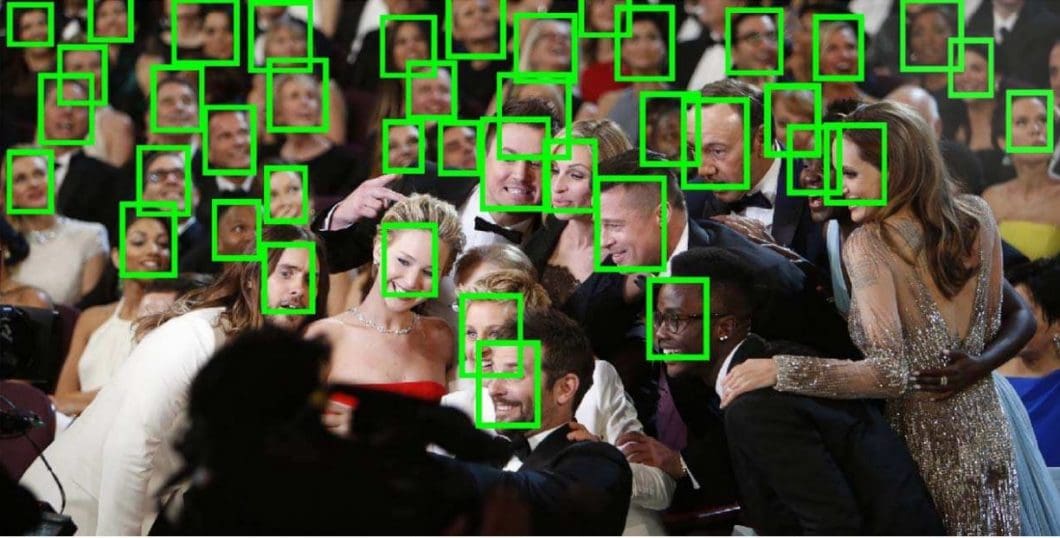
* 1. **Objectives**
* To capture and preprocess face images.
* To augment the dataset to improve model performance.
* To train a face recognition model.
* To implement a real-time face recognition system.

**Chapter 2**

**Literature Survey**

* 1. **Face Detection Techniques**

Face detection is the first step in face recognition. Various methods such as Haar Cascades, HOG+SVM, and deep learning-based methods like MTCNN and SSD have been explored. Haar Cascades, despite being older, remain popular due to their speed and simplicity.



Haar cascades: Haar Cascades are a classic approach to facial detection. They rely on a cascade of trained classifiers, each specialized in detecting specific features of a face, such as eyes, nose, and mouth.

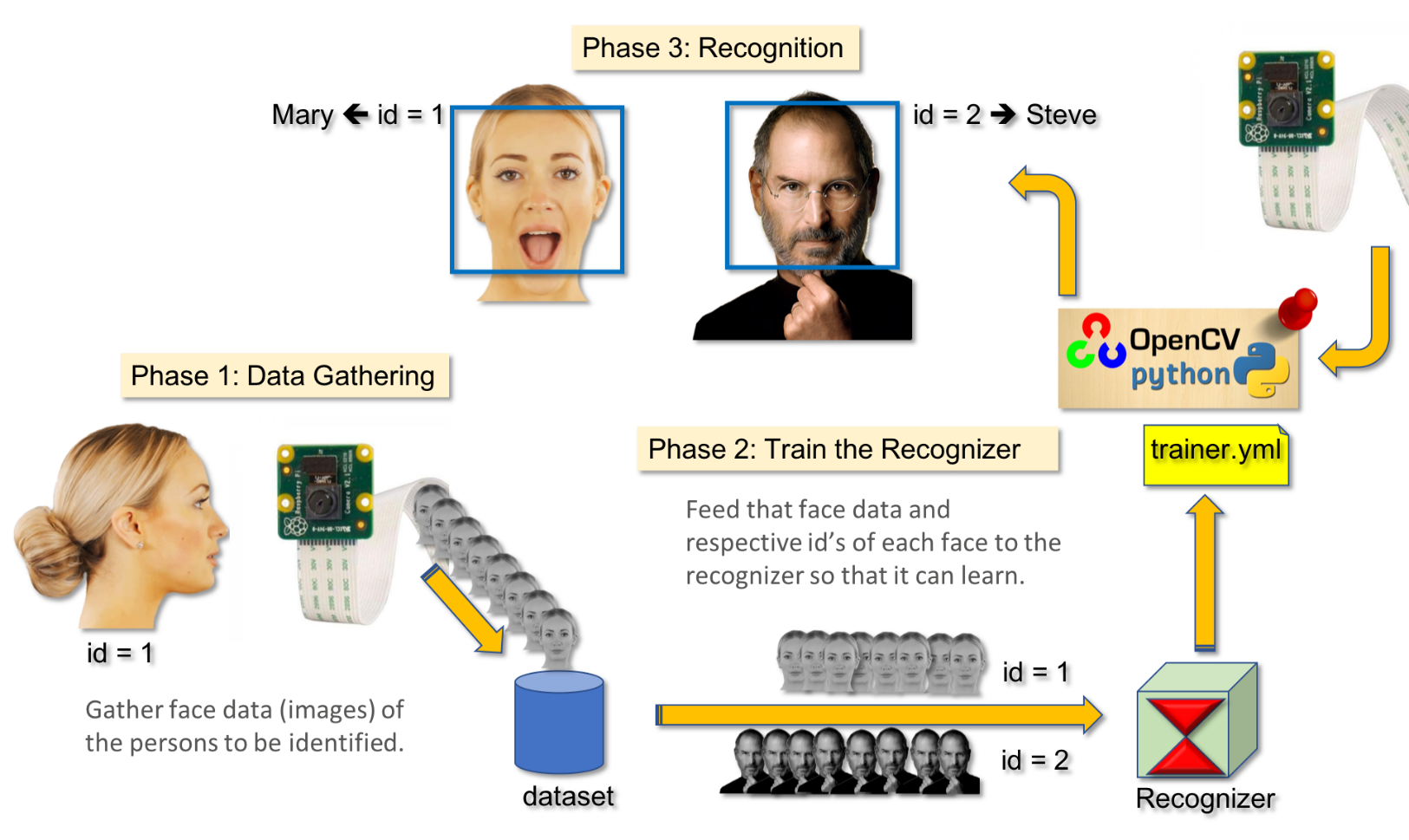
* 1. **Face Recognition Models**

Face recognition has evolved from traditional methods like Eigenfaces and Fisherfaces to modern approaches like Convolutional Neural Networks (CNNs). Models like VGG-Face, FaceNet, and Open Face leverage deep learning for high accuracy.

* 1. **Data Augmentation**

Data augmentation techniques such as rotation, scaling, and flipping are crucial for increasing the diversity of the training dataset, thus improving the robustness of the model.

* 1. **Model-**

Once a face has been detected, the next step is to recognize and identify the person. OpenCV provides several face recognition algorithms, each with its own strengths and weaknesses. The Eigenfaces algorithm, based on Principal Component Analysis (PCA), is a classic approach that works well for small, controlled datasets. The Fisherfaces algorithm, on the other hand, uses Linear Discriminant Analysis (LDA) to improve performance in more challenging scenarios with varying lighting and expressions. The more recent deep learning-based algorithms, such as FaceNet and DeepFace, have demonstrated superior accuracy and robustness, making them the preferred choice for modern face recognition systems.

**Chapter 3**

**Methodology**

**3.1 Training in OpenCV**

#### Data Collection

Gathering a diverse and representative dataset of facial images is crucial for training accurate face recognition models. This includes collecting images with varying lighting conditions, poses, expressions, and age groups.

#### Data Preprocessing

Preprocessing the facial images, such as aligning, cropping, and normalizing them, can significantly improve the performance of the face recognition model.

#### Model Selection

Choosing the appropriate face recognition algorithm and fine-tuning its hyperparameters can be a complex task, but it is essential for achieving the desired accuracy and performance.

#### Model Training and Evaluation

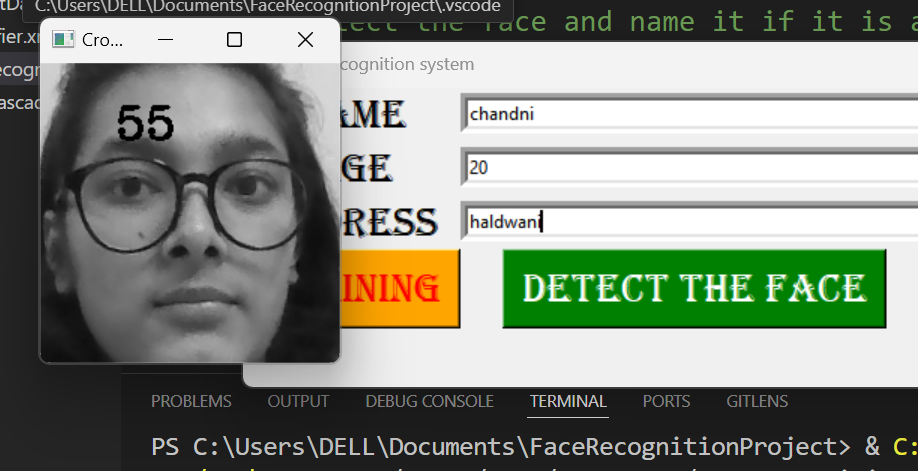
Properly training the face recognition model and evaluating its performance on a separate test set is crucial to ensure its reliability and generalization capabilities.

**3.2 Training the classifiers**

# Face Detection using Haar Cascade Classifiers

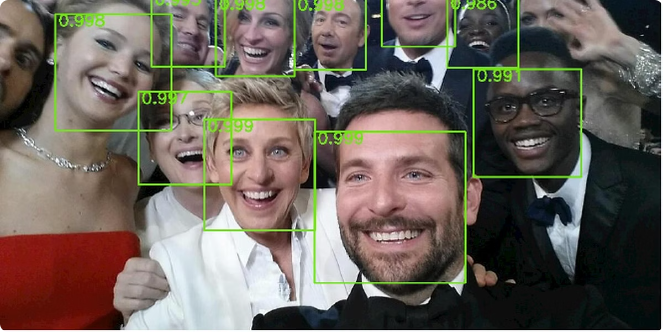
One of the fundamental steps in a face recognition system is accurately detecting the presence of faces within an image or video stream. The Haar Cascade Classifier, a machine learning-based approach developed by OpenCV, is a widely-used technique for face detection.

The Haar Cascade Classifier works by scanning an image for patterns that match pre-trained face features, such as the eyes, nose, and mouth. By analysing these features and their spatial relationships, the classifier can accurately identify the locations of faces within the input. This process is efficient and can be performed in real-time, making it a suitable choice for many face recognition applications.

**3.3 Code**

* **Generating Dataset:**
* **Train Classifier:**





* **Detecting Face:**
* **GUI**

**3.4 Real Time Face Recognition Implementation**

The final step in the face recognition pipeline is to implement a real-time system that can detect and recognize faces in live video streams or camera input. OpenCV's powerful video processing capabilities make this possible, allowing developers to build interactive and responsive face recognition applications.

The real-time implementation typically involves capturing video frames, detecting faces using the Haar Cascade Classifier, and then passing the detected faces through the trained LBPH model to identify the individuals. The recognized faces can then be displayed on the screen, along with additional information or actions based on the specific application requirements.

#### Video Capture

Continuously capture video frames from a camera or other video source.

#### Face Detection

Use the Haar Cascade Classifier to detect the presence of faces in each frame.

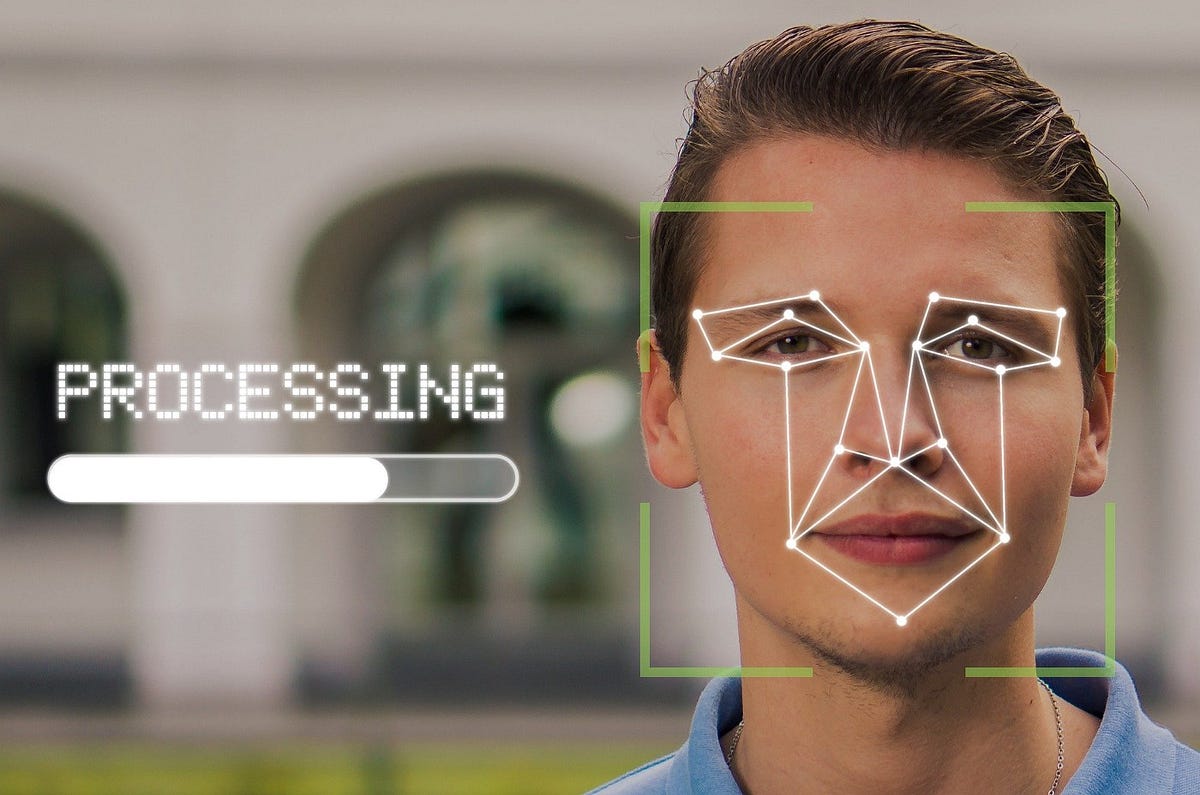
#### Face Recognition

Apply the trained LBPH model to the detected faces to identify the individuals.

**3.5 LBPH**

# Face Recognition using Local Binary Patterns Histograms (LBPH)

* Once the faces have been detected, the next step in the face recognition process is to identify and verify the individual. The Local Binary Patterns Histogram (LBPH) algorithm is a popular and effective technique for this task.
* LBPH works by analysing the local texture and patterns of a face image, capturing the unique facial features that distinguish one person from another. The algorithm creates a histogram representation of these patterns, which can then be used to compare and match faces in the system's database.
* LBPH is known for its robustness to variations in lighting, pose, and expression, making it a reliable choice for real-world face recognition applications.



**Chapter 4**

**Result and Discussion**

**4.1 Dataset Generation**

The dataset generation process successfully captured and stored images of faces. Each subject had approximately 200 images collected, ensuring a robust dataset for training.

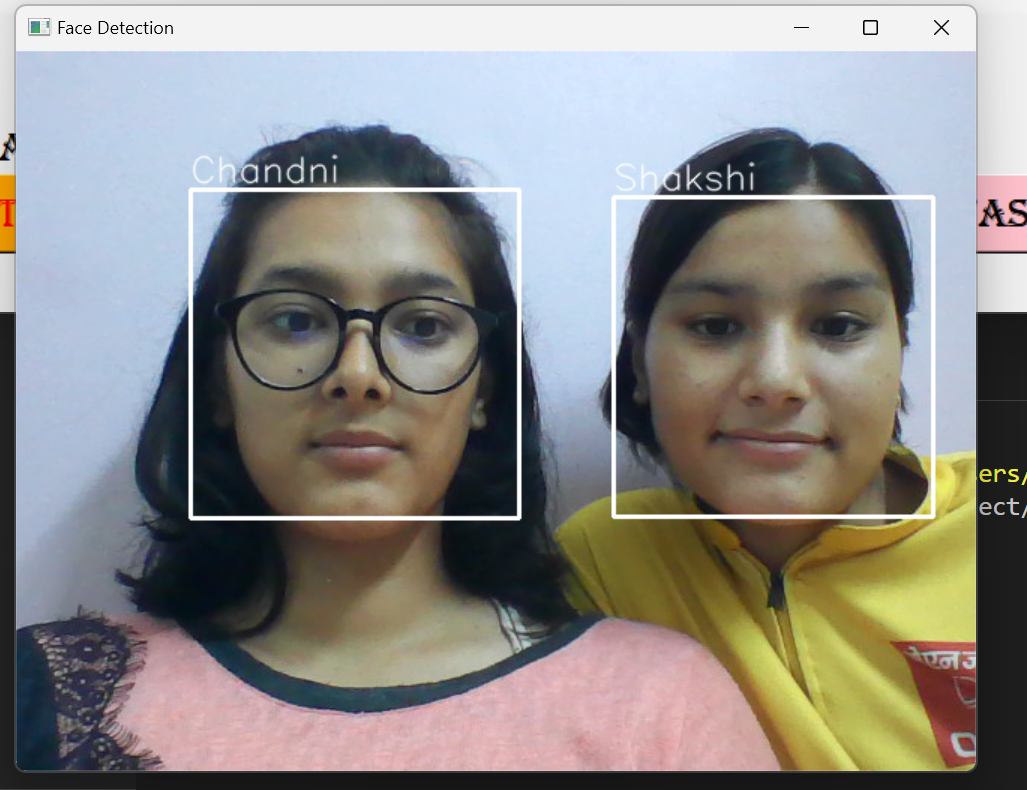
**4.2 Data Augmentation**

Data augmentation techniques effectively increased the diversity of the dataset, providing more variability in training samples and improving model performance.

**4.3 Model Training**

The LBPH face recognizer was trained successfully, with a label map saved for future reference. The model demonstrated high accuracy in distinguishing between different subjects.

**4.4 Real-Time Face Recognition**

**** he real-time face recognition system performed well in live tests, accurately recognizing faces with a confidence threshold set at 70%.

**Chapter 5**

**Conclusion and Future Work**

**5.1 Future Work**

Future enhancements could include:

* Utilizing more advanced face detection models like MTCNN or SSD for better detection accuracy.
* Implementing a CNN-based face recognition model for improved recognition accuracy.
* Expanding the dataset to include more subjects and varying conditions.
* Exploring transfer learning with pre-trained models for better performance.

**5.2 Limitation**

• Camera Angle: The relative angle of the target’s face with the camera impacts the recognition rate drastically. These conditions may not always be suitable, therefore creating a major drawback.

• Computational Power: The requirement of computational power also increases with increase in the size of the database. This becomes financially out of bounds for smaller organizations.

**5.3 Conclusion**

This project successfully implemented a comprehensive face recognition system, from data collection and augmentation to model training and real-time recognition. The system demonstrated high accuracy and real-time performance. This project report explains the implementation of face detection and face recognition using OpenCV with Python and also lays out the basic information that is needed to develop a face detection and face recognition software.

**References**

 Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.

 Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1), 71-86.

 Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. In BMVC.

 King, D. E. (2009). Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10(Jul), 1755-1758.