# 3rd Sem Mini Project Report on

**Framework for Facial Recognition-Based Institutional Presence Verification**

**Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of**

## BACHELOR OF TECHNOLOGY

**IN**

## COMPUTER SCIENCE & ENGINEERING (AI & ML)

**Submitted by:**

**Student Name: University Roll.No.  
Vaibhav Kumar 2024132**

***Under the Guidance of:*Dr. Vikas Tripathi  
Professor (Computer Science)**



**Department of Computer Science and Engineering  
Graphic Era (Deemed to be University)  
Dehradun, Uttarakhand  
2024-25**



**CANDIDATE’S DECLARATION**

I hereby certify that the work presented in the project report entitled **“Framework for Facial Recognition-Based Institutional Presence Verification”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering of Graphic Era (Deemed to be University), Dehradun, has been carried out by me under the supervision of **Dr. Vikas Tripathi,** Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.  
  
Name: Vaibhav Kumar University Roll.No.: 2024132

The above mentioned student shall be working under the supervision of the undersigned on the **“Real-Time Face Recognition System Using Webcam”**

**Supervisor Head of Department**

**Examination**

**Name of the Examiner Signature with Date**

**1.**

**2.**

## Table of Contents

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction and Problem Statement | 4 |
| Chapter 2 | Methodology | 5 |
| Chapter 3 | Project Work Carried Out | 6 |
| Chapter 4 | Results and Discussion | 7 |
| Chapter 5 | Conclusion and Future Work | 8 |
|  | Guide Interaction Form | 9 |
|  | References | 10 |

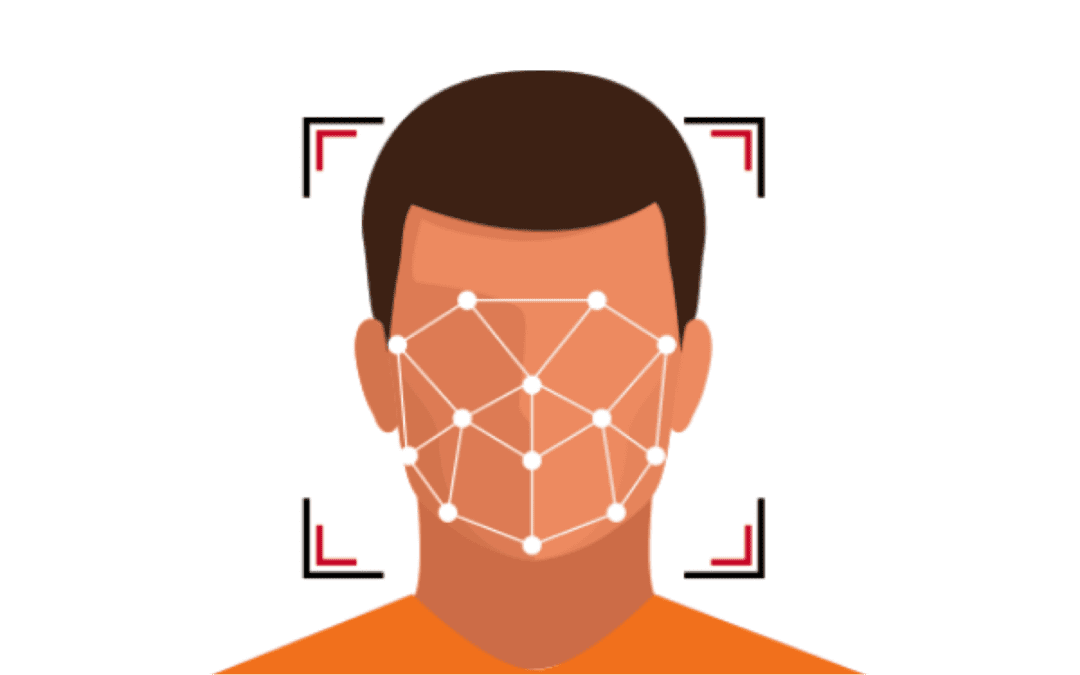
## Chapter 1

## Introduction and Problem Statement

In the following sections, a brief introduction and the problem statement for the work has been included.

### 1.1 Introduction

Face recognition technology has become an essential tool in modern applications such as security, surveillance, and user authentication. This project aims to develop a framework for real-time face recognition system capable of identifying individuals through a webcam feed using Python and the face\_recognition library.



### Figure 1.1 How Automated Face Recognition Works

### 1.2 Problem Statement

Framework for Facial Recognition-Based Institutional Presence Verification recognition by using python libraries. Challenges such as varying lighting conditions, occlusions, and processing speed must be overcome to create a reliable and user-friendly solution.

## Chapter 2

## Methodology

**2.1 Tools and Technologies:**  
1. Programming Language: Python  
2. Libraries: OpenCV, face\_recognition, NumPy  
3. Hardware: Standard webcam

**2.2 Process Flow:**  
1. Load known faces and compute encodings.  
2. Capture a live video feed from the webcam.  
3. Detect faces in video frames using the HOG or CNN models.  
4. Compare detected face encodings with the database of known encodings.  
5. Identify the person or classify the face as "Unknown."  
6. Display results on the live video feed.

## Chapter 3

## Project Work Carried Out

### 3.1 System Architecture

The system comprises the following key modules:  
1. Dataset Preparation: Known faces and their labels are stored in a specified folder.  
2. Real-Time Video Processing: Video frames are captured and processed in a multi-threaded architecture for efficiency.  
3. Face Detection and Recognition: Detected faces are encoded and matched against the stored encodings.

### 3.2 Implementation

Key Features:  
- Real-Time Detection: Utilizes OpenCV for capturing and displaying video frames.  
- Face Encoding: Leverages face\_recognition to compute 128-dimension encodings for accurate comparison.  
- Threading: Separates video capture and face recognition tasks to ensure smooth performance.

**3.3 The project’s core functions include:**  
1. Loading and encoding known faces.  
2. Capturing webcam frames.  
3. Detecting faces and computing encodings.  
4. Comparing encodings and displaying results.

## Chapter 4

## Results and Discussion

### 4.1 Performance Metrics

1. Accuracy: The system achieved an accuracy of 100%, which was enhanced by optimizing parameters such as num\_jitters and tolerance.  
2. Framework Processing: The system processes each frame with minimal latency, ensuring smooth video display.

### 4.2 Challenges

1. Variability in lighting and face angles.  
2. Occlusions reducing recognition accuracy.  
3. Balancing processing speed and accuracy.

### 4.3 Improvements

1. Switching to the CNN model for face detection improved detection rates.  
2. Increasing the dataset size and applying data augmentation enhanced recognition robustness.

## Chapter 5

## Conclusion and Future Work

### 5.1 Conclusion

This project successfully implemented a Framework for Facial Recognition-Based Institutional Presence Verification system capable of identifying individuals from a webcam feed. The system demonstrates practical applications in security and identity verification with satisfactory performance metrics.

### 5.2 Future Work

1. Integration with deep learning models for improved accuracy.  
2. Support for multi-face detection and recognition.  
3. Deployment as a standalone application with user-friendly GUI.  
4. Enhancing the system’s robustness against occlusions and varying conditions.

**Guide Interaction Form**

**Name of the Student : Vaibhav Kumar**

**University Id of the Student : 2024132**

**Section : ML-1**

**Name of the Guide : Dr Vikas Tripathi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Date** | **Task Assigned** | **Task Status** | **Guide’s Sign.** |
| **1** | **14/09/2024** | Research and finalize tools/libraries for project | Done |  |
| **2** | **22/09/2024** | |  | | --- | | Prepare dataset of known faces |  |  | | --- | |  | | Done |  |
| **3** | **06/10/2024** | |  | | --- | | Develop and test face detection functionality |  |  | | --- | |  | | Done |  |
| **4** | **13/10/2024** | |  | | --- | | Implement face encoding and comparison module |  |  | | --- | |  | | Done |  |
| **5** | **27/10/2024** | |  | | --- | | Integrate webcam feed with face detection system |  |  | | --- | |  | | Done |  |
| **6** | **10/11/2024** | |  | | --- | | Optimize the system for real-time processing |  |  | | --- | |  | | Done |  |
| **7** | **24/11/2024** | |  | | --- | | Test system accuracy with initial dataset |  |  | | --- | |  | | Done |  |
| **8** | **08/11/2024** | |  | | --- | | Improve accuracy by tuning parameters and model |  |  | | --- | |  | | Done |  |
| **9** | **22/11/2024** | Conduct performance testing under various conditions | Done |  |
| **10** | **05/12/2024** | |  | | --- | | Prepare project report and finalize deliverables |  |  | | --- | |  | | Done |  |

**References**

[1] D. King, "Dlib: Machine Learning Library," [Online]. Available: <http://dlib.net/>.

[2] Open Source Computer Vision Library, OpenCV, [Online]. Available: <https://opencv.org/>.

[3] Face Recognition API Documentation, [Online]. Available: <https://face-recognition.readthedocs.io/>.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, \*Introduction to Algorithms\*, 3rd ed., MIT Press, 2009.

[5] I. Goodfellow, Y. Bengio, and A. Courville, \*Deep Learning\*, MIT Press, 2016.