A SYNOPSIS ON

REAL-TIME FACE RECOGNITION SYSTEM WITH AI-POWERED ANALYSIS

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

BACHELOR OF COMPUTER APPLICATIONS

Submitted by:

Student Name : Kunal Kumar **University Roll No :** 2103577

Under the Guidance of
Mr. Priyansh Kumar
Assistant Professor

Department of Computer Applications
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
October-2025

CANDIDATE'S DECLARATION

I, **Kunal Kumar**, hereby certify that the work which is being presented in the Synopsis entitled **"Real-time Face Recognition System with AI-Powered Analysis"** in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Computer Applications** in the Department of Computer Applications of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **Mr. Priyansh Kumar**, [Assistant Proffesor], Department of Computer Applications, Graphic Era (Deemed to be University), Dehradun.

Name : Kunal Kumar. University Roll no : 2103577 signature :

The above mentioned students shall be working under the supervision of the undersigned on the **"Real-time Face Recognition System with AI-Powered Analysis"**

Signature
Supervisor

Signature
Head of the Department

Table of Contents

Chapter 1

Introduction and Problem Statement

1.1 Introduction

In the current era of artificial intelligence and smart computing, computer vision has emerged as a transformative technology, enabling machines to perceive, interpret, and interact with the visual world. Among its various sub-fields, face recognition stands out as one of the most significant and widely adopted biometric technologies due to its non-contact nature and high accuracy. The evolution of this technology from basic template matching algorithms to sophisticated deep learning models has opened doors for its application in numerous domains, including security, access control, human-computer interaction (HCI), and personalized services.

This project presents the design and development of an advanced, multi-featured, real-time face recognition system. It moves beyond the fundamental task of identification to create a holistic, interactive, and intelligent application. Built using Python and a suite of powerful libraries, the system operates on a live webcam feed to perform its core functions.

The foundation of the project is a robust face recognition module based on deep metric learning, which can dynamically enroll and identify multiple individuals with high precision. However, the primary innovation of this work lies in its integration of several additional AI-powered layers. The system incorporates the **DeepFace** library to perform real-time facial attribute analysis, allowing it to determine a person's **emotion**. This adds a layer of affective computing, enabling the system not just to know *who* a person is, but also to gain insight into their emotional state.

Furthermore, the project emphasizes a natural and interactive user experience. It integrates a bi-directional voice module, utilizing **Text-to-Speech (TTS)** with gTTS to deliver personalized greetings in Hindi, and **Speech-to-Text (STT)** with the SpeechRecognition library to understand basic user replies. This transforms the system from a passive observer into an active participant.

To ensure its practical utility, the system also features a persistent data management backend. All attendance records are logged into a structured **SQLite database**, allowing for reliable storage and retrieval of "last seen" information. This project, therefore, is a comprehensive amalgamation of real-time video processing, deep learning inference, voice synthesis, and database management, culminating in a prototype for a next-generation smart surveillance and interaction environment.

Figure 1.1: System Architecture - Real Time Face Recognition & Voice Interaction

1.2 Problem Statement

The development of this advanced system is motivated by the limitations of existing, conventional security and attendance systems. The problem statement for the present work is to address the following key challenges:

- **Lack of Integration:** Many systems perform a single function in isolation (e.g., only face recognition or only video recording). There is a need for a single, integrated platform that combines identification, attribute analysis, interaction, and logging into a cohesive workflow.

- **Performance Bottlenecks:** Running multiple AI models (for face detection, recognition, and emotion analysis) on a continuous, real-time video stream is computationally expensive. This presents a significant performance challenge, often leading to unacceptable lag and a poor user experience on standard hardware. The problem is to design an architecture that can handle these tasks concurrently without freezing the main application.

- **Passive User Experience:** Traditional systems are typically passive and non-interactive. They do not engage with the user, leading to an impersonal and often intimidating experience. The problem is to create a more natural and engaging human-computer interface by incorporating personalized voice greetings and interactions.

- **Inefficient Data Management:** Many simple applications log data to unstructured text or CSV files, which are not scalable, efficient, or easily queryable. The problem is to implement a structured, robust, and persistent data backend using a relational database (SQLite) to manage attendance records effectively.

- **Limited Contextual Awareness:** Standard recognition systems only answer "who" is present. They lack the ability to understand the context of the situation. By not analyzing attributes like emotion, they miss a significant layer of information. The problem is to add this layer of contextual awareness to make the system's output more meaningful.

Chapter 2

Background/ Literature Survey

Research in real-time object detection and recognition has been ongoing for decades. Early methods for face detection, such as the Viola-Jones algorithm, utilised Haar-like features and a cascade of classifiers to achieve real-time performance. For recognition, methods like Eigenfaces and Fisherfaces were popular.

However, with the rise of deep learning, Convolutional Neural Networks (CNNs) have become the state-of-the-art. Modern systems, such as those reviewed by Kanhere and Birchfied in [1], achieve high accuracy even in challenging conditions. This project leverages a modern approach based on deep metric learning. The face_recognition library used here is built upon dlib's deep learning model, which computes a 128-dimensional embedding (face encoding) for each face. Recognition is then performed by comparing the Euclidean distance between these embeddings.

For emotion analysis, this project utilises the DeepFace library, a comprehensive framework that wraps several state-of-the-art CNN models. The model for emotion classification has been pre-trained on large datasets of facial expressions, allowing it to classify emotions such as happy, sad, and neutral from a face image, as detailed in works like [2

Chapter 3

Objectives

The primary objectives of the proposed work are as follows:

- To design and develop a system for real-time face **detection** from a live webcam feed using the Histogram of Oriented Gradients (HOG) model.
- To implement a face **identification** module capable of recognizing multiple enrolled individuals by comparing their 128-dimensional face encodings.
- To integrate a pre-trained Convolutional Neural Network (CNN) via the DeepFace library for real-time facial **emotion classification**.
- To create a **bi-directional voice interaction** system using Text-to-Speech (gTTS) for personalized Hindi greetings and Speech-to-Text (SpeechRecognition) for understanding user replies.
- To design and implement a persistent **attendance logging system** using an SQLite database to store and retrieve user check-in times.
- To ensure a **smooth and responsive user experience** by optimizing the real-time processing loop with techniques like frame resizing and skipping.
- To develop a **file access control** feature that automatically unlocks a specific file or folder upon recognition of an authorized user.
- To build a **dynamic enrollment** system where new users can be added simply by placing their image in a folder without any code modification.

Chapter 4

Hardware and Software Requirements

4.1 Hardware Requirements

Sl. No	Name of the Hardware	Specification
1	Computer System	MacBook Air
2	Processor	Apple M1/M2 or Intel i5 (or above)
3	Memory (RAM)	8 GB or more
4	Webcam	Integrated Webcam
5	Microphone	Integrated Microphone

4.2 Software Requirements

Sl. No	Name of the Software	Specification
1	Operating System	macOS
2	Programming Language	Python 3.11
3	Environment	Anaconda
4	Code Editor	Visual Studio Code
5	Python Libraries	OpenCV, face_recognition, DeepFace, gTTS, SpeechRecognition, playsound, PyAudio, SQLite3

Chapter 5

Possible Approach/ Algorithms

The proposed system follows a modular approach, integrating several algorithms to achieve its objectives. The core workflow is centered around a real-time processing loop.

5.1 Face Encoding and Enrollment The system first enrolls known individuals by processing image files (e.g., kunal.jpg) from a known_faces directory.

- The face_recognition library loads each image.
- A pre-trained deep learning model computes a unique 128-dimensional vector (face encoding) for each face.
- These encodings, along with their names, are stored in memory for real-time comparison.

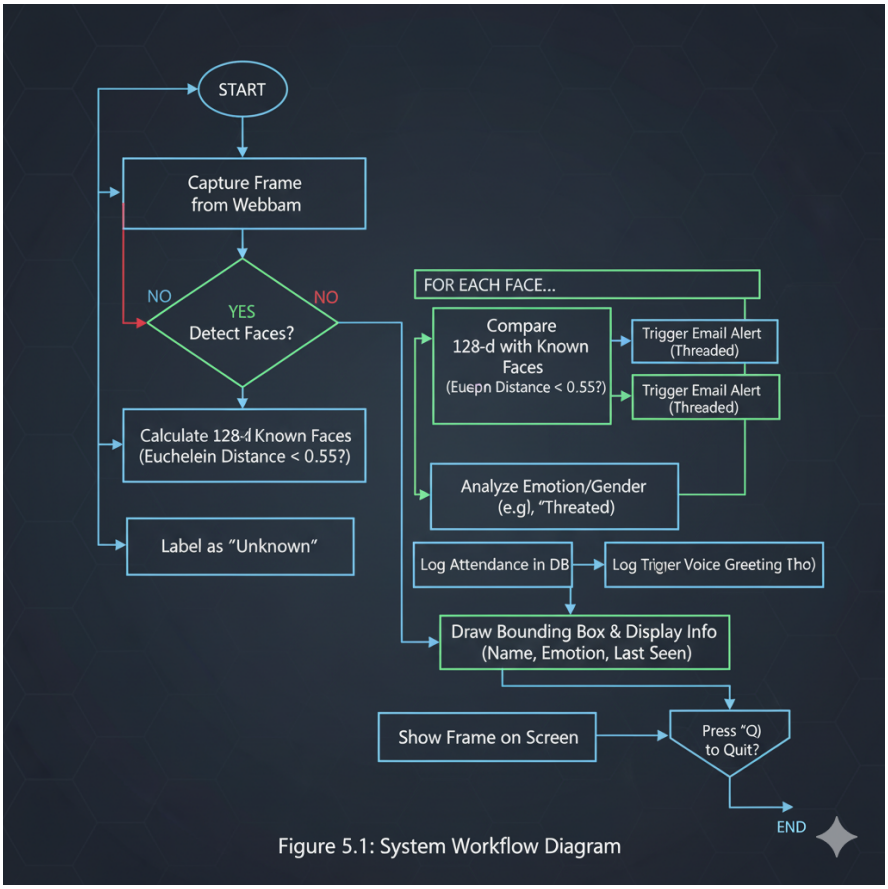

Figure 5.1: System Workflow Diagram

5.2 Real-time Detection and Recognition Loop The main loop continuously processes frames from the webcam. For each frame:

1. **Frame Capture & Preprocessing:** A frame is captured using OpenCV and resized for faster processing.

2. **Face Location:** The HOG (Histogram of Oriented Gradients) method is used to detect the bounding boxes of all faces.

3. **Encoding & Comparison:** A 128-d encoding is generated for each detected face and compared against the known encodings using Euclidean distance. A match is declared if the distance is below a set tolerance.

4. **Attribute Analysis:** The detected face's image area is passed to the DeepFace library, which uses a CNN to classify the dominant emotion.

5. **Voice Interaction:** For a recognized user, gTTS library sends text (e.g., "Namaste Kunal") to the Google Translate API, receives an MP3 audio stream, saves it, and plays it. The SpeechRecognition library then listens for a reply and uses the Google Web Speech API to convert it to text.

6. **Data Logging:** Attendance is recorded in an SQLite database. The system checks if an entry for the person exists for the current date. If not, it inserts a new record.

Figure 5.2 : Face Encoding Process

References

[1] N. K. Kanhere and S. T. Birchfied, “Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp.148-160, March 2008.

[2] K. Onoguchi, “Moving object detection using a cross correlation between a short accumulated histogram and a long accumulated histogram”, *Proc. 18th Int. Conf. on Pattern Recognition*, Hong Kong, August 20 - 24, 2006, vol. 4, pp. 896 – 899.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, “Introduction to Algorithms”, 2nd ed., The MIT Press, McGraw-Hill Book Company, 2001.

[6] A. Geitgey, "face_recognition Library" [Online]. Available: https://github.com/ageitgey/face_recognition. (Accessed: October 8, 2025).

[7] S. Serengil, "deepface Library" [Online]. Available: https://github.com/serengil/deepface. (Accessed: October 8, 2025).

[8] OpenCV (Open Source Computer Vision Library) [Online]. Available: https://opencv.org/. (Accessed: October 8, 2025).