Face Recognition System with Live Detection using OpenCV and FaceNet

BY

NANDANI KUMARI

(B. TECH CSE)

OBJECTIVE:

To develop a real-time face recognition system that detects, learns, and recognizes faces using **OpenCV** for detection and **FaceNet** for deep face embeddings.

TECHNOLOGIES USED:

- **Python**
- **OpenCV** – for face detection and webcam streaming
- **FaceNet (keras-facenet)** – for extracting 128-d facial embeddings
- **Scikit-learn** – for comparing similarity using cosine similarity
- **Tkinter** – for a simple GUI
- **NumPy & Pickle** – for array manipulation and saving data

PROJECT MODULES:

1. Face Detection:

- Used OpenCV's Haar Cascade to detect faces in real-time from webcam.

2. Face Embedding Extraction:

- FaceNet generates a 128-dimensional vector (face embedding) for each face image.

- These vectors uniquely represent each person.

3. Face Recognition:

- Compared live face embeddings with known ones using **cosine similarity**.
- Threshold > 0.6 determines a match; otherwise, labelled as "Unknown".

4. GUI Features:

- Capture and save face images to dataset
- Train model on new face embeddings
- Start real-time recognition
- Clean, easy-to-use interface via Tkinter

PROJECT STRUCTURE:

```
face_recognition_project/
 ├── dataset/           # Stores known face images
 ├── embeddings/        # Saves trained embedding data
 ├── utils.py           # Helper functions for embedding and detection
 ├── train_embeddings.py # Extracts embeddings from dataset
 ├── recognize.py       # Live face recognition
└── gui.py              # GUI application
```

HOW IT WORKS:

1. **Add Faces**: Use the GUI to capture multiple images of a person.
2. **Train Model**: Generate and save embeddings from captured faces.
3. **Recognize**: Run live webcam feed and identify known faces in real-time.

HOW TO RUN THE PROGRAM:

Step 1: Install Required Libraries

Before running the project, make sure the necessary Python libraries are installed:

pip install **OpenCV**-python keras-facenet scikit-learn **NumPy** pillow

Step **2**: Run the GUI Application

To start the project with a simple interface, run: **GUI.py**

This opens a GUI where you can see these option:

- Capture Face – Add new person’s images to the dataset
- Train Embeddings – Generate facial embeddings from the dataset
- Start Recognition – Begin live webcam-based face recognition

RESULTS:

- Real-time detection and recognition speed: ~20 FPS

- High accuracy for faces captured under decent lighting conditions
- Easily extendable by adding more face data

REFERENCES:

- FaceNet Research Paper
- keras-facenet GitHub
- OpenCV Documentation
- Scikit-learn cosine similarity