SAFE SIGHT

An AI-Powered Guardian for Visually Impaired Women's Home Security

INTRODUCTION

- The security and autonomy of the visually impaired female in her home are particularly difficult because home security appliances rely upon recognition and surveillance via vision. Indeed, the earlier systems would be of no use to a visually impaired person, and such a person cannot recognize visitors or judge their intentions. Such emotions make them vulnerable, anxious, and even less in control of their personal lives.

- Nowadays, AI and ML have advanced to the point where it is possible to create a customized solution to fill this gap. "Safe Sight" is an AI-based system designed to help bridge these gaps by facial recognition and emotion analysis. This project aims to provide audio-based feedback in real time to visually impaired women, thereby enabling them to identify visitors and their states of emotions for the most safe and independent living experience.

PROBLEM STATEMENT

- Blind women are highly compromised concerning safety inside their homes and thus lead to emotional vulnerabilities and psychological anxieties. She will not be able to determine who is coming to the house by visual inspection, and therefore, will make use of the auditory signal which may not be sufficient in identifying the intruders.

- Security systems available today are not produced based on their requirements, and thus, she does not have a proper means of knowing who is standing at her door or what that person's intentions are.

- End This would undermine their sense of autonomy and security in their living environments. "Safe Sight" aspires to fill this gap by developing an AI-powered facial recognition and emotion detection system that lets a woman suffering from a visual impairment identify known faces and recognize the emotional status of that particular face through real-time audio feedback,

OBJECTIVE

- To build a smart, AI-integrated security system specifically for visually impaired women.
- Provide real-time alerts through voice and messaging.
- Ensure real-time alerts, facial recognition, and easy accessibility.
- Promote independence, safety, and confidence through AI and IoT.
- Detect and recognize faces, emotions, and unusual activities.
- Empower independent living and reduce reliance on others.

ABSTRACT

- Safe Sight uses AI, IoT, and computer vision to deliver a smart, voice-driven home security experience.
- Key features: visitor recognition, real-time alerts, voice notifications, and emergency handling.
- Safe Sight combines AI, IoT, and computer vision technologies.
- Delivers a fully voice-controlled, smart home security system.
- Features include motion sensing, image capture, facial/emotion recognition, keypad-controlled entry, and alarm system.
- Sends alerts via Telegram and gives voice feedback to the user.

SYSTEM ARCHITECTURE

- Three Primary Modules:
  a. Motion Detection Module (Ultrasonic sensor, ESP8266)
  b. Visual Observation Module (ESP32-CAM, AI face detection)
  c. Access Control Module (Keypad, Arduino, Buzzer)
- Controlled centrally via Arduino & Wi-Fi-enabled ESP32.
- Telegram Bot as the notification medium.

METHODOLOGY

1.User Research and Empathy Mapping:

- She conducted interviews and surveys with conversations with the visually impaired women to gain an understanding of the special, particular safety concerns and special home security needs of women in this condition.

- She made empathy maps from the user's experience to show her pain points pertaining to the existing security systems and features that would enhance their safety and independence.

2. System Design:

- She defined the system architecture of having a high-definition camera with its features attached, including night vision and motion detection for continuous monitoring. Included an AI-based facial recognition module, which used machine learning algorithms to recognize known individuals from a pre-trained database.

- Included the emotion recognition module, which analyzed the real-time facial expressions and indicated the emotional state of the visiting crowd.

METHODOLOGY

3. Prototype Development:

- Developed the initial prototype with a friendly user interface which facilitated voice command interactions to enable easier use.
- Introduced an audio feedback mechanism that is speech-enabled to inform the users of the visitor's identities and emotional states to check within the shortest time possible.

4. User Testing and Collection of Feeding:

- Provided a hands-on testing experience to a small sample of female visually impaired to evaluate the prototype in scenarios.
- Qualified feedback collection by interviewing and application of a questionnaire in terms of user experience, correctness of recognition, and audio feedback.

5. Iterative Improvement:

- Analyzed the feedback to optimize the system with enhanced preciseness in the facial recognition and emotion detection algorithms.
- The user interface was fine-tuned to improve access; voice navigation and configuration were made even easier to use.

METHODOLOGY

6. Performance Evaluation:

- Measurement of key performance metrics in recognition accuracy, response time, and user satisfaction through validation of the system's effectiveness.
- Data privacy was implemented in the form of secure storage and manipulation of user data.

7. Review and Future Extensions:

- User review was finally conducted to ensure all improvements agreed with customers' needs and expectations.
- Further extensions included planning for further testing and integration into mobile apps for remote notifications and settings adjustment.

MODULE 1 - MOTION DETECTION AND ALERT

- Ultrasonic Sensor detects movement within 2–4 meters.
- ESP8266 sends notification to Telegram bot.
- Real-time alert system for nearby movement at entry points.
- Ensures users are notified even before someone reaches the door.
- Great for intruder detection and early awareness.

MODULE 2 - VISUAL OBSERVATION AND IMAGE CAPTURE

- ESP32-CAM captures images on user request or motion detection.
- Telegram bot delivers images instantly.
- AI model recognizes faces and detects unfamiliar visitors.
- Images are processed using Haarcascade AI model.
- Additional emotion detection model evaluates visitor’s expressions.

MODULE 3 - ALARM AND ACCESS CONTROL

- Arduino-based numeric keypad lock.
- Adds security and controls access to the home.
- A numeric keypad is attached to the door.
- Only the correct PIN can deactivate the alarm.
- Wrong PIN triggers a buzzer, alerts are sent instantly.
- Deters unauthorized access and enhances security.

ACCESSIBILITY FEATURES

System supports Natural Language Processing (NLP).

Voice commands like:

“Who is at the door?”

“Lock the door.”

Provides vibration and audio responses.

Reduces dependency on mobile or visual interfaces.

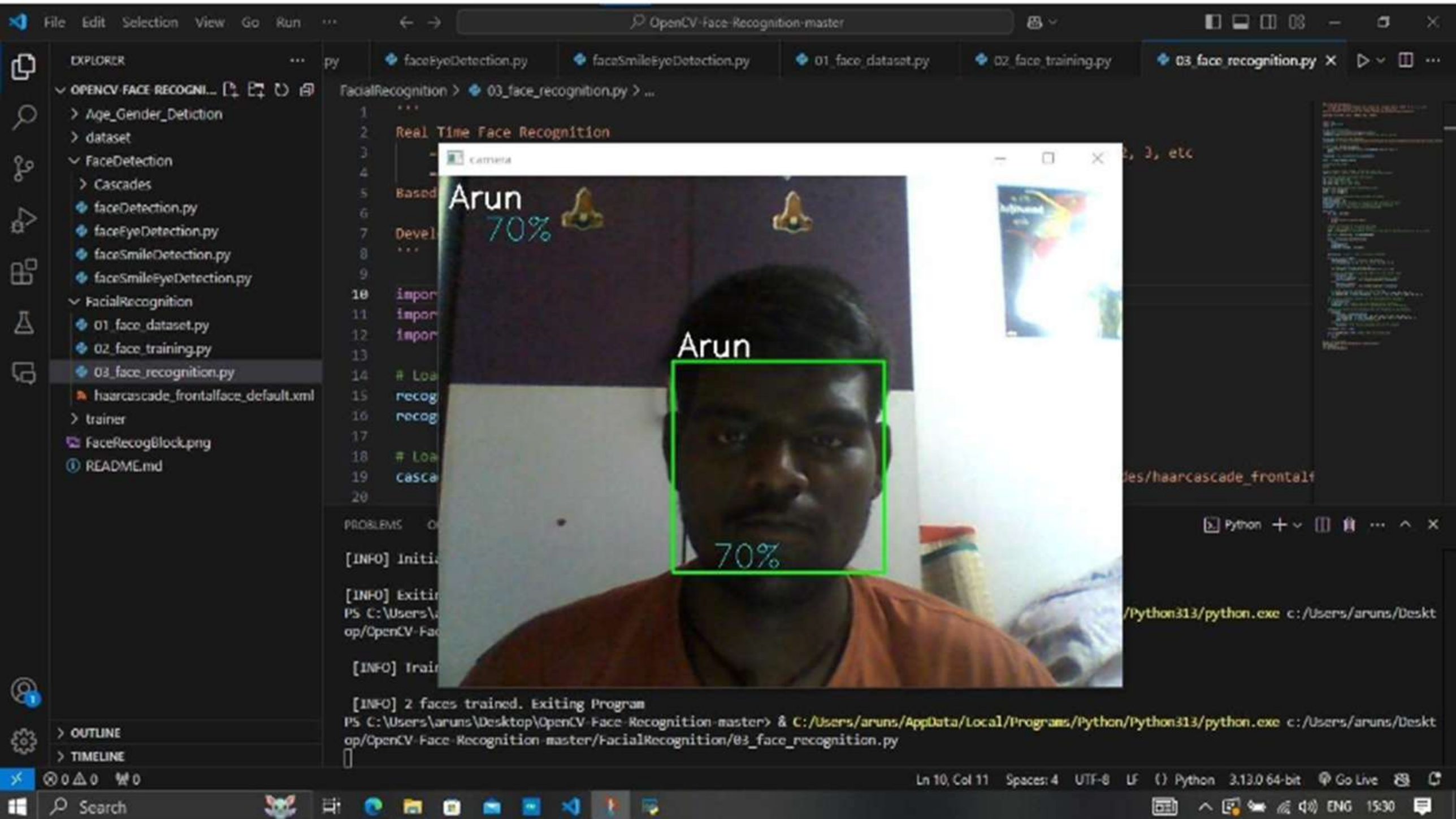
Vibration and audio feedback for alerts.

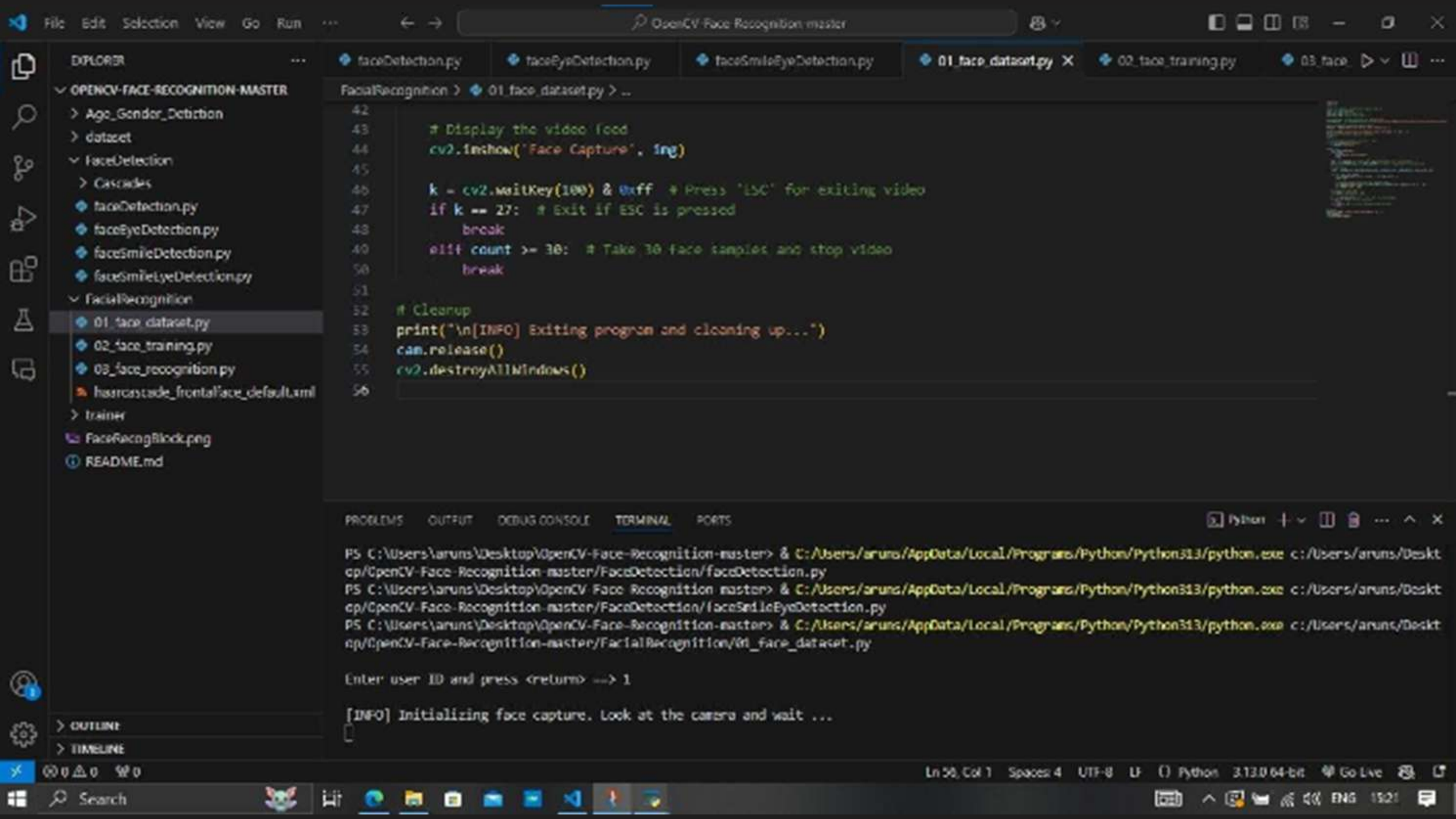
AI FEATURES

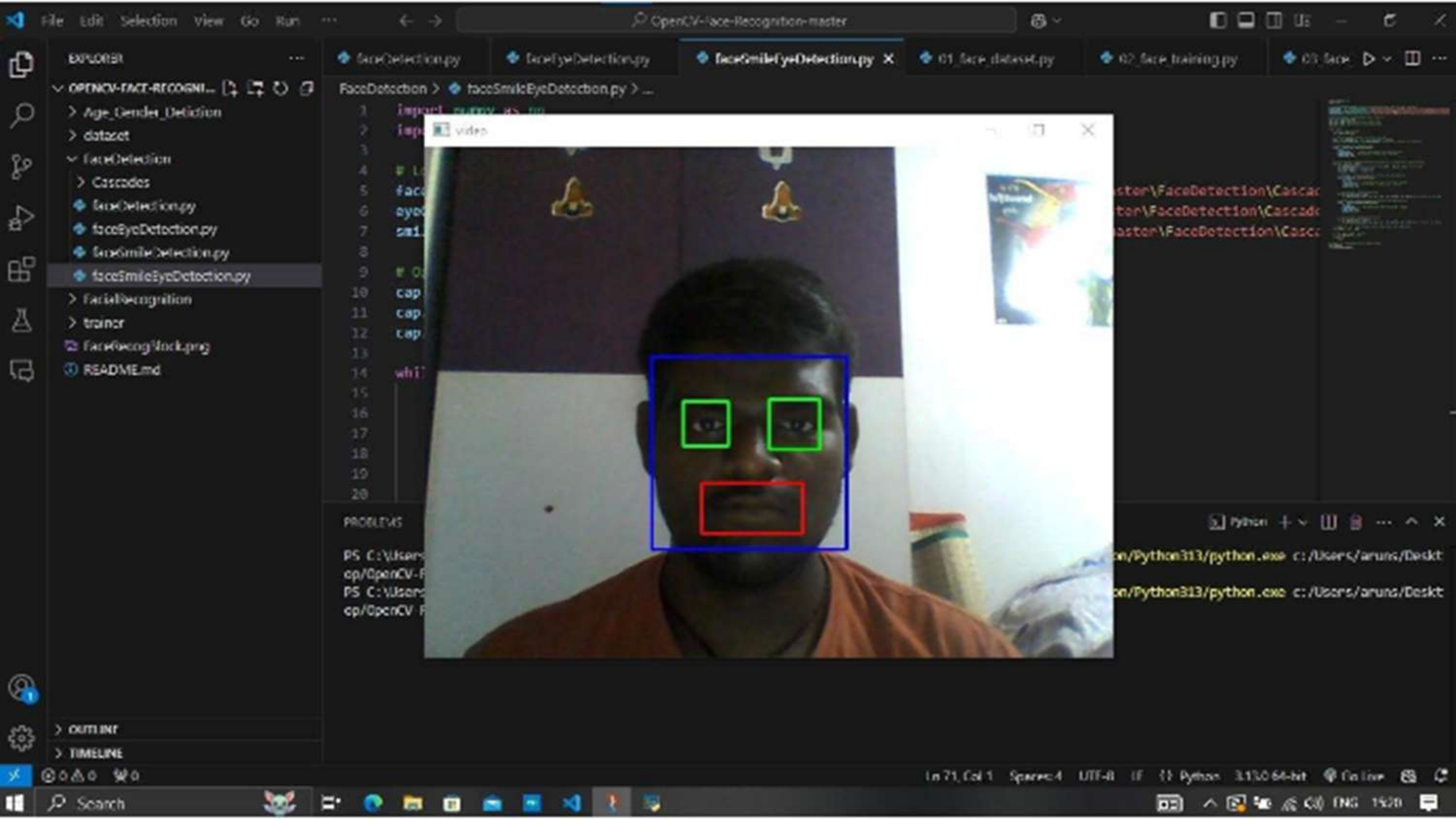
- Face Detection Accuracy: ~90%
- Emotion Recognition Accuracy: 80–85%
- Trained on custom datasets.
- Faces are logged and stored securely.
- Emotion detection helps interpret mood of visitor.

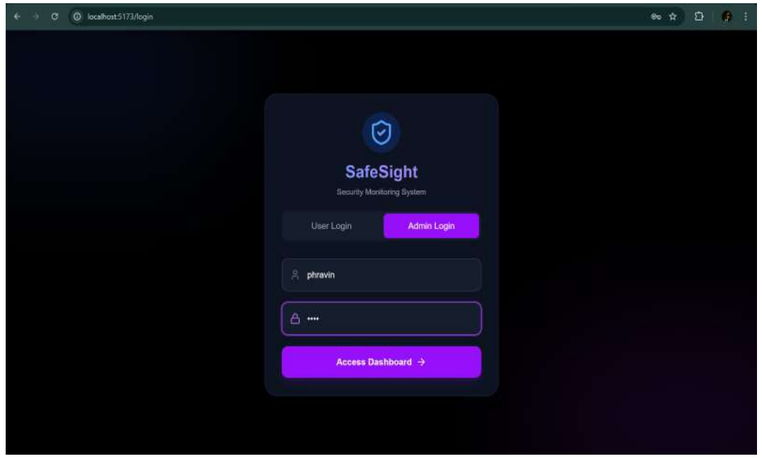
Implementation Methodology

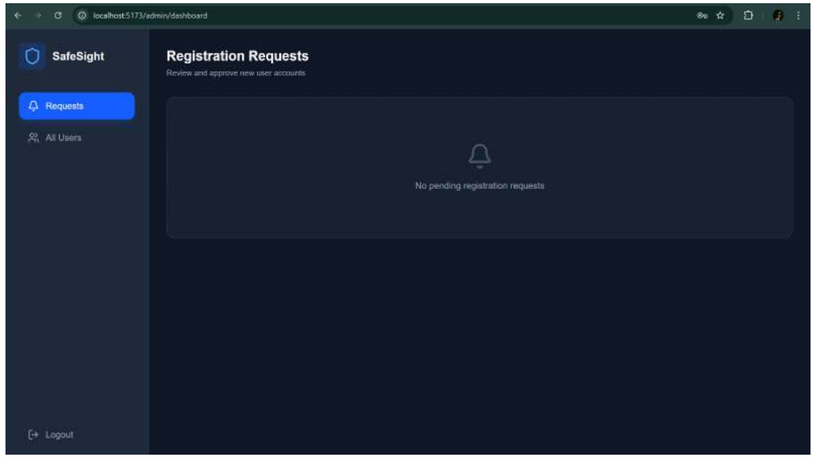
- Modules integrated using Arduino IDE.
- ESP modules configured to send data to Telegram bot.
- Haarcascade and pre-trained emotion models used in Python.
- Power-efficient and works on low-cost hardware.
- System is modular and scalable.

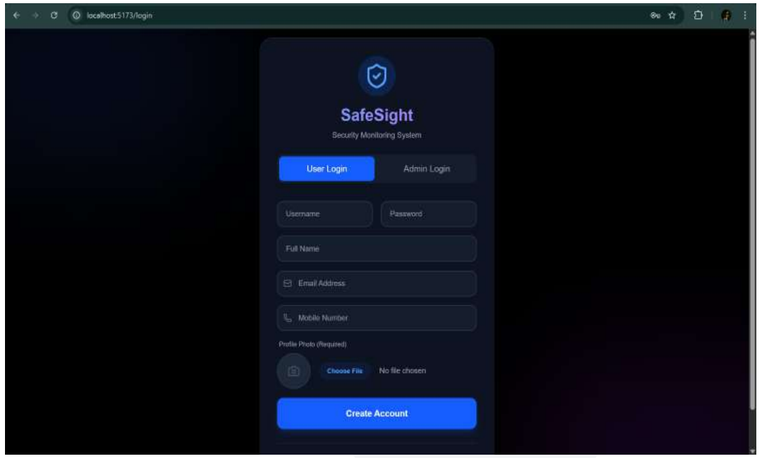

File Edit Selection View Go Run
OpenCV-Face-Recognition-master
EXPLORER
faceEyeDetection.py
faceSmileEyeDetection.py
01_face_dataset.py
02_face_training.py
03_face_recognition.py
OPENCV-FACE-RECOGNI...
Age_Gender_Detection
dataset
FaceDetection
Cascades
faceDetection.py
faceEyeDetection.py
faceSmileDetection.py
faceSmileEyeDetection.py
FacialRecognition
01_face_dataset.py
02_face_training.py
03_face_recognition.py
haarcascade_frontalface_default.xml
trainer
FaceRecogBlock.png
README.md
FacialRecognition > 03_face_recognition.py > ...
Real Time Face Recognition
camera
Arun
70%
Arun
70%
PROBLEMS
[INFO] Initi
[INFO] Exiti
PS C:\Users\
op/OpenCV-Fac
[INFO] Trai
[INFO] 2 faces trained. Exiting Program
PS C:\Users\aruns\Desktop\OpenCV-Face-Recognition-master> & C:/Users/aruns/AppData/Local/Programs/Python/Python313/python.exe c:/Users/aruns/Desktop/OpenCV-Face-Recognition-master/FacialRecognition/03_face_recognition.py
OUTLINE
TIMELINE
Ln 10, Col 11
Spaces: 4
UTF-8
LF
Python
3.13.0 64-bit
Go Live
Search
ENG
15:30

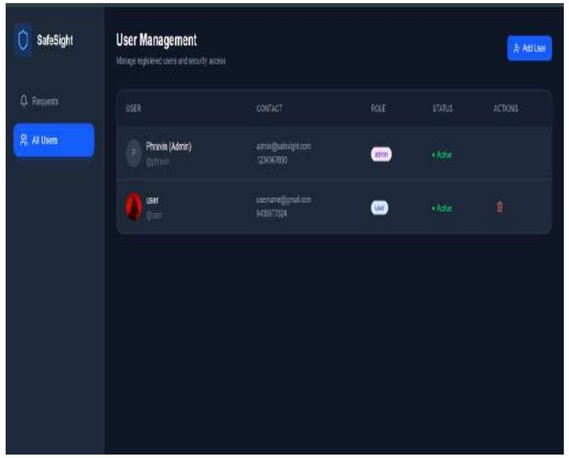

File Edit Selection View Go Run
OpenCV Face Recognition master
EXPLORER
OPENCV-FACE-RECOGNITION-MASTER
Age_Gender_Detection
dataset
FaceDetection
Cascades
faceDetection.py
faceEyeDetection.py
faceSmileDetection.py
faceSmileEyeDetection.py
FacialRecognition
01_face_dataset.py
02_face_training.py
03_face_recognition.py
haarcascade_frontalface_default.xml
trainer
FaceRecogBlock.png
README.md
faceDetection.py
faceEyeDetection.py
faceSmileEyeDetection.py
01_face_dataset.py
02_face_training.py
03_face_
FacialRecognition > 01_face_dataset.py > ...
42
43 # Display the video feed
44 cv2.imshow('Face Capture', img)
45
46 k = cv2.waitKey(100) & 0xff # Press 'ESC' for exiting video
47 if k == 27: # Exit if ESC is pressed
48 break
49 elif count >= 30: # Take 30 face samples and stop video
50 break
51
52 # Cleanup
53 print("\n[INFO] Exiting program and cleaning up...")
54 cam.release()
55 cv2.destroyAllWindows()
56
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
PS C:\Users\aruns\Desktop\OpenCV-Face-Recognition-master> & C:/Users/aruns/AppData/Local/Programs/Python/Python313/python.exe c:/Users/aruns/Desktop/OpenCV-Face-Recognition-master/FaceDetection/faceDetection.py
PS C:\Users\aruns\Desktop\OpenCV-Face-Recognition-master> & C:/Users/aruns/AppData/Local/Programs/Python/Python313/python.exe c:/Users/aruns/Desktop/OpenCV-Face-Recognition-master/FaceDetection/faceSmileEyeDetection.py
PS C:\Users\aruns\Desktop\OpenCV-Face-Recognition-master> & C:/Users/aruns/AppData/Local/Programs/Python/Python313/python.exe c:/Users/aruns/Desktop/OpenCV-Face-Recognition-master/FacialRecognition/01_face_dataset.py
Enter user ID and press <return> --> 1
[INFO] Initializing face capture. Look at the camera and wait ...
OUTLINE
TIMELINE
Ln 56, Col 1 Spaces: 4 UTF-8 LF Python 3.13.0 64-bit Go Live
Search
ENG 1521

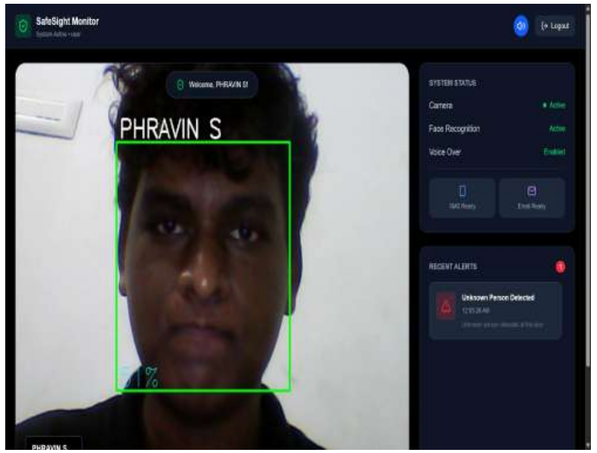

File Edit Selection View Go Run
OpenCV-Face-Recognition-master
EXPLORER
OPENCV-FACE-RECOGNI...
Age_Gender_Detection
dataset
FaceDetection
Cascades
faceDetection.py
faceEyeDetection.py
faceSmileDetection.py
faceSmileEyeDetection.py
FacialRecognition
trainer
FaceRecogMock.png
README.md
faceDetection.py
faceEyeDetection.py
faceSmileEyeDetection.py
01_face_dataset.py
02_face_training.py
FaceDetection > faceSmileEyeDetection.py > ...
video
PROBLEMS
OUTLINE
TIMELINE
Search

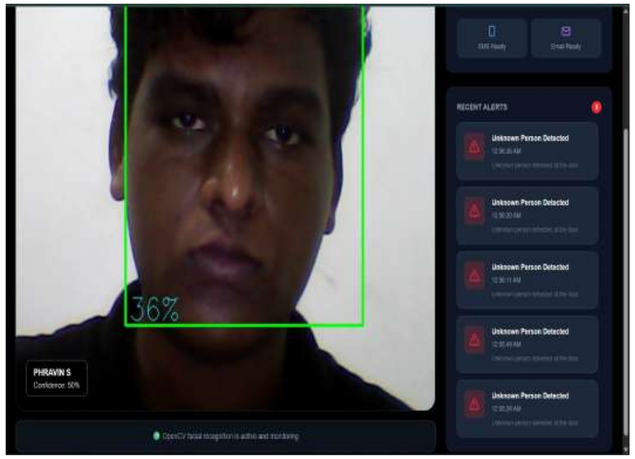

localhost:5173/login
SafeSight
Security Monitoring System
User Login
Admin Login
phravin
Access Dashboard →


localhost:5173/admin/dashboard
SafeSight
Requests
All Users
Logout
Registration Requests
Review and approve new user accounts
No pending registration requests


localhost:5173/login
SafeSight
Security Monitoring System
User Login
Admin Login
Username
Password
Full Name
Email Address
Mobile Number
Profile Photo (Required)
Choose File
No file chosen
Create Account


SafeSight
User Management
Manage registered users and security access
Add User
Requests
All Users
USER
CONTACT
ROLE
STATUS
ACTIONS
Phravin (Admin)
admin@safesight.com
1234567890
admin
Active
user
username@gmail.com
9405977324
user
Active


SafeSight Monitor
Logout
Welcome, PHRAVIN S!
PHRAVIN S
SYSTEM STATUS
Camera
Active
Face Recognition
Active
Voice Over
Enabled
SMS Ready
Email Ready
RECENT ALERTS
Unknown Person Detected
12:01:26 AM


36%
PHRAVIN S
Confidence: 50%
SMS Ready
Email Ready
RECENT ALERTS
Unknown Person Detected
12:36:35 AM
Unknown Person Detected
12:36:20 AM
Unknown Person Detected
12:36:11 AM
Unknown Person Detected
12:35:49 AM
Unknown Person Detected
12:35:39 AM

PROJECT OBJECTIVES

Promote Independent Living through AI & Voice Assistance

Reduce dependency by allowing users to interact using voice commands and receive auditory alerts.

Detect, Identify, and Alert Instantly

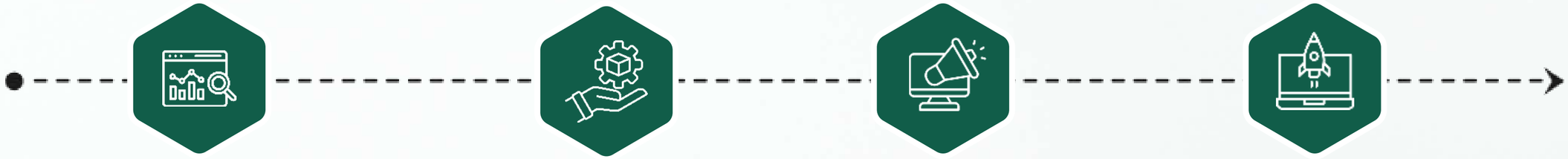
Use computer vision to recognize faces and detect emotions or unfamiliar individuals at the door.

Create an Inclusive, Accessible Smart Home Experience

Bridge the gap between modern security and accessibility needs for the blind and visually impaired.

SCOPE OF WORK

Study the safety needs and daily challenges of visually impaired women through surveys, interviews, and case studies.



Smart System Development

- Design and implement the Safe Sight system integrating ESP32-CAM, Arduino, motion sensors, AI-based face/emotion detection, and Telegram bot alerts.

Accessibility Optimization

- Incorporate voice control (NLP), vibration, and audio feedback to ensure complete usability for visually impaired users.

Testing & Validation

- Evaluate system performance across lighting conditions, user environments, and hardware configurations; ensure reliability and real-time responsiveness.

Awareness and Deployment Plan

- Create user manuals, demos, and awareness content; plan for pilot installation and community outreach programs to introduce Safe Sight to real users.

ALGORITHM USED

Category	Algorithm
Face Detection	Haarcascade Classifier (OpenCV)
Face Recognition	LBPH Algorithm
Emotion Recognition	CNN with FER-2013 Dataset
Voice Command Handling	STT + Intent Matching (NLP)
Notifications	Telegram Bot API
Motion Detection	Threshold-based Echo Return Algorithm

PROJECT ACCURACY

Face Recognition Accuracy – 90%

- The AI model correctly identifies authorized individuals with high precision under good lighting.
- Ensures the user is notified accurately about visitors.

Motion Detection Accuracy – 98%

- Ultrasonic sensor detects movement within a 2–4 meter range with near-perfect consistency.
- Enables proactive alerts before the visitor interacts.

Emotion Detection Accuracy – 85%

The system can classify basic emotions like happy, angry, neutral, etc., aiding in judgment of visitor intent.

Voice Command Recognition Accuracy – 92%

- Supports natural language commands like “Who is at the door?” or “Lock the door.”
- Ensures smooth and accessible interaction for visually impaired users.

HARDWARE COMPONENTS

- ESP32-CAM
- Ultrasonic Sensor (HC-SR04)
- ESP8266 Wi-Fi Module
- Arduino UNO
- Keypad Module
- FTDI Programmer
- Buzzer

LIMITATIONS AND FUTURE WORK

Current Limitations:

- Dependent on stable Wi-Fi connection
- Poor lighting affects face recognition
- Telegram dependency for notifications

Future Plans:

- Use edge computing for offline decision-making
- Integrate IR lighting for night vision
- Add voice assistants like Alexa/Google Home
- Expand support for multi-room integration

BIG THANKS

- Safe Sight is more than a security system; it is a compassionate companion.
- Brings together AI, IoT, and inclusivity for a safer tomorrow.
- Thank You! (Include contact details or project QR code)

Date: July 25, 2025