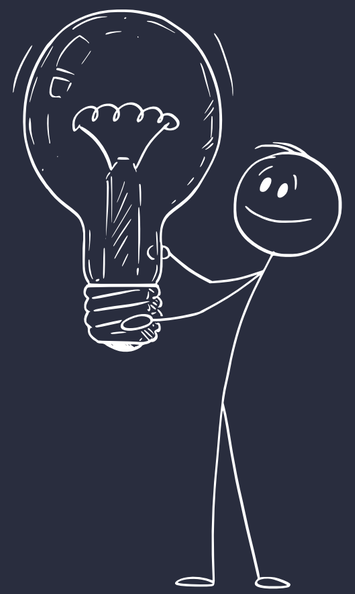
Introduction and Motivation

Aim - A face recognition system that marks the attendance of individual by recognising face.

- Accuracy and Reliability
- Time Efficiency
- Contactless Operation
- Scalability and Automation
- Technological Learning



OBJECTIVES

- **High Accuracy and Reliability**

To accurately identify and verify registered individuals in real-time, minimizing both false acceptances (marking the wrong person) and false rejections (failing to identify a correct person), even under varied lighting and pose conditions.

- **Automation and Efficiency**

To completely automate the attendance-taking process, eliminating manual entry and significantly reducing the time and labor required from administrators or instructors.

- **Secure Data Management**

To create a secure and centralized database that records attendance data, timestamps each entry, and allows for easy generation of reports, analysis, and integration with existing management systems (like a student information system or employee payroll).

Project Roadmap

5) Save Attendance

Give a path of csv file to save the attendance to that particular time

4) Link Model to Database

Link that face recognition model to SQL database to store the user details

3) Model to recognize face

Build a face recognition model using Haar Cascade to detect face and Linear Binary Pattern to Recognize face.

2) Database

Created SQL Database to store the user’s details

1) User Interface

Create User Interface to enter the details of the user for the database.

PROJECT WORKFLOW

Student Detail

Firstly store the details of student to maintain a database and to identify the name and roll number.

Take Photo Sample

Take the live photo sample whether previously stored or not.

Train Dataset

Train the dataset using the live captured photos or the already stored photos

Face Detection

Recognize the face and identify the person.

Update Attendance

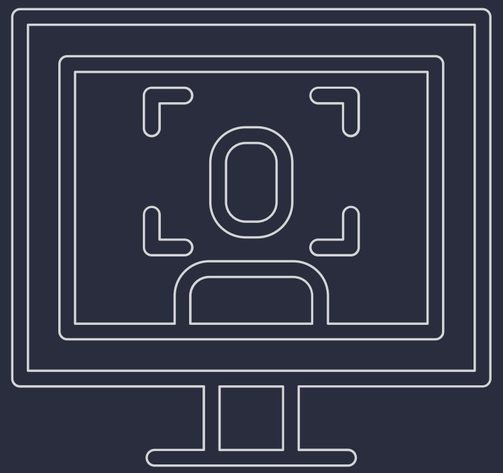
Store the attendance in csv file with date and time.

Goal: Detect faces/eyes using OpenCV’s cv

Haar Cascade Classifier

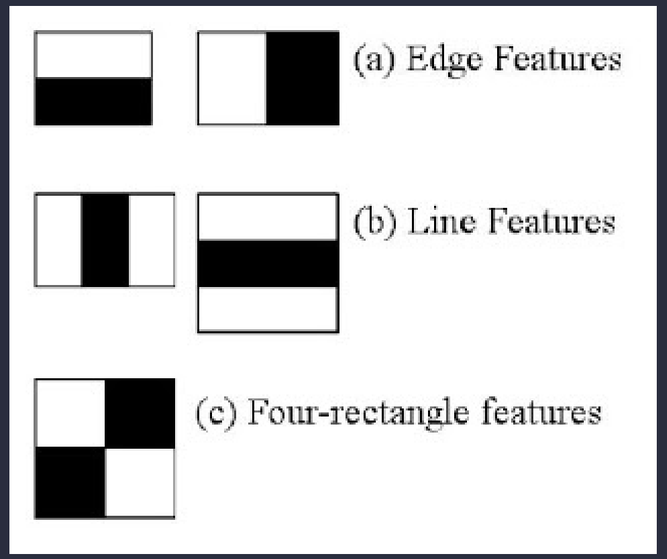
How Haar Cascade Works:

- Uses Haar features (black–white rectangle differences).
- Integral Image makes feature computation fast.
- AdaBoost selects the most useful features.
- Cascade of Classifiers quickly rejects non-face regions.
- Result: Fast, real-time face detection.



OpenCV Workflow:

- Load classifier .xml using CascadeClassifier::load.
- Convert frame to grayscale + equalize.
- Detect faces/eyes using detectMultiScale.
- Draw bounding shapes and display output.



Result:

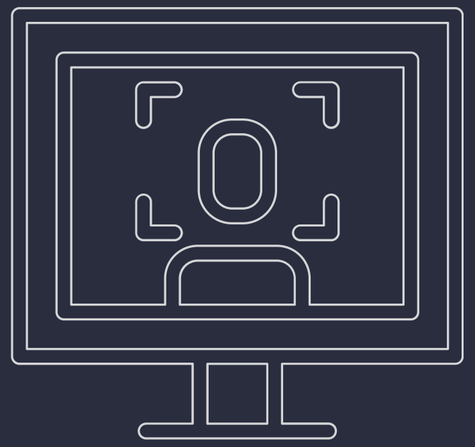
- Haar → accurate, detect faces.

Goal: Local Binary Patterns Histograms (LBPH)–based face recognition

LBPH Face Recognizer

Key Parameters :

- radius: size of circular LBP neighborhood
- neighbors: number of sampling points (usually 8)
- grid_x, grid_y: number of histogram cells (spatial grid)
- threshold: max allowed distance; above it → returns -1



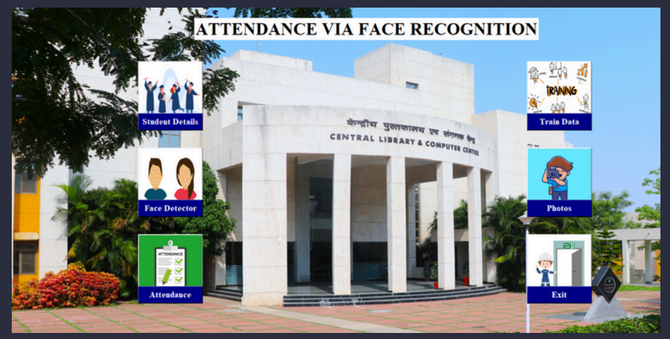
Notes:

- Works on grayscale images
- Supports model updating
- Stores histograms + labels internally

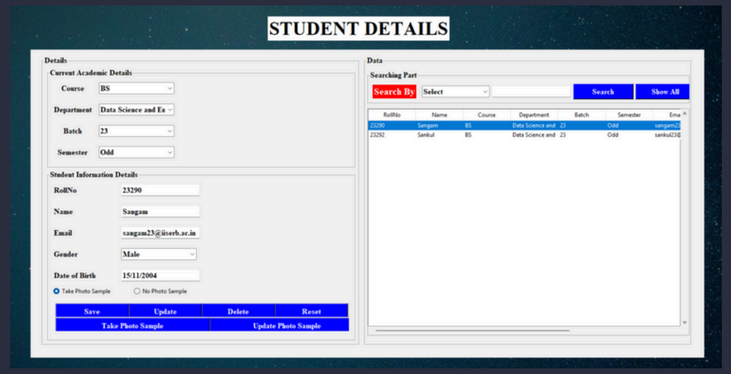
Main Functions:

- create() – initialize LBPH model with custom settings
- getRadius(), getNeighbors(), getGridX(), getGridY(), getThreshold()
- setRadius(), setNeighbors(), setGridX(), setGridY(), setThreshold()
- getHistograms() – returns LBP histograms
- getLabels() – training labels

OUR MODEL DESIGN



1



2

6

	A	B	C	D	E	F
1	RollNo	Name	Time	Date	Attendance	
2	23290	Sangam	09:33:51	15/11/2025	Present	
3	23292	Sankul	09:33:55	15/11/2025	Present	
4						
5						

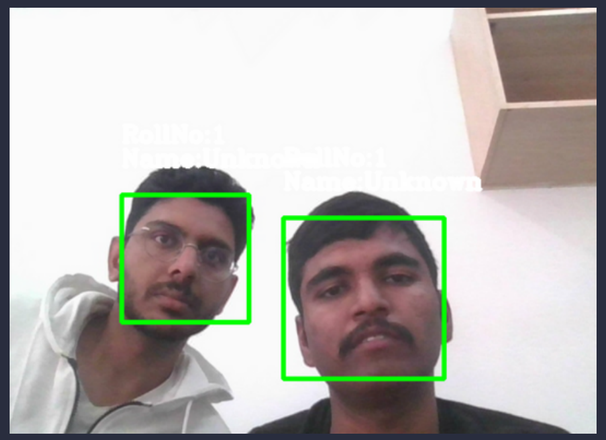
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

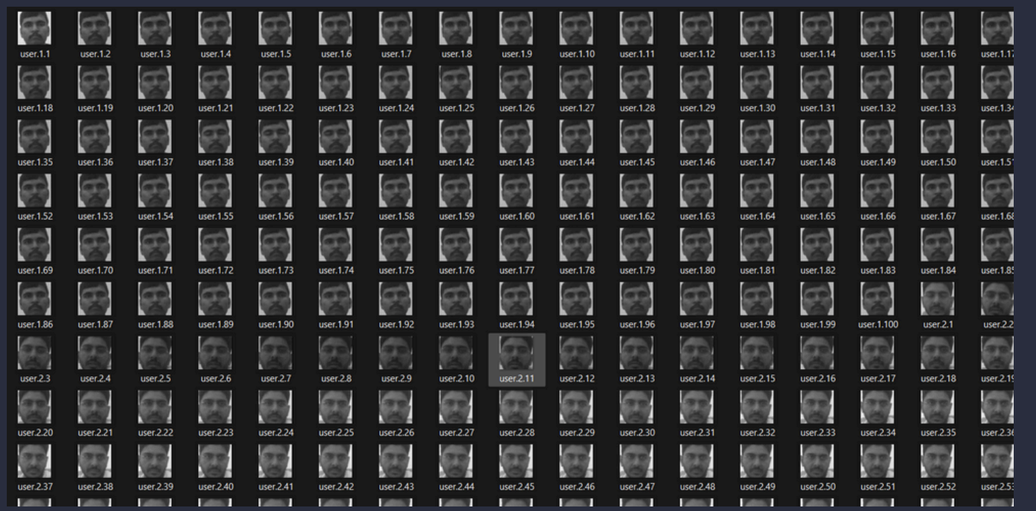
RollNo	Name	Course	Department	Batch	Semester	Email	Gender	DateOfBirth	PhotoSample
23290	Sangam	BS	Data Science and Engineering	23	Odd	sangam23@iiserb.ac.in	Male	15/11/2004	Yes
23292	Sankul	BS	Data Science and Engineering	23	Odd	sankul23@iiserb.ac.in	Male	13/02/2004	Yes
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3

4

5





Limitations

Anti face spoofing refers to techniques used to detect and prevent fake or fraudulent face inputs during face recognition.

Goal: Verify liveness and protect face recognition systems from being fooled.

It ensures the system can tell the difference between a real live face and spoof attacks like:

- Printed photos
- Mobile screen photos/videos
- 3D masks
- Deepfake video

---

https://github.com/minivision-ai/Silent-Face-Anti-Spoofing/blob/master/README_EN.md

Thank You