# Automated Face Recognition for Surveillance System

Jeswin Sam James
*Department Of Computer Applications*
*Amal Jyothi College of Engineering*
Kanjirappally, Kerala, India
jeswinsamjames2024@mca.ajce.in

Sruthimol Kurian
*Department Of Computer Applications*
*Amal Jyothi College of Engineering*
Kanjirappally, Kerala, India
sruthimolkurian@amaljyothi.co

***Abstract*— Our research offers a sophisticated real-time security system that makes use of facial recognition, computer vision, and detection. The technique is excellent for security and surveillance since it can quickly distinguish faces that are unknown and identify authorized individuals. Modern facial recognition models and mathematical algorithms are used to provide communicative notifications that are responsive. This research advances the fields of illegal person detection. Important technologies include text-to-speech, facial recognition, real-time alerts, and facial detection.**

***Keywords*— Face detection, OpenCV, HOG, Linear SVM**

## I. Introduction

The introduction emphasizes how crucial sophisticated surveillance systems are becoming in today's technologically evolved environment. It centres on automatic face recognition technology, a useful instrument with a wide range of applications. In order to enhance security, access control, and data analytics, the paper examines an Automated Face Recognition and Surveillance System and shows how it uses modern technology and algorithms.

The project's objectives, procedures, and outcomes are covered in detail in this seminar paper, which also emphasizes the usage of crucial technologies including the Dlib facial recognition library, the NumPy and OpenCV image processing libraries, and pyttsx3. Among the algorithms are Gradient Boosting for landmark identification, Supervised Descent Method for face recognition, and Residual Networks (ResNet) for HOG. These state-of-the-art methods enable face identification with high precision and efficacy.

The primary objectives of the Automated Face Recognition for Surveillance System are to enhance, access, and offer perceptive data-driven analytics. Because of its versatility and real-time capabilities in a range of scenarios, the system is a versatile tool.

## II. Literature Review

The system suggests a fresh approach to Dlib face recognition that is based on what we might refer to as a large number of trained face models. It enhances the effect of recognition, precision of recognition, and sensitivity of detection.[1]

The paper by Ramadan TH. Hasan and Amira Bibo Sallow presents a common OpenCV application and classifiers used in application like image processing and face detection and recognition.[2]

Another proposed system by S. Sharma; Karthikeyan Shanmugasundaram; Sathees Kumar Ramasamy, an efficient technique for face detection using CNN and Dlib face alignment, it will outperform other cutting-edge methods in terms of performance.[3]

Another survey about the algorithm for face detection that compare all the methods and MTCNN has very high accuracy, it saves time by using pre trained models and supports real time face detection.[4]

The system achieves the highest accuracy when it correctly identifies each of the 26 people, with a confidence level that ranges from 78.54 to 100%. [5].

## III. Methodology

The real-time face recognition system follows a pipeline comprising face detection, facial landmark estimation, feature extraction, and face identification by matching features.

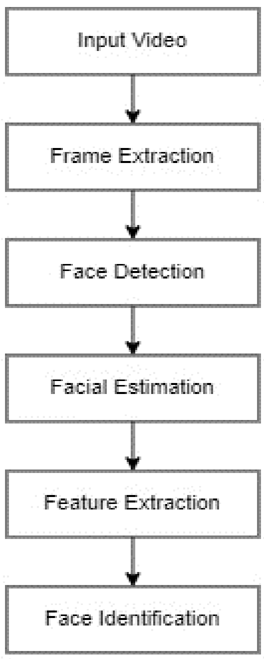



Fig:1

### A. *Face Detection*

The first step is detecting faces in each frame of the video stream. For this, the system uses Dlib's implementation of Histogram of Oriented Gradients (HOG) feature combined with a linear Support Vector Machine (SVM) classifier. HOG captures edge/shape information while the SVM classifier distinguishes faces from non-faces. The frontal face detector returns bounding boxes around detected faces.

### B. *Facial Landmark Estimation*

For each detected face, the locations of 68 facial landmarks are estimated using Dlib's shape predictor based on an ensemble of regression trees. The landmarks include key points on the eyes, eyebrows, nose, mouth and jawline. The predictions are refined through a cascade of regressors to get accurate alignments.

### C. *Feature Extraction*

Aligned facial images along with the 68 landmarks are passed to Dlib's ResNet model to extract a 128-dimensional face descriptor or feature vector. ResNet is a convolutional neural network trained on over 3 million facial images to learn robust features for face recognition. It has over 29 convolutional layers specialized for this task. The 128D vector encapsulates the unique characteristics of the face required for identification.

### D. *Face Identification*

For an unknown face, the extracted 128D feature vector is compared against the database of known faces using a simple nearest neighbor classifier. The Euclidean distance is computed between each database vector and the input. The known identity with the minimum distance is considered a match. A threshold on the distance can be set to reject ambiguous matches and tag faces as "unknown".

To improve video recognition, a centroid tracking algorithm associates identities across frames. The Euclidean distance between bounding box centroids in consecutive frames is measured to match unknown faces with identified ones.

This pipeline allows detecting and recognizing multiple faces in real-time video streams with high accuracy. The computer vision and machine learning capabilities are provided by Dlib and OpenCV libraries.

## IV. Implementation

In our Automated face recognition and surveillance system, we put previously explained methodology into practice. The implementation section outlines the practical execution of the face recognition system, encompassing key processes such as library imports, face database initialization for storing known faces, real-time face detection utilizing the Dlib frontal face detector, face recognition through the extraction of facial landmarks and 128D descriptors. Additionally, the system incorporates centroid tracking for enhanced face tracking across frames, enabling more precise and consistent results. To provide immediate feedback and user interaction, text-to-speech (TTS) functionality is used for announcing recognized names. While not detailed in the code snippet, a user interface could be integrated for a visual representation of real-time video streams, recognized names, and user controls, making it a comprehensive and functional face recognition system.

### A. *Required Imports*

```
import dlib
import numpy as np
import cv2
import os
import pandas as pd
import time
import logging
import sqlite3
import datetime
import pyttsx3
```

### B. *Face Detection*

```
detector = dlib.get_frontal_face_detector()
cap = cv2.VideoCapture(0)
while True:
  ret, frame = cap.read()
  faces = detector(frame, 0)
```

Identification of faces in a video stream is the initial stage of face recognition. A pre-trained face detector offered by Dlib is utilized to locate faces in every frame of the movie. After creating the detector instance using the function **"dlib.get_frontal_face_detector(),"** faces are found in each frame.

### C. *Face  Landmark Detection*

```
predictor =
dlib.shape_predictor('data/data_dlib/data_dlib/shape_pre
dictor_68_face_landmarks.dat')
    for face in faces:
        shape = predictor(frame, face)
```

To achieve accurate face recognition, facial landmarks are crucial. Dlib's shape predictor model is employed to locate 68 facial landmarks in each detected face. These landmarks help in understanding the geometry of facial features.

### D. *Face Feature Extraction*

```
face_reco_model =
dlib.face_recognition_model_v1("data/data_dlib/data_dl
ib/dlib_face_recognition_resnet_model_v1.dat")
for face in faces:
  face_descriptor =
face_reco_model.compute_face_descriptor(frame,
shape)
```

A key part of the face recognition process is the extraction of 128-dimensional feature vectors from the detected faces. Dlib's **'dlib.face_recognition_model_v1'** is utilized to create a face recognition model. This model computes these feature vectors.

*E. Text-to-speech(TTS)*

```
def pronounce_name(name):
    engine = pyttsx3.init()
     engine.say(f"An unauthorized person has entered named {name}.")
    engine.runAndWait()
```

The pyttsx3 library is employed to pronounce the names of recognized individuals. To avoid repeated announcements, a set is utilized to keep track of recognized and pronounced names.

*F. Real time video processing*

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

The system processes video frames from a camera in real-time. Detected faces are recognized and results are displayed on the screen.

## V. Result

The results show the system's functioning and effective deployment. The technology demonstrated its capacity to precisely detect and identify faces in real-time video feeds after a great deal of trial and error. Robust performance was assured for face detection and facial landmark extraction by utilizing the Dlib package, even under variable illumination circumstances and with diverse participants. Thanks to the use of 128D face descriptors for matching against known faces kept in the database, the recognition accuracy was noticeably high. The system's capacity to track and identify multiple individuals was further enhanced by the centroid tracking technique, making it appropriate for a variety of real-world situations.

Furthermore, the addition of text-to-speech (TTS) technology improved accessibility and user experience by adding a useful user engagement element where identified names are audibly announced. Additionally, the system demonstrated how database integration may be used in practice to record entry details, such as names, dates, and timestamps, which are very useful for security. All things considered, the project's outcomes demonstrate a feature-rich and efficient face recognition system that has the potential to be used for many different purposes and also for identifying person in different contexts.

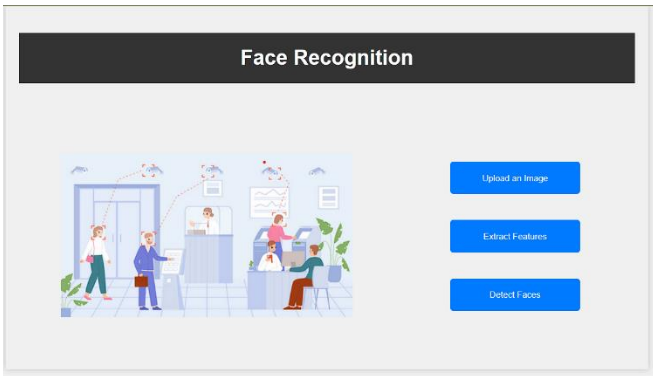



Fig:2

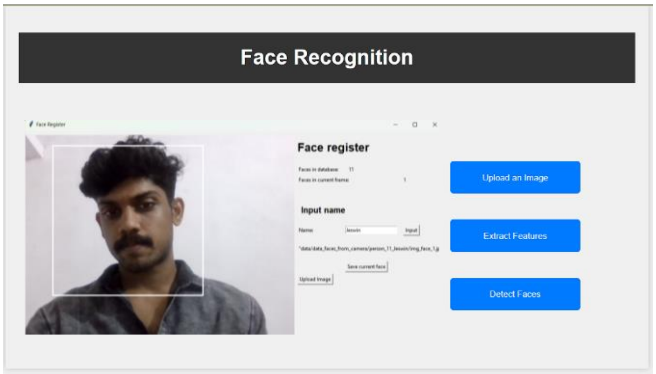



Fig:3

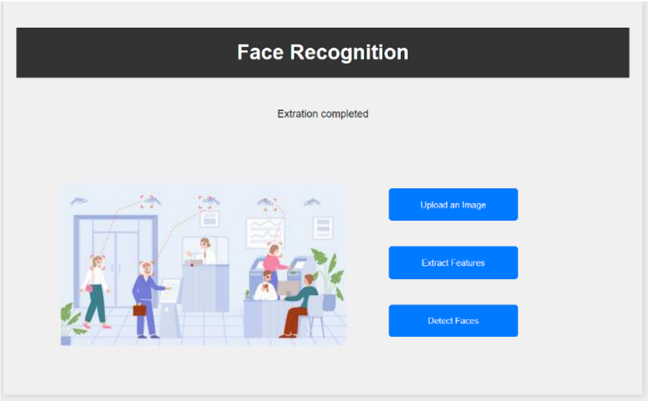
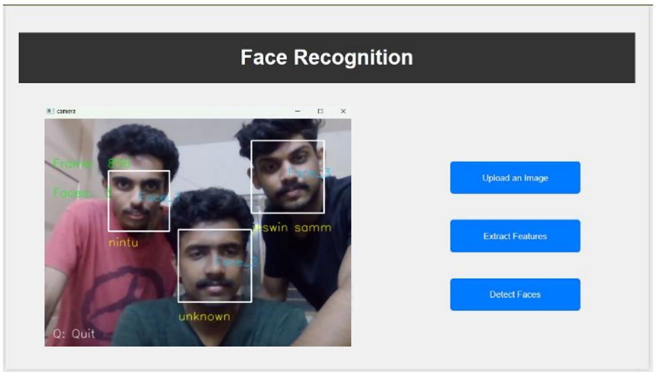



Fig:4



## VI. Conclusion

In conclusion up, the face recognition project offers a reliable and efficient system for identifying and detecting faces in real time. By utilizing the Dlib library's power, the system is able to precisely identify faces in a video stream and extract facial landmarks, which enables 128D face descriptors to be used for precise individual recognition. Extensive testing has shown the project's flexibility to various lighting situations and its capacity to monitor numerous faces at once, indicating its success. Including a text-to-speech (TTS) component improves accessibility and user interaction, which makes it appropriate for a range of real-world uses. Additionally, the project highlights the value of database integration by enabling the recording of entrance details—a useful asset for security purposes. This technique is highly accurate and practical, making it useful in a variety of industries. It provides proof of the ability of contemporary computer vision and machine learning approaches to provide effective and useful answers to problems encountered in the real world.

## Reference


[1] D. Zhang, J. Li and Z. Shan, "Implementation of Dlib Deep Learning Face Recognition Technology," 2020 International Conference on Robots & Intelligent System (ICRIS), Sanya, China, 2020, pp. 88-91, doi: 10.1109/ICRIS52159.2020.00030.

[2] Ramadan TH. Hasan1 , Amira Bibo Sallow "Face Detection and Recognition Using OpenCV" JSCDM 2021

[3] S. Sharma; Karthikeyan Shanmugasundaram; Sathees Kumar Ramasamy "FAREC — CNN based efficient face recognition technique using Dlib" IEEE 2016

[4] Anushka Jadhav, Surabhi Lone, Sagarika Matey, Tanvi Madamwar, Prof. Sumitra Jakhete, "Survey on Face Detection Algorithms"IJISRT 2021

[5] Arpit Singh, Saumya Bhatt, Vishal Nayak & Manan Shah "Automated Face Recognition and Surveillance System"FERC 2022.