K.R. Mangalam University

SCHOOL OF ENGINEERING AND TECHNOLOGY

Mini Project Report On

"Face Recognition Using Python"

Submitted By:

- Tarun Sharma (Team Leader)2401730104
- Swastik Sharma 2401730032
- Gourav 2401730019

Guided By:

- Mr. Apporva Jain

CERTIFICATE

This is to certify that the mini project entitled **"Face Recognition Using Python"** has been successfully completed by the following students:

1.**Tarun Sharma** (Team Leader)

2.**Swastik Sharma**

3.**Gourav**

Under the guidance of **Mr. Apporva Jain**, in partial fulfilment of the requirements for the completion of the mini project in the Department of Computer Science and Engineering, **K.R. Mangalam University.**

K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our guide, Mr. Apporva Jain, for his invaluable support, guidance, and encouragement throughout this project.

His knowledge and insight have been instrumental in the completion of our mini project. We would also like to thank our faculty members and the Department of Computer Science and Engineering for providing us with the necessary resources and environment to work on this project.

Finally, we thank each member of our team for their cooperation and hard work in making this project successful.

ABSTRACT

Proposed:

The purpose of face recognition using Python is to identify or verify individuals in images or video streams by analyzing facial features, enabling applications like surveillance, authentication, and access control using open-source libraries such as OpenCV and face recognition.

Features:

- **Real-time face detection and recognition**
- **High accuracy using facial encodings**
- **Non-intrusive biometric verification**
- **Easy integration with cameras and video streams**
- **Support for multiple faces in a frame**
- **Works with open-source libraries like OpenCV and face_recognition**
- **Can label recognized and unrecognized faces**

BENEFITS/IMPACTS

- **Cost-effective due to open-source tools and libraries**
- **Easy to implement with libraries like OpenCV and face_recognition**
- **Real-time processing for instant identification**
- **Enhances security in restricted areas**
- **Automates attendance and identity verification**
- **Non-contact method—more hygienic than fingerprint scanners**

1. INTRODUCTION

Face recognition using Python is a technology that can identify or verify a person by analysing their facial features from an image or video. It is widely used in areas like security systems, mobile phone unlocking, and attendance tracking. This technology works by detecting a face, converting it into a set of numerical features (called encodings), and then comparing it with stored data. Python makes this process easier with powerful libraries like OpenCV and face recognition. These tools allow us to build face recognition systems with less code and high accuracy. It is a fast, reliable, and contactless method of identification.

2. Review of Recent Works (2020 onwards)

1.Dardagan et al. (2021): Evaluated seven object trackers in OpenCV against the MOT20 dataset, highlighting the strengths and limitations of each in multi-object tracking

2.Chapel & Bouwmans (2020): Provided a comprehensive review of methods for detecting moving objects with moving cameras, categorizing approaches based on scene representation and motion compensation techniques.

3.Vadlamudi (2020): Explored object tracking systems using OpenCV in Python, combining YOLO detection with built-in trackers to enhance tracking accuracy in various

4.Mishra et al. (2022): Developed an intelligent motion detection system using OpenCV, addressing challenges like environmental fluctuations and emphasizing applications in surveillance and object counting. ResearchGate

5.Pandey et al. (2022): Discussed methodologies to detect and track any moving object using OpenCV, emphasizing the importance of real-time processing and adaptability in varying scenarios.

6.Yu et al. (2018): Introduced ReMotENet, a unified, end-to-end method using spatialtemporal attention-based 3D ConvNets for efficient motion event detection in large-scale home surveillance

7.GeeksforGeeks (2022): Compared various background subtraction techniques, including Adaptive Background Learning and ZivkovicGMM, highlighting their respective advantages in motion detection tasks.

8.IJERT Survey (2020): Analyzed motion detection, tracking, and classification methods for automated video surveillance, discussing the efficacy of techniques like temporal differencing and Gaussian Mixture Models.

9.ResearchGate Study (2020): Conducted a comparative study of motion detection

methods for video surveillance systems, emphasizing the limitations of traditional GMM approaches under varying illumination conditions. ResearchGate 10. ScienceDirect Survey (2022): Reviewed moving object detection methods, focusing on practical perspectives and the integration of lightweight CNN models for improved foreground.

11. SpringerLink Chapter (2022): Detailed the design and implementation of real-time object detection systems based on single-shot detectors and OpenCV, emphasizing applications in smart vehicle systems. Frontiers

12. MDPI Study (2019): Introduced WisenetMD, a motion detection method using dynamic background region analysis, addressing challenges posed by dynamic backgrounds and varying weather conditions.

13. Journal of Big Data (2019): Reviewed intelligent video surveillance techniques through deep learning, discussing models like GMM–MRF and HMM for crowd analysis and abnormal behavior detection.

14. Tsai & Yeh (2019): Proposed an adaptive frame differencing method for intelligent moving object detection, enhancing accuracy in dynamic environments.

15. Hou et al. (2020): Developed a lightweight CNN model for refining moving vehicle detection from satellite videos, emphasizing efficiency and accuracy.

16. Giraldo et al. (2021): Introduced a fast, lightweight 3D separable convolutional neural network with multi-input multi-output capabilities for moving object detection.

17. Fu et al. (2017): Presented a fast detection method for moving objects based on an improved frame-difference approach, focusing on real-time applications.

18. Zhu & Wang (2017): Explored foreground detection via background subtraction and improved three-frame differencing, enhancing motion detection accuracy.

19. (2019): Discussed intelligent moving objects detection via adaptive frame

differencing methods, addressing challenges in varying environmental conditions. ScienceDirect 20. Ju et al. (2020): Provided insights into moving object detection based on smoothing

2. Objectives

- To study the concept and working of face recognition technology.

- To develop a simple face detection and recognition system.

- To implement the project using Python and OpenCV.

- To test and evaluate the system for accuracy and performance.

3. Tools and Technologies Used

- Programming Language: Python

- Libraries: OpenCV, NumPy, face recognition

- Development Environment: Jupyter Notebook / VS Code

- Hardware: Webcam / External camera

- OS: Windows/Linux

4. Methodology

4.1 Data Collection

Images of team members were collected under different lighting conditions and angles for training the

recognition model.

4.2 Face Detection

Used Haar Cascade Classifiers from OpenCV to detect faces in real-time video streams.

4.3 Face Encoding

Used the face_recognition library to convert facial features into numerical vectors (encodings).

4.4 Face Matching

Compared new face encodings with known ones using Euclidean distance to determine the identity of the

person.

5. System Architecture

1. Input: Real-time video from camera.

2. Face Detection: Identify faces in each frame.

3. Encoding: Convert detected faces into encodings.

4. Comparison: Match with known encodings.

5. Output: Display the name of the identified person or 'Unknown'.

Project Report: Face Recognition System

6. Results

- Successfully detected and recognized registered faces with over 90% accuracy in well-lit environments.

- Recognition performance decreased slightly in low-light or angled faces.

- Unregistered faces were labeled as 'Unknown'.

7. Applications

- Smart surveillance systems

- Attendance monitoring in schools and offices

- Access control in secure environments

- Personalized user experiences in smart devices

8. Future Enhancements

- Integrate deep learning models for better accuracy.

- Improve performance in low-light conditions using image preprocessing.

- Add emotion recognition and age estimation features.

- Build a GUI for easier use.

9. Conclusion

This project demonstrates the fundamentals of a face recognition system using Python and open-source

tools. The implementation provides a basic but functional system that can be extended for real-world

applications. The project helped the team understand the integration of computer vision and machine learning

in biometric systems.

10. Team Contribution

Project Report: Face Recognition System

Name	Role
Tarun Sharma	Team Leader, Developer, Report Writer
Swastik Sharma	Data Collection, Testing & Documentation
Gourav	Implementation, Debugging & Presentation