International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04109
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

UNAUTHORIZED ACCESS DETECTION SYSTEM

Nikita V. Zurange , Sameer S. Kakade

Nikita V. Zurange MCA & Trinity Academy Of Engineering ,Pune
Sameer S. Kakade MCA & Trinity Academy Of Engineering ,Pune

ABSTRACT- In an era where personal and organizational data security is paramount, traditional authentication mechanisms such as username and password are no longer sufficient to prevent unauthorized access. This project presents a Flask-based real-time facial recognition system integrated with OTP (One-Time Password) verification and email alert mechanisms to enhance device security. The system uses the face recognition library to compare the user's face with a pre-stored authorized image via webcam. If the face is recognized, the system sends a randomly generated. OTP to the user's registered email for secondary verification. Only after successful OTP verification can the user proceed to login using their credentials, which are validated against a MySQL database.

Key Words: data security, authentication, facial recognition, flask framework, otp verification, email alert.

1.INTRODUCTION

With the increasing reliance on digital systems for handling sensitive information, securing access to personal and organizational data has become a critical challenge. Traditional authentication techniques, primarily based on static credentials such as usernames and passwords, have proven to be inadequate in the face of modern security threats. These methods are susceptible to various attacks, including phishing, credential stuffing, and brute-force intrusions, thereby compromising system integrity and user privacy.

In response to these challenges, multi-factor authentication (MFA) has emerged as a more secure alternative by incorporating multiple layers of user verification. Among the various biometric methods, facial recognition has gained prominence due to its non-intrusive nature and ease of integration with camera-enabled devices.

2. BODY OF PAPER

2.1 FACIAL RECOGNITION AUTHENTICATION

Facial recognition is employed as the first level of authentication. Using the face_recognition Python library, the system captures a real-time image from the user's webcam and compares it with a pre-stored facial image. This comparison is achieved through facial encodings and Euclidean distance measurement to determine similarity. If the face matches a registered user, the system initiates the second stage: OTP verification.

2.2 OTP GENERATION AND EMAIL VERIFICATION

To enhance the security of the login process, the system employs an OTP (One-Time Password) mechanism. Once facial recognition is successful, the backend generates a random six-digit OTP using Python's random library. This OTP is sent to the user's registered email address using the smtplib and email.mime libraries. The OTP remains valid for a limited time and must be entered by the user to proceed. Invalid or expired OTPs result in denial of access.

2.3 USERNAME AND PASSWORD AUTHENTICATION

After passing the facial and OTP authentication stages, the user is prompted to enter a username and password. These credentials are validated against records stored in a **MySQL database**. This final layer of authentication ensures that only the intended user can gain access, even if their face and email are compromised.

Table -1: technology stack used in the authentication system

Module	Technology/ Tool Used	Purpose
Frontend Interface	HTML, CSS	User interaction and input forms
Backend Framework	Flask (Python)	Application logic and routing
Face Recognition	face_recognition (Python lib)	Face encoding and matching
OTP Generator	Python random module	Random OTP generation
Email Service	smtplib, email.mime	Send OTP and alert emails
Database	MySQL	Store user credentials and face data

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04109
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

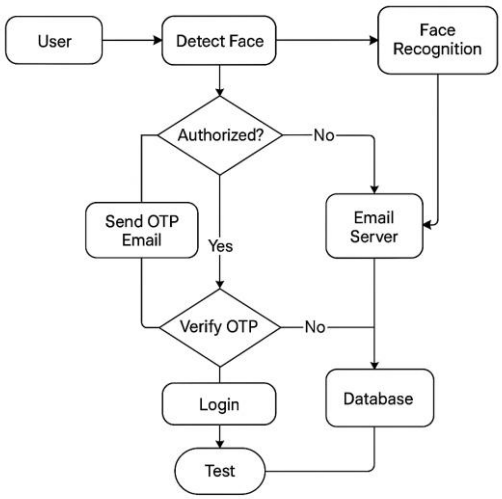
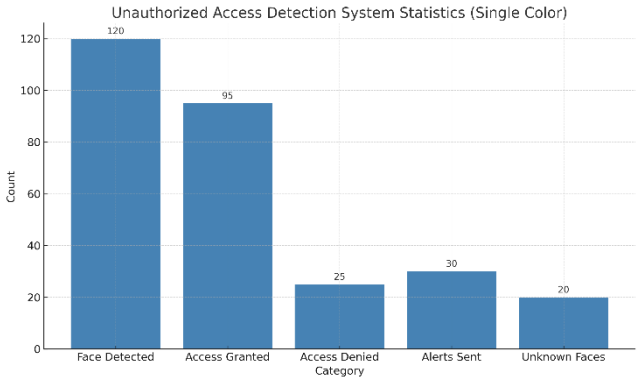


Fig -1: Figure

CHARTS



3. CONCLUSIONS

The proposed Unauthorized Access Detection System integrating facial recognition, OTP verification, and real-time alerts offers a significant improvement over traditional authentication methods. By combining biometric verification with dynamic security measures, the system minimizes the risk of unauthorized access and strengthens user authentication protocols

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who contributed to the successful completion of this project, **"Unauthorized Access Detection System."**
First and foremost, we thank our project guide and faculty members for their continuous support, expert guidance, and valuable feedback throughout the development process.
We also extend our appreciation to the developers and maintainers of open-source libraries and tools such as **Flask, OpenCV, face_recognition, MySQL, and Python**, which played a vital role in building this system.
Finally, we thank our peers, family, and friends for their constant encouragement, motivation, and support in helping us achieve our goals and overcome challenges during the project development.

REFERENCES

1. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
— Reference for Flask microframework used for web backend.
2. Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.
— OpenCV library used for video processing and face detection.
3. Geitgey, A. (2017). *face_recognition: Simple face recognition library using deep learning with Python*.
— Face recognition library used for comparing face encodings.
4. Oracle Corporation. (2021). *MySQL 8.0 Reference Manual*.
— Relational database system used for user authentication and logging.
5. Python Software Foundation. (2023). *smtplib — SMTP protocol client*.
— Used for sending OTP and alert emails.
6. Python Software Foundation. (2023). *email.mime — MIME handling package*.