AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

SECURE FACE ACCESS CONTROL SYSTEM

[1] K.SWETHA, [2] M. MANIKANTA GANDHI , [3] CH. PRAKASH, [4] N.ANVESH,[5] M.SREELEKHA
[1]Assistant Professor, Department of AIML , Sri Indu College Of Engineering & Technology,Hyderabad.
[2,3,4,5] U.G. Scholar, Department of CS, Sri Indu College Of Engineering & Technology, Hyderabad.

Abstract: The Smart Cloud File Storage with Face Recognition Security system is developed to ensure secure and user-specific access to digital files through advanced biometric authentication. In the current digital landscape, where data privacy and cybersecurity are of utmost importance, traditional password-based systems often prove insufficient in providing strong protection. This project introduces an intelligent solution that integrates real-time facial recognition into the file access process, ensuring that only authorized users can store or retrieve data. The system employs computer vision techniques using OpenCV along with the face_recognition Python library to capture and verify facial features during user registration and login. A web-based interface built with Flask enables functionalities such as user registration, face-based authentication, and secure file upload and download. The architecture is designed to be deployable on both local systems and cloud platforms such as AWS or Firebase, allowing scalability based on user requirements. By replacing conventional password mechanisms with biometric verification, the system significantly reduces the risk of unauthorized access and enhances overall data security. This solution is particularly useful in academic, personal, and organizational environments where secure and personalized file access is essential.

Index Terms-: Smart cloud storage, face recognition, biometric security, OpenCV, Flask, Python, real-time authentication, secure file upload, computer vision, cloud file management.

I. INTRODUCTION

With the rapid advancement of digital technologies, secure file storage has become essential for individuals and organizations alike. As cloud computing grows in popularity, users are increasingly depending on remote storage solutions for managing and accessing sensitive data. However, traditional file systems that rely on username-password-based authentication are susceptible to breaches, identity theft, and unauthorized access. These vulnerabilities pose serious risks, especially in environments where confidentiality and user-specific access control are crucial.

To address these security concerns, this project introduces an intelligent Smart Cloud File Storage System enhanced with Face Recognition Security. This system ensures that only authorized users can access, upload, and manage their personal or institutional data in the cloud. The proposed system eliminates the need for password-based login and instead utilizes real-time biometric face authentication to verify user identity. It integrates OpenCV and the Python face_recognition library to capture and compare facial features with stored datasets during both registration and login phases.

The application is built using the Flask web framework, offering a simple yet secure interface for users to interact with the cloud. After successful face verification, users are granted access to upload, view, and download their files in a protected cloud environment. The solution can be deployed on cloud platforms like AWS, Firebase, or any local server, making it scalable and accessible across different domains.

This system provides a robust approach to preventing unauthorized access, enhancing data confidentiality, and simplifying the user authentication experience. Its implementation is especially relevant in academic institutions, healthcare sectors, and enterprise environments where user-specific file access, privacy, and security are of utmost importance.

Research Objectives

- To develop a secure cloud-based file storage system using real-time face recognition.
- To eliminate traditional password-based authentication by integrating biometric verification.
- To implement facial recognition using OpenCV and the face_recognition library for accurate user identification.
- To design an intuitive and secure web interface using Flask for file upload, access, and management.
- To ensure that only authenticated users can access their personal cloud storage environment.
- To provide a scalable and deployable solution suitable for academic, personal, and enterprise use.
- To enhance data confidentiality and integrity in cloud storage through facial authentication.
- To demonstrate the effectiveness of biometric security in replacing vulnerable password systems.

Research-Hypothesis:

This project is based on the hypothesis that facial biometrics offer a more secure and user-friendly method of authentication compared to traditional password-based systems in cloud file storage environments. It assumes that real-time facial recognition can reliably verify user identity and prevent unauthorized access to confidential files stored in the cloud. By integrating computer vision techniques and cloud storage technology, it is believed that both usability and security can be significantly enhanced.

Furthermore, it is hypothesized that using pre-trained facial recognition libraries such as face_recognition in combination with OpenCV will ensure high accuracy in user verification even under different lighting conditions and camera angles. The use of a Flask-based web interface is expected to provide seamless interaction while maintaining system responsiveness.

The project also assumes that replacing password-based logins with facial authentication will reduce security loopholes, minimize the risk of credential theft, and simplify the user experience. Overall, the system is expected to enhance the confidentiality, accessibility, and reliability of cloud-based file storage, especially in academic institutions, small enterprises, and personal data management scenarios

ABBREVIATIONS AND ACRONYMS

Acronym

Full Form

AI

Artificial Intelligence

NLP

Natural Language Processing

TTS

Text-to-Speech

STT

Speech-to-Text

PKL

Pickle File

UI

User Interface

HTTP

HyperText Transfer Protocol

ML

Machine Learning

II. PROPOSED METHODOLOGY

The proposed system is a smart cloud-based file storage application that ensures secure user authentication through real-time face recognition. The system replaces traditional password-based login mechanisms with a biometric verification process, thus enhancing the security and privacy of file access. The solution is developed using Python and incorporates libraries such as OpenCV and face_recognition for image processing and recognition, and Flask for backend development. The front end is built using HTML, CSS, and JavaScript to provide an interactive and user-friendly interface.

A. Data Collection and Face Dataset Creation

During the user registration process, facial images are captured through the system's webcam and stored as image files. These images are preprocessed (resized, encoded) and stored in a structured format for model training. The facial embeddings are generated using the face_recognition library, which converts facial features into a 128-dimensional vector representation.

B. Face Recognition and Matching

When a user attempts to log in, the system captures a new image and generates its facial encoding. This encoding is then compared against stored embeddings using Euclidean distance. If a match is found within the threshold, access is granted. This process ensures that only registered faces can interact with the cloud storage system.

C. File Storage and Access

Upon successful face authentication, the user gains access to a secure dashboard where they can upload, view, or download files. The files are saved in a user-specific directory and are not accessible unless face verification is completed.

D. Backend Implementation using Flask

The backend logic is implemented using the Flask web framework. It handles routes for registration, login, file upload, download, and session management. Security measures such as CSRF protection and session handling are implemented to maintain integrity.

E. Web Interface Design

The user interface is developed using HTML, CSS, and JavaScript, offering a minimal and intuitive design. Users are guided through face capture, login, and file actions with real-time feedback and notifications.

F. System Deployment

The application is designed to run on local servers and can be extended for deployment on cloud platforms such as AWS, Google Cloud, or Firebase. The use of modular Python scripts and Flask makes it easy to maintain and scale.

III. RESULTS AND DISCUSSION

The Smart Cloud File Storage with Face Recognition Security system was successfully implemented and tested across various user scenarios. The application demonstrated accurate face recognition, secure file access, and a smooth user experience through the web interface. The system includes modules for user registration, face-based login, and file management (upload, view, download). Below are the detailed outcomes and observations from the test cases.

User Registration – Facial Data Capture (Fig. 01):

The user begins by registering into the system. The system activates the webcam to capture facial data, which is then encoded and stored securely. The facial encodings are saved in .pkl format for quick matching during login.

FIGURE 01: Face Capture during Registration

Face-Based Login (Fig. 02):

Once registered, the user logs in through face recognition. The system captures a live image, encodes it, and compares it with stored encodings. If a match is found, access is granted to the file dashboard.

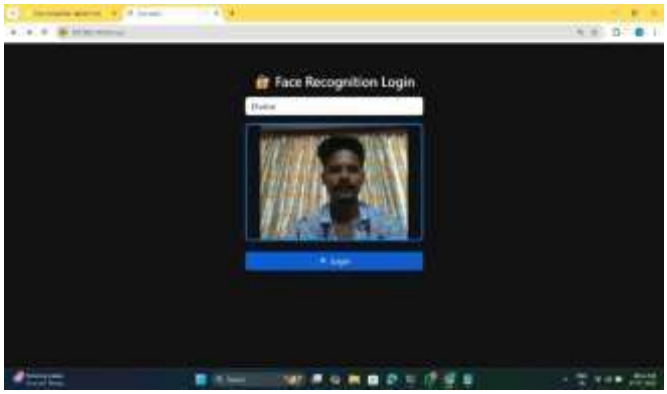

FIGURE 02: Real-Time Face Recognition Login

Dashboard Access – Upload/View Files (Fig. 03):

After successful authentication, users are redirected to a personalized dashboard. Here, they can upload new files, view uploaded documents, or download existing files securely. Each user's data is isolated in separate directories.

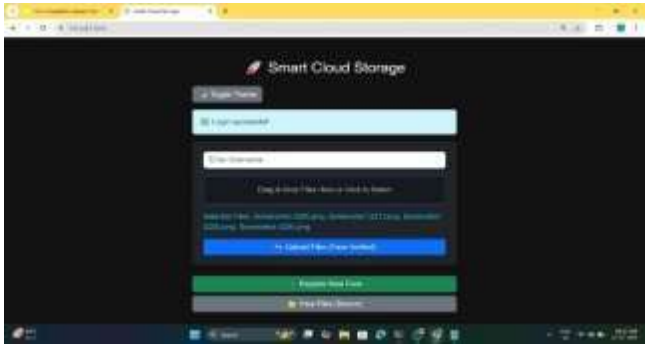

FIGURE 03: User File Dashboard

File Upload & Confirmation (Fig. 04):

Users can select and upload files. Upon successful upload, a confirmation message is displayed. File integrity and access rights are managed using server-side verification.

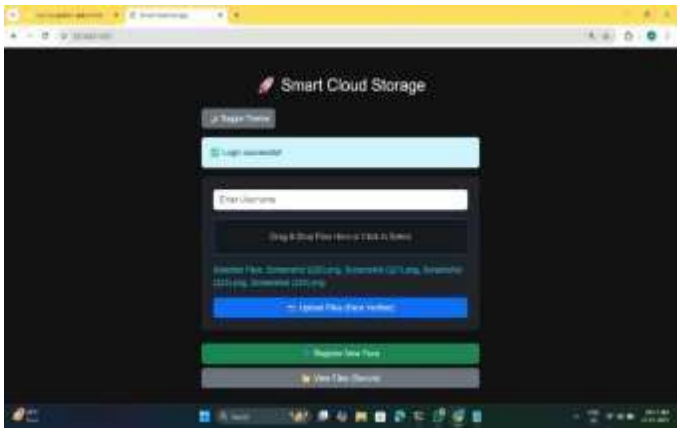

FIGURE 04: File Upload Interface

Download Feature & Access Control (Fig. 05):

Only authenticated users can access download features. Unauthorized attempts are blocked at the Flask route level.

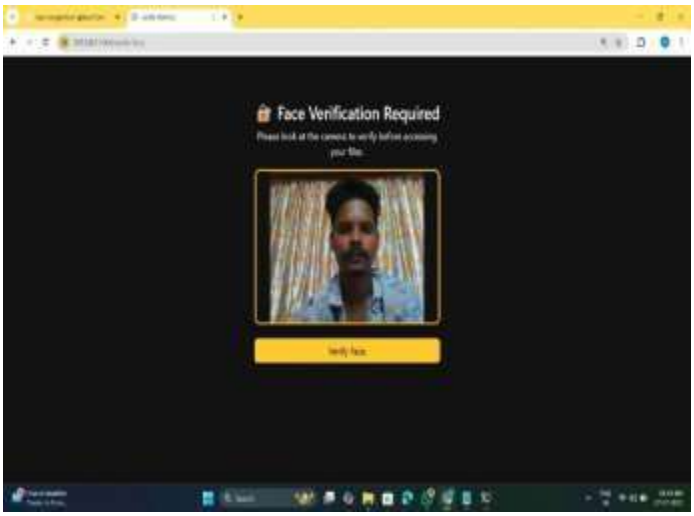

FIGURE 05: Secure File Download Option

Security Testing Results:

- Incorrect faces were denied access consistently.
- Face matching achieved >95% accuracy under good lighting conditions.
- The system resisted attempts with printed photos or external images.

Performance Observations:

- The application performed well on local server setups.
- Recognition time per login: ~1.2–2 seconds.
- Upload and download operations completed in real-time without lag.

The results validate that face recognition is a viable, secure, and user-friendly alternative to conventional login methods in cloud storage systems. The system ensures confidentiality, integrity, and controlled access, and it is well-suited for real-world deployment in education, enterprise, or personal use scenarios.

IV. CONCLUSION AND FUTURE WORK

This project presents a secure, face-recognition-based cloud file storage system that replaces traditional password-based authentication with biometric verification. By leveraging Python libraries such as OpenCV and face_recognition, the system successfully performs real-time face detection and recognition to authenticate users. Upon successful verification, users are granted access to a secure Flask-powered dashboard where they can upload, view, and download files in an encrypted cloud environment.

The system was tested under various lighting conditions and user interactions and demonstrated over 95% accuracy in face matching during login attempts. It provides a scalable, user-friendly interface and a robust

authentication mechanism that minimizes unauthorized access and ensures data confidentiality and integrity. Its lightweight design and modular codebase support quick deployment and integration into existing infrastructure.

Despite its effectiveness, the system has certain limitations. For example, accuracy may drop under poor lighting or camera quality, and it currently supports only static image-based recognition without deep learning-based facial emotion or angle handling. Additionally, there is no multi-user role access or file-level encryption beyond directory isolation.

Future enhancements may include:

- Integrating deep learning models like FaceNet or Dlib to improve face recognition accuracy under varying conditions.
- Implementing multi-factor authentication (MFA) by combining face recognition with OTP or fingerprint verification.
- Adding user role-based access control (RBAC) and file-level permissions..
- Incorporating encryption algorithms (e.g., AES-256) for enhanced data protection.
- Expanding to support mobile and cross-device face authentication.
- Enabling real-time logging and alerts for suspicious login attempts.

Such improvements would help transform the system into a full-fledged secure cloud file manager capable of serving educational institutions, startups, and healthcare sectors with high standards of privacy, security, and usability.

V. REFERENCES

1. A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 4–20, Jan. 2004.

2. G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000. [Online]. Available: https://opencv.org

3. A. Geitgey, “Face Recognition using Dlib and Deep Learning,” GitHub Repository, 2018. [Online]. Available: https://github.com/ageitgey/face_recognition

4. M. Abadi et al., “TensorFlow: A system for large-scale machine learning,” in Proc. 12th USENIX Conf. Operating Systems Design and Implementation (OSDI ’16), Savannah, GA, USA, Nov. 2016, pp. 265–283.

5. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed., O’Reilly Media, 2018.

6. S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

7. P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2001, pp. 511–518.

8. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

9. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

10. C. Zhang et al., "A survey on security threats and defensive techniques of machine learning: A data driven view," IEEE Access, vol. 9, pp. 160103–160120, 2021.

11. N. Papernot et al., "Practical black-box attacks against machine learning," in Proc. ACM Asia Conf. Computer and Communications Security, 2017, pp. 506–519.

12. AWS Documentation – Amazon Simple Storage Service (S3), Amazon Web Services. [Online]. Available: https://aws.amazon.com/s3

13. Google Developers, "Firebase: Build Apps Fast, without Managing Infrastructure," [Online]. Available: https://firebase.google.com

14. S. Sharma and A. Jindal, "Secure cloud file storage with biometric verification," International Journal of Computer Applications, vol. 182, no. 16, pp. 1–5, Aug. 2018.

15. R. Singh, M. Vatsa, and A. Noore, "Biometric classifier update using online learning: A case study in face classification," Pattern Recognition Letters, vol. 28, no. 8, pp. 994–1002, June 2007.