ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-1/453-460
A. Yashwanth *et. al.*, /International Journal of Engineering & Science Research

Full Length Article

Secure And Real-Time 1-To-N Face Recognition System For Web-Based User Authentication

T.Ramakrishna[1],A. Yashwanth[2],B. Prabhas[3],B. Vishnu Vardhan[4]
[1]Assistant Professor; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.
[2,3,4]B.Tech Students; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.
Mail Id; yashwanthaella@gmail.com[2], banothprabhas74@gmail.com[3],bathulavishnu519@gmail.com[4]
Accepted 25-03-2026

Abstract
Traditional authentication mechanisms primarily rely on textual credentials such as usernames, passwords, or identification numbers. These approaches are prone to security risks including password theft, guessing attacks, and user forgetfulness. Although face recognition has been explored as an alternative many existing solutions depend on static image storage or manual verification, which limits automation and real-time usability. Furthermore, such systems often lack dynamic retrieval of user-specific information, leading to inefficient and less secure identification processes.To address these limitations, this paper proposes a real-time face capture and identification framework integrated with OpenCV for intelligent user authentication. During the registration phase, the system captures facial data live through a web-based interface connected to a webcam using OpenCV. The acquired facial images are securely stored in the server-side database or file system. During authentication, a new live image is captured and processed using face similarity algorithms to compare it with previously stored templates. Once a match is confirmed, the system dynamically retrieves associated user details from the database and presents them securely on the interface.The proposed architecture combines real-time camera integration, secure storage mechanisms, and efficient facial comparison techniques to provide reliable identity verification. By ensuring that access is granted only to authenticated individuals, the system enhances security while improving usability. The modular design allows seamless deployment in modern web-based applications requiring automated and intelligent authentication.

Keywords— *Face Recognition, OpenCV, Real-Time Authentication, Biometric Security, Image Processing, Identity Verification.*

INTRODUCTION

Face identification has become an essential component in many real-world applications and plays a critical role in modern authentication systems. With the rapid growth of digital platforms, cloud services, and smart devices, there is an increasing demand for reliable and efficient identity verification mechanisms. Facial recognition technology is widely used in scenarios such as smartphone unlocking, attendance monitoring, surveillance systems, and secure facility access. These applications highlight the importance of automated and contactless authentication solutions that can operate efficiently without requiring complex hardware setups.However, facial data is considered highly sensitive information, and its misuse may compromise user privacy. Consequently,privacy-preserving face identification has gained significant attention in recent research. Such approaches aim to authenticate users based on facial images while protecting sensitive biometric information. Face identification protocols are generally categorized into two types: one-to-one (1:1) and one-to-many (1:N) identification. In a 1:1 system, a user's facial data is enrolled during registration and later compared against stored data during authentication. Privacy-preserving methods ensure that raw facial information is not directly exposed or accessed. In contrast, a 1:N system involves multiple registered users, and authentication is performed by comparing the input face with all stored templates to identify the correct individual without revealing sensitive information.
Earlier research, initiated by various scholars, proposed several privacy-preserving face identification techniques using cryptographic methods. Many of these approaches rely on homomorphic encryption to ensure secure comparisons. While these techniques provide strong privacy guarantees, they often suffer from computational overhead and reduced efficiency. Recent methods, such as CryptoMask, attempt to improve performance by combining encryption with

optimized similarity matching strategies. Although these solutions enhance efficiency, there remains a need for lightweight and practical face authentication systems suitable for real-time web-based applications.

Scope of the Project

The scope of this project is to design and implement a real-time face-based authentication system integrated with a web application using OpenCV. The system captures facial data dynamically through a webcam during both registration and login phases, removing reliance on static image uploads. Captured facial information is securely stored in the server database or file system. During authentication, the system compares live facial input with stored templates and retrieves corresponding user details upon successful identification.

The project adopts a modular architecture that integrates live face capture, recognition logic, database management, and secure access control. The solution can be applied in various domains such as academic portals, enterprise systems, secure login platforms, and identity verification modules. The framework is designed to be scalable, efficient, and adaptable to future enhancements.

Objective

The primary objective of this project is to develop an intelligent authentication system based on real-time face recognition using OpenCV. The system aims to replace traditional credential-based login mechanisms with facial verification to improve both security and usability. The specific objectives include enabling live face capture during registration and login, performing accurate face matching using similarity techniques, securely storing facial templates, and dynamically retrieving user information after successful authentication. Additionally, the system seeks to prevent unauthorized access, reduce identity fraud, and provide a fast, automated, and contactless authentication mechanism suitable for real-world applications.

Problem Statement

Conventional authentication systems rely heavily on textual credentials such as usernames, passwords, or identification numbers. These methods are vulnerable to security threats including password theft, brute-force attacks, duplication, and user-related errors such as forgetting credentials. Although face recognition has emerged as an alternative, many existing implementations depend on static images, manual verification, or offline comparison techniques. Such systems lack real-time face capture, automated identification, and dynamic database interaction. Furthermore, they often fail to ensure secure live verification, making them susceptible to spoofing attacks. Therefore, there is a need for a robust authentication mechanism that supports live facial capture, automated comparison, and secure retrieval of user information to provide reliable and efficient identity verification.

Existing System

Most current authentication systems continue to rely on traditional input-based methods such as usernames and passwords. Even when facial recognition is incorporated, the process typically involves pre-uploaded images rather than live camera input. These systems often lack integration with real-time video streams and advanced computer vision libraries. Additionally, face matching is commonly performed offline or in batch mode, resulting in delayed validation. Such systems also do not support dynamic retrieval of user-specific information upon successful authentication, limiting their effectiveness in modern security-critical environments.

Proposed System

The proposed system introduces a real-time face authentication platform using OpenCV for live video capture and identification. Unlike static verification models, the system captures facial data dynamically through a connected webcam, ensuring the presence of the user during authentication. During registration, facial features are captured, processed, and securely stored. During login, a new facial image is captured and compared with stored templates using similarity algorithms. Once a match is confirmed, associated user information is automatically retrieved from the database and displayed.

This architecture integrates real-time facial recognition with dynamic data retrieval, improving identification accuracy, security, and usability. The system is designed to be efficient, scalable, and suitable for modern web applications.

PROJECT DESCRIPTION

The proposed project aims to develop a secure and intelligent authentication framework based on real-time facial recognition using OpenCV. Conventional authentication approaches such as usernames and passwords remain widely adopted but are vulnerable to security risks including credential theft, sharing, and brute-force attacks. To address these limitations, the proposed system replaces text-based verification with biometric facial authentication. By utilizing live webcam input, the system ensures that authentication is performed only when a real user is present, thereby reducing the possibility of spoofing attacks that rely on static images or compromised credentials.

During the registration stage, the system captures the user's facial image in real time through a web interface integrated with OpenCV. The captured image is processed to extract distinctive facial features and is stored securely in the server file

system or database along with user profile details. This stored facial template serves as the reference for future authentication. During login or verification, the system again captures a live facial image and compares it with stored templates using similarity-based face recognition techniques. This automated process eliminates manual intervention and improves both efficiency and accuracy.

When a match is successfully identified, the system dynamically retrieves the corresponding user information from the database and displays it securely on the interface. If the facial data does not match, access is denied, ensuring strict security enforcement. The architecture follows a modular design that integrates real-time image acquisition, feature extraction, secure storage, and database interaction. The proposed solution is scalable and suitable for applications such as secure web portals, institutional management systems, and access control platforms. Overall, the project demonstrates the effectiveness of computer vision techniques in enhancing authentication security and user convenience.

Methodologies

The system is organized into the following five modules:

- **Module 1: User Registration Module** – Captures user information and facial data during enrollment and stores it securely.
- **Module 2: Live Face Capture Module** – Obtains real-time facial images using a webcam through OpenCV.
- **Module 3: Face Recognition and Matching Module** – Performs comparison between live images and stored templates.

REQUIREMENTSENGINEERING

Requirements engineering for the proposed real-time face recognition–based authentication system involves identifying, analyzing, and documenting both functional and non-functional requirements to ensure secure and efficient identity verification. The system requires a user-friendly web interface that supports user registration, live face capture, and authentication using a webcam integrated with OpenCV. It must securely store facial data and user information, perform accurate real-time face matching, and dynamically retrieve authorized user details from the database upon successful authentication. In addition, the system should provide high security, reliability, and accuracy while maintaining fast response time and ease of use. The hardware requirements include a webcam-enabled system with sufficient processing capability, while the software requirements include a web server, database management system, OpenCV library, and supporting backend technologies. These requirements collectively ensure that the proposed system delivers a scalable, efficient, and secure biometric authentication solution suitable for real-world applications.The hardware requirements define the minimum infrastructure needed for the effective development and deployment of the system. The proposed system requires a processor equivalent to Intel Core i3 or higher, at least 4 GB of DDR4 RAM, a 15.6-inch LED monitor, and a minimum of 100 GB hard disk storage. Additionally, input and interaction devices such as a keyboard and mouse are necessary, along with a webcam for capturing facial images during registration and authentication. These components ensure that the system can efficiently perform real-time face detection and recognition without performance degradation.The functional requirements describe the operations that the system must perform. The user registration module allows new users to enroll by providing personal details and capturing a live facial image through the webcam. The system processes the captured image using OpenCV and stores it securely along with user information in the database. The live face capture module activates the webcam during authentication and detects the user's face in real time, ensuring that the captured image is valid and suitable for recognition. The face recognition and matching module extracts facial features from the live image and compares them with stored templates using similarity measures. If the similarity score exceeds the defined threshold, authentication is considered successful; otherwise, access is denied. After successful authentication, the secure authentication and access control module creates a protected session and restricts unauthorized users from accessing system resources. The user data retrieval module dynamically fetches the authenticated user's information from the database and displays it securely. Additionally, the system monitoring and performance evaluation module analyzes recognition accuracy, response time, and system reliability. The database management module stores user profiles, facial data references, authentication logs, and performance metrics, ensuring efficient data retrieval and secure storage.

The non-functional requirements define the quality attributes and operational constraints of the system. The system must provide fast performance by completing live face capture and authentication with minimal delay. It should be scalable to support a growing number of users without reducing recognition accuracy. Reliability is ensured by consistent face detection and continuous availability of authentication services. The system must maintain high accuracy while minimizing false acceptance and false rejection rates. Security requirements include secure storage of facial data, prevention of spoofing attacks, and restricted access to authorized users. Usability requirements emphasize an intuitive interface and minimal user

interaction. Maintainability is achieved through modular architecture and well-documented code, allowing future enhancements. Portability ensures compatibility with standard operating systems and browsers, while data integrity guarantees consistent storage of facial information and logs. Finally, interoperability allows integration with external databases and other biometric systems, ensuring flexibility for future expansion.

DESIGN ENGINEERING

Design engineering plays a vital role in representing the structure, behavior, and interaction of system components through Unified Modeling Language (UML) diagrams. It acts as a blueprint that assists in visualizing, specifying, constructing, and documenting the proposed software solution. For the Real-Time Face Recognition–Based User Authentication System, design engineering is used to model interactions among system entities including users, the web application, the OpenCV-based face recognition module, application server, and database. These components collectively enable secure and real-time authentication.

The proposed system consists of multiple entities such as the User, Administrator, Webcam Interface, Face Capture Module, Face Recognition Engine, Application Server, and Database Server. These entities communicate through structured workflows that include live face capture, feature extraction, similarity comparison, and secure data retrieval. To represent these interactions effectively, various UML diagrams are utilized.

Use Case Diagram

The Use Case Diagram describes interactions between system actors and functionalities. The main actors in the system are the User and the Administrator. The user performs operations such as registration, live face capture, facial login, and secure access to personal information. The administrator supervises user activities, manages records, and evaluates system performance. This diagram clearly outlines system boundaries and functional responsibilities.

Class Diagram

The Class Diagram represents the static structure of the system by defining classes and their relationships. Major classes include User, Admin, FaceCapture, FaceRecognizer, FacialFeatures, AuthenticationManager, and DatabaseHandler. These classes contain attributes and methods responsible for storing user information, capturing images, extracting features, matching faces, and handling database operations. The diagram provides a clear understanding of object relationships and modular system design.

Object Diagram

The Object Diagram presents a runtime snapshot of the system showing instances of classes. Objects such as User1, FaceCaptureObj, FaceRecognizerObj, AuthManagerObj, and DatabaseObj illustrate how data is handled during authentication. It demonstrates actual values including user ID, captured image location, session status, and authentication results. This diagram helps visualize object interactions in memory.

State Chart Diagram

The State Chart Diagram models various states in the authentication lifecycle. A user transitions through states such as Idle, Registration Initiated, Face Captured, Face Processing, Authentication Successful, Authentication Failed, and Session Active. These transitions occur due to events like successful capture or matching results. This diagram ensures consistent authentication behavior.

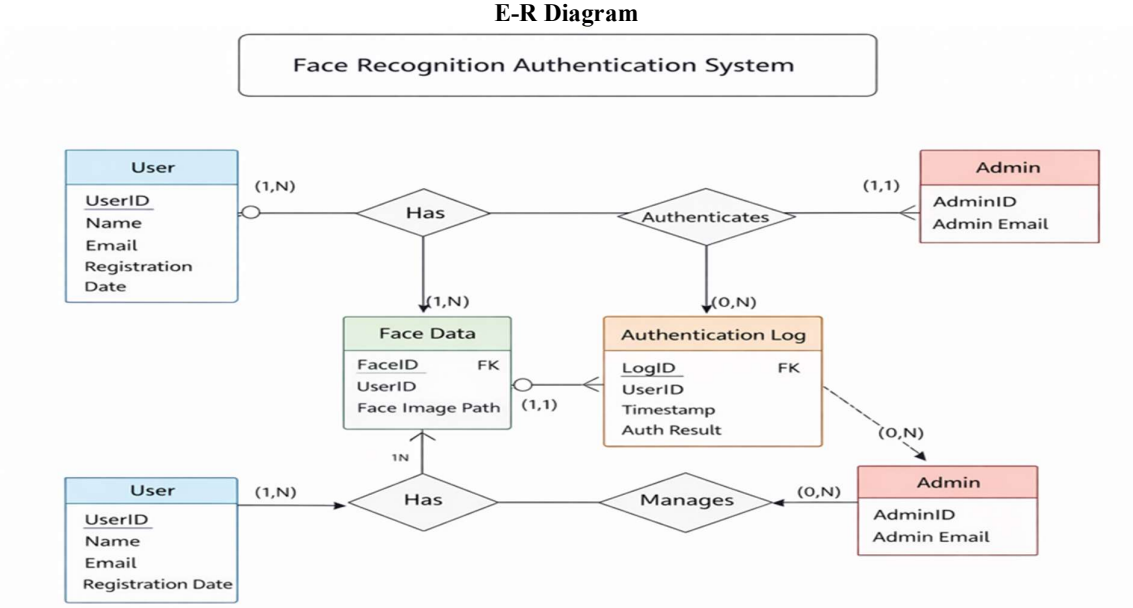
Sequence Diagram

The Sequence Diagram illustrates the chronological message flow between system components. The process begins with user input, followed by webcam capture, OpenCV processing, feature matching, and database verification. This diagram highlights real-time interaction among modules.

Collaboration Diagram

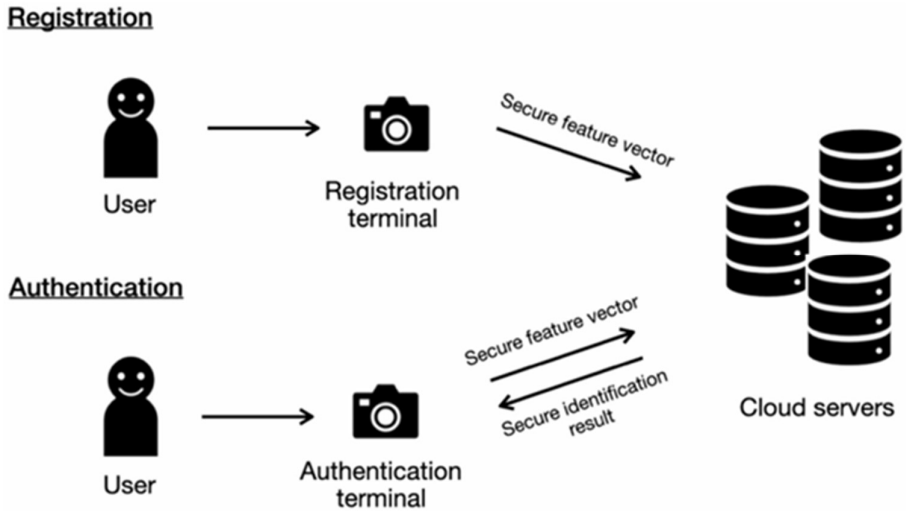
The Collaboration Diagram focuses on object interactions required to perform authentication. Objects including User, Webcam, FaceCapture, FaceRecognizer, AuthenticationManager, and DatabaseHandler collaborate to complete registration and login operations. The diagram emphasizes structural relationships between objects.

E-R Diagram



The ER diagram illustrates the logical database structure. The primary entity is User, which contains attributes such as user ID, name, email, and registration details. The Face Data entity is linked to the User entity through a one-to-many relationship, allowing multiple facial records per user. This ensures secure linkage between biometric data and user profiles.

System Architecture



The proposed system architecture adopts a two-stage filtering and matching approach to improve authentication efficiency. In large-scale biometric systems, comparing a captured feature vector with all stored vectors leads to high computational cost. To reduce this overhead, the system first generates a reduced candidate set before performing detailed similarity computation.

In the first stage, extracted feature vectors are transformed using a binarization process. Each feature value is converted into a binary representation using a sign function. Positive values are mapped to one, while negative values are mapped to zero. This conversion simplifies the comparison process and enables the use of Hamming distance. The Hamming distance between the captured vector and stored vectors is calculated to determine similarity. Since this operation is computationally lightweight, candidate selection is performed quickly.Only the top k vectors with the smallest Hamming distance are selected as

candidate matches. This significantly reduces the number of comparisons required in the second stage. In the final stage, cosine similarity is computed between the captured feature vector and candidate vectors. The vector with the highest similarity score is selected as the closest match.This architecture improves performance while maintaining accuracy. Experimental evaluation indicates that filtering reduces the candidate set to approximately 0.5% of the total dataset while preserving identification accuracy close to 94%. The binarization process also enhances privacy by partially obfuscating facial features. Only Hamming distance values are revealed, while actual feature vectors remain protected. Cosine similarity computations in the final stage remain fully secure.

The proposed architecture therefore provides faster authentication, improved scalability, and enhanced privacy protection. By combining binarization-based filtering with cosine similarity matching, the system achieves efficient and secure real-time face recognition suitable for modern web-based applications.

DEVELOPMENT TOOLS

This chapter describes the programming languages, frameworks, and tools used for implementing the proposed Real-Time Face Recognition–Based User Authentication System. The development platform selected for this project is Java, which provides portability, scalability, and strong support for web-based applications. The system is implemented using Java technologies including Core Java and J2EE for building server-side components and integrating database operations. J2EE is chosen due to its capability to support distributed, multi-tier web applications with enhanced security and performance. Additionally, Java-based frameworks enable seamless integration with OpenCV libraries for face recognition functionality and database connectivity.

Features of Java

Java Framework

Java is a high-level programming language developed by James Gosling at Sun Microsystems and released in 1995. It is designed to be platform-independent, allowing applications to run on any system that supports the Java Virtual Machine (JVM). Java syntax is influenced by C and C++, but it simplifies memory management and removes low-level programming complexities. Java programs are compiled into bytecode, which is interpreted by the JVM, enabling the "write once, run anywhere" principle.Java is object-oriented, concurrent, secure, and robust, making it suitable for network-based and enterprise applications. It supports multithreading, exception handling, automatic memory management, and strong security features. These capabilities make Java widely used in web applications, enterprise systems, mobile platforms, and distributed computing environments. Its portability and reliability make it an ideal choice for implementing real-time authentication systems.

Multithreading in Java

A thread represents an independent execution path within a program. Multithreading allows multiple tasks to run concurrently, improving application performance. In authentication systems, multithreading enables simultaneous face capture, processing, and database operations.

Threads in Java can be created by extending the Thread class or implementing the Runnable interface. Implementing interfaces is preferred as it allows classes to inherit from other classes while still supporting multithreading. Java's thread management enhances responsiveness and efficiency in real-time systems.

Advanced Java Technologies

Advanced Java includes libraries and APIs used for developing enterprise and web-based applications.

Servlets: Server-side Java programs that handle HTTP requests and responses. They act as controllers in web applications.

JavaServer Pages (JSP): Technology that allows embedding Java code in HTML for generating dynamic web pages. JSP is compiled into servlets during execution.

JDBC: Java Database Connectivity provides an API for interacting with relational databases. It enables execution of SQL queries and data retrieval.

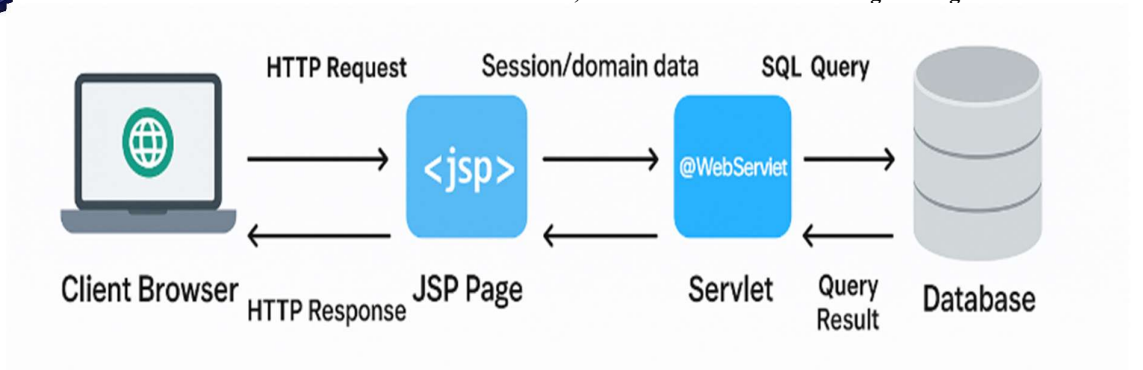
JavaBeans: Reusable components used to encapsulate data and logic. They support modular application design.

MVC Architecture: Model-View-Controller separates business logic, user interface, and control flow. Servlets act as controllers, JSP as views, and Java classes as models.

Filters: Used for request preprocessing, such as authentication and logging.

Listeners: Handle application lifecycle events like session creation and termination.

Session Management: Maintains user state using cookies, sessions, or URL rewriting.



SOFTWARE TESTING

Software testing is an essential phase in the software development life cycle aimed at identifying errors and ensuring that the developed system meets functional and performance requirements. Testing involves executing the software with the intention of detecting faults, verifying expected behavior, and validating system reliability. It evaluates whether the application satisfies user expectations and performs correctly under different operating conditions.

In the proposed Real-Time Face Recognition–Based User Authentication System, testing ensures that modules such as face capture, feature extraction, database interaction, and authentication logic operate accurately and efficiently. Various testing techniques are applied to validate individual components as well as the complete integrated system. Each testing type focuses on a specific aspect of functionality and performance.

Developing Methodologies

The testing process begins with the preparation of a structured test plan that evaluates core functionalities and advanced features across different operating environments. The methodology focuses on verifying system requirements, detecting defects, and ensuring stable operation. Quality assurance procedures are followed to confirm that the implemented system performs as specified in the requirements document.

The testing strategy includes validating individual modules, checking module integration, analyzing system performance, and confirming user acceptance. These steps ensure that the application operates reliably and supports secure face-based authentication.

Types of Tests

Unit Testing

Unit testing is performed to verify the correctness of individual components. Each module is tested independently to ensure that internal logic produces expected results. This testing validates decision branches, input handling, and output generation.

In the proposed system, unit testing is conducted for modules such as face capture, feature extraction, database storage, and authentication manager. Each module is tested using predefined inputs, and outputs are compared with expected results. This process helps detect errors early and improves code reliability.

CONCLUSION

This study introduced novel privacy-preserving 1:N face identification protocols constructed entirely using secure multi-party computation (MPC). The proposed approach demonstrates that robust privacy protection can be achieved without sacrificing computational efficiency. The first protocol ensures complete confidentiality of facial biometric data and performs effectively in environments with small to medium-sized databases. The second protocol intentionally allows minimal controlled information leakage to significantly improve scalability, enabling real-time identification even when the system handles up to 30,000 registered users.

By incorporating live face capture and recognition through OpenCV, the developed system addresses limitations found in traditional authentication mechanisms that depend on static passwords, tokens, or pre-stored images. Unlike earlier solutions that often lack automation, real-time capability, or formal privacy guarantees, the proposed framework supports dynamic face acquisition, secure matching, and immediate retrieval of associated user information.

Experimental evaluations and comparisons with existing MPC- and homomorphic encryption-based approaches demonstrate that the proposed protocols provide an improved balance among privacy preservation, computational performance, and usability. These characteristics make the system suitable for deployment in practical, large-scale authentication environments where both security and responsiveness are critical. Overall, the work confirms that privacy-preserving face identification can be implemented efficiently while maintaining strong theoretical guarantees and operational feasibility.

REFERENCES

[1] "Premier League attendance statistics," 2025. [Online]. Available: https://www.transfermarkt.co.uk/premier-league/besucherzahlen/wettbewerb/GB1/plus/?saison_id=2023

[2] —

[3] J. Bai, X. Zhang, X. Song, H. Shao, Q. Wang, S. Cui, and G. Russello, "CryptoMask: Privacy-preserving face recognition," in *Proc. Int. Conf. Information and Communication Security*, Singapore, 2023, pp. 333–350.

[4] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proc. Advances in Cryptology*, Heidelberg, Germany, 1992, pp. 420–432.

[5] J. C. Bernal-Romero, J. M. Ramirez-Cortes, J. D. J. Rangel-Magdaleno, P. Gomez-Gil, H. Peregrina-Barreto, and I. Cruz-Vega, "A review on protection and cancelable techniques in biometric systems," *IEEE Access*, vol. 11, pp. 8531–8568, 2023.

[6] J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, Mar. 2013.

[7] O. Catrina and S. de Hoogh, "Secure multiparty linear programming using fixed-point arithmetic," in *Proc. European Symposium on Research in Computer Security*, Berlin, Germany, 2010, pp. 134–150.

[8] M. A. P. Chamikara, P. Bertok, I. Khalil, D. Liu, and S. Camtepe, "Privacy-preserving face recognition utilizing differential privacy," *Computers & Security*, vol. 97, 2020, Art. no. 101951.

[9] H. Chun, Y. Elmehdwi, F. Li, P. Bhattacharya, and W. Jiang, "Outsourceable two-party privacy-preserving biometric authentication," in *Proc. 9th ACM Symposium on Information, Computer and Communications Security*, 2014, pp. 401–412.

[10] T. Cilloni, W. Wang, C. Walter, and C. Fleming, "Ulixes: Facial recognition privacy with adversarial machine learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 1, pp. 148–165, 2022.

[11] R. Cramer, I. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. New York, NY, USA: Cambridge University Press, 2015.

[12] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for online auctions," in *Proc. Australasian Conference on Information Security and Privacy*, Heidelberg, Germany, 2007, pp. 416–430.

[13] A. Das, M. Degeling, X. Wang, J. Wang, N. Sadeh, and M. Satyanarayanan, "Assisting users in a world full of cameras: A privacy-aware infrastructure for computer vision applications," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1387–1396.

[14] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.