Biometric-Based Encryption System to Enhance Cloud Data Security through the Integration of Facial Recognition and Homomorphic Encryption

Omeje, K. N.[1] and Asogwa, T. C.[2]

[1]Department of computer Science, David Umahi Federal University of Health Sciences

[2]Department of Computer Science, Enugu State University of Science and Technology.

DOI: https://dx.doi.org/10.51584/IJRIAS.2026.110100128

Received: 07 February 2025; Accepted: 12 February 2026; Published: 19 February 2026

ABSTRACT

The rapid growth of cloud computing has introduced significant benefits in terms of data storage and processing but it has also increased the risks of unauthorized access and data breaches. Therefore, this study presents a biometric-based encryption system which is designed to enhance cloud data security through the integration of facial recognition and homomorphic encryption. The proposed system employs an Autoencoder (AE) for feature extraction, Convolutional Neural Network (CNN) for facial recognition, and the Brakerski-Gentry-Vaikuntanathan (BGV) algorithm for secure data encryption and decryption. The adopted AE is used to efficiently compresses facial features into latent vectors used both for recognition and as encryption keys. Furthermore, the experimental evaluation of the techniques adopted using both primary facial datasets and the LFW dataset demonstrated that the AE achieved a training accuracy of 99.84% and validation accuracy of 98.59%, while the CNN attained a training accuracy of 97.05% and validation accuracy of 95.04%. Additionally, the result of the BGV encryption process recorded an average encryption time of 0.023 seconds and decryption time of 0.019 seconds, indicating minimal computational overhead. Results confirm that the integration of biometric encryption enhances both data confidentiality and authentication reliability in cloud environments. This system provides a robust and efficient framework for securing sensitive data in modern cloud infrastructures, ensuring privacy, integrity, and accessibility for authorized users.

Keywords: Biometric Encryption; Cloud Security; Facial Recognition; AE; CNN

INTRODUCTION

The rapid advancement of cloud computing has revolutionized data storage, processing, and sharing, offering organizations and individuals scalable, flexible, and cost-effective solutions. However, these benefits come with significant security challenges, as cloud networks are frequently targeted by cyberattacks, data breaches, and unauthorized access (Rahman *et al.* 2024). Ensuring secure access to data within cloud networks requires robust authentication and encryption mechanisms. The traditional security measures, such as passwords and two-factor authentication, have proven to be vulnerable, leading to an increasing demand for more secure and reliable solutions like biometric encryption technology (Lu and Zhao, 2023).

Biometric encryption technology combines the use of biometric data (unique biological traits such as fingerprints, iris patterns) and facial recognition with encryption methods to enhance the security of authentication systems (Babu, 2023). Unlike conventional methods, where passwords or tokens can be stolen or compromised, biometric data provides a highly personalized form of authentication that is difficult to replicate (Bagga *et al.* 2022). In the context of cloud networks, where data accessibility and security are of paramount concern, integrating biometric encryption can significantly reduce the risks associated with unauthorized access to sensitive data (Shruti *et al.* 2024).

The concept of encryption in the security domain involves encoding information in such a way that only authorized individuals can access it (Helmy *et al.* 2023). When biometric data is encrypted, it adds an extra layer of protection, ensuring that even if the data is intercepted, it remains inaccessible without proper biometric verification (Bagga *et al.* 2022). This fusion of biometrics and encryption creates a strong safeguard against common cyber threats such as identity theft, hacking, and data breaches. Moreover, biometric encryption offers the advantage of eliminating the need for users to remember complex passwords, thereby enhancing the user experience while maintaining security (Bagga *et al.* 2022).

However, the adoption of biometric encryption in cloud networks is not without its challenges. Storing and processing biometric data raises privacy concerns, as the misuse or theft of such data could lead to significant personal and security risks (Mostafa *et al*. 2023). Furthermore, the computational demands of encrypting and decrypting biometric information in real-time can affect the performance of cloud-based systems. Despite these challenges, biometric encryption remains a promising solution to address the evolving security needs of cloud networks, especially as cyber threats become more sophisticated (Umar *et al*. 2024).

Overall, the integration of biometric encryption technology into cloud network security represents a forward-thinking approach to addressing modern security challenges. This study seeks to explore the potential of biometric encryption in securing cloud environments, examining both its benefits and limitations. As organizations continue to adopt cloud solutions, the need for advanced security mechanisms, such as biometric encryption, becomes increasingly critical in ensuring the confidentiality, integrity, and accessibility of sensitive data.

METHODOLOGY

The methodology used for this work is Agile and Object-Oriented Analysis and Software Design Methodology. Agile was chosen because it supports flexible, iterative development, allowing for continuous adaptation based on feedback and ongoing testing ideal for a security-focused project like a biometric encryption system for cloud security. With Agile, each phase of the system’s development, from user registration to encryption key generation, can be developed, tested, and improved in iterative sprints.

This approach helps quickly address any issues with biometric data handling, encryption processes, or cloud integration, ensuring the system remains secure and responsive to user requirements. Agile also allows for incremental releases, which means that core features like facial recognition or access control can be launched and validated early in the process, helping to identify and mitigate risks faster.

The proposed system will operate in the following steps: First, User Registration captures and securely stores facial data from authorized users, creating unique biometric profiles. When a user attempts access, Facial Recognition and Feature Extraction analyze live facial data, producing a distinct biometric signature. This signature serves as the basis for Encryption Key Generation, creating a dynamic encryption key unique to each user. Data is encrypted with this key and stored in the cloud.

When a registered user requests access, the system performs Face Matching and Verification against stored profiles. A successful match triggers Decryption using the biometric key, granting access to the user while preventing unauthorized attempts. This ensures that only validated users with matching biometrics can access data, securing privacy and data integrity in the cloud environment.

Data collection

The data used for the work are primary and secondary data of faces. The primary data contains 20 images of the admin user, collected from the self-volunteered individuals. The secondary data is the LFW (Labelled Faces of the Wild) dataset of faces collected from Kaggle.

The data sample size is 13,000 images of faces labelled with the name of the person. The data source is https://www.kaggle.com/datasets/atulanandjha/lfwpeople. Each picture is centered on a single face. Each pixel of each channel (color in RGB) is encoded by a float in range 0.0 - 1.0. The original images are 250 x 250 pixels. Another secondary data used for this work are secured personal information of the volunteered users which were synthetically generated to protect their privacy. The sample data was reported in Table 1.

Table 1: Synthetic Data for Biometric Encryption Testing

User ID	Name	Face Image ID	BVN	NIN	ATM Number	Timestamp
U001	Chinedu Okoro	IMG001.jpg	BVN-2233445566	NIN-74583910285	4536-XXXX-XXXX-1280	2025-07-24 10:12:35
U002	Aisha Bello	IMG002.jpg	BVN-1098765432	NIN-88374592037	5521-XXXX-XXXX-9954	2025-07-24 10:15:18
U003	Emeka Onyekachi	IMG003.jpg	BVN-3322110099	NIN-66849273018	4263-XXXX-XXXX-7641	2025-07-24 10:16:52
U004	Zainab Yusuf	IMG004.jpg	BVN-5566778899	NIN-90237461539	4539-XXXX-XXXX-1002	2025-07-24 10:19:20
U005	John Adebayo	IMG005.jpg	BVN-1144556688	NIN-74839020175	5412-XXXX-XXXX-5876	2025-07-24 10:20:49

Develop biometric feature extraction techniques using auto encoder

The deep learning extractor of the data applied for this work is Autoencoder (AE). The system is made of four main components which are the input layer, encoder layer, decoder and the feature extraction phase. The autoencoder takes a face image as input and passes it through the encoder, which compresses the image step-by-step using convolutional and pooling layers.

This results in a compact feature vector of the face. Then the decoder tries to reconstruct the original face image from this compressed vector. During training, the model learns to retain only the most important features of the face (like the eyes, nose, mouth spacing) while discarding irrelevant details. This training is necessary as it allows the AE know the features to extract. Once trained, the decoder is discarded and uses just the encoder to extract face features for training CNN. The stepwise of the AE is presented below.

Stepwise of the AE face extractor

Step 1: Input Image

Start with a 250×250 grayscale image of a person's face.

Normalize pixel values between 0 and 1.

Step 2: Encoding Phase

Pass the image through convolutional layers to detect patterns (edges, shapes).

Use max-pooling layers to reduce image size and keep important info.

Flatten the data and compress it into a dense feature vector

This vector is the face embedding

Step 3: Decoding Phase

The compressed vector is passed through upsampling and convolution layers.

The network reconstructs the original image as closely as possible.

The loss (difference between input and reconstructed image) is minimized.

Step 4: Feature Extraction

After training, use only the encoder part.

Feed a new face image → get the compressed feature vector.

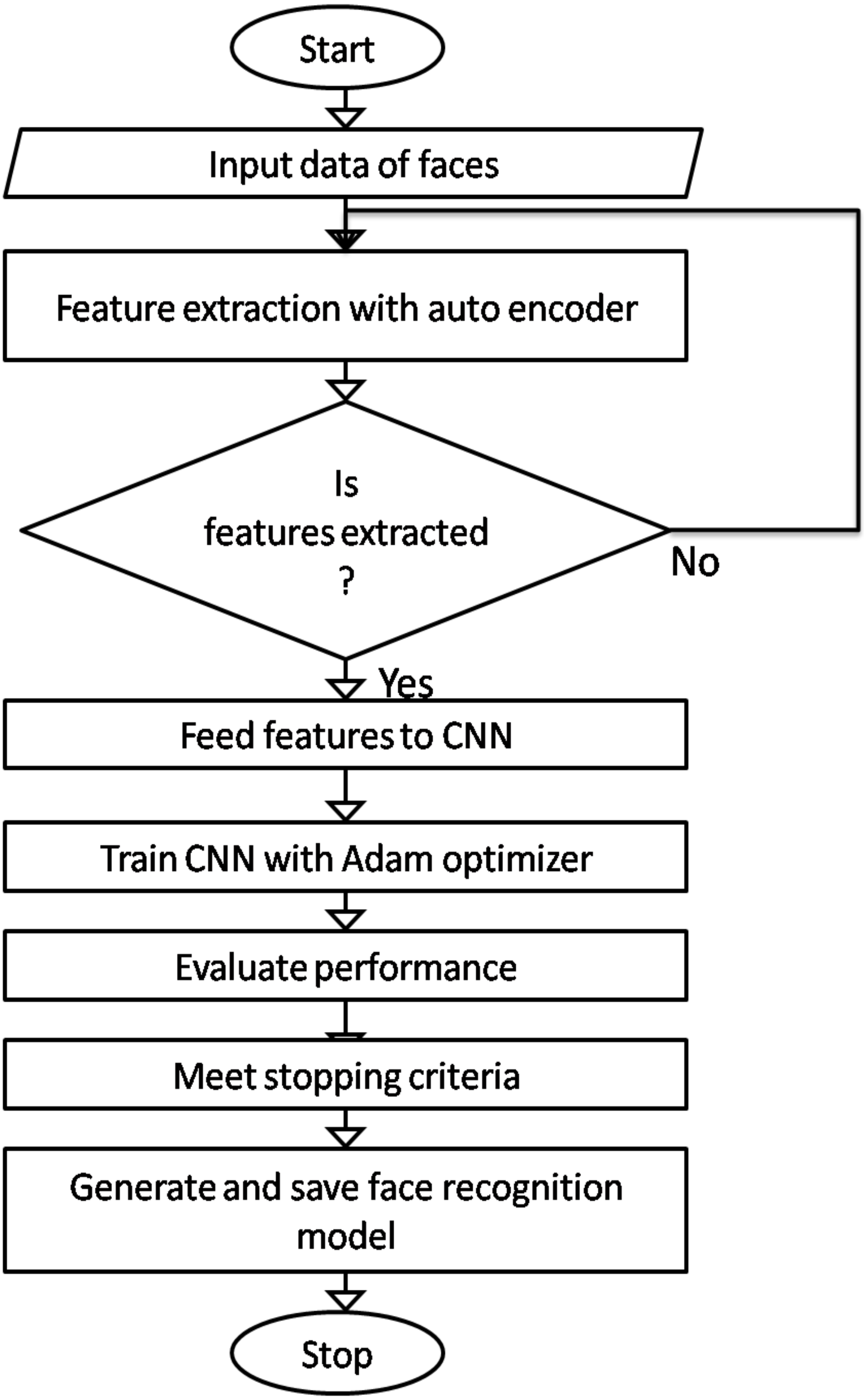
Use this for biometric matching, encryption, or classification.

End

Develop biometric authentication model through facial recognition

This section developed a biometric facial recognition model with Convolutional Neural Network (CNN) as the deep learning adopted. The data extracted from the AE is applied to train the CNN and then generate a facial recognition model. The CNN is made of input, convolutional, fully connected layer and output layer. The flowchart of the CNN operation is presented as Figure 1;

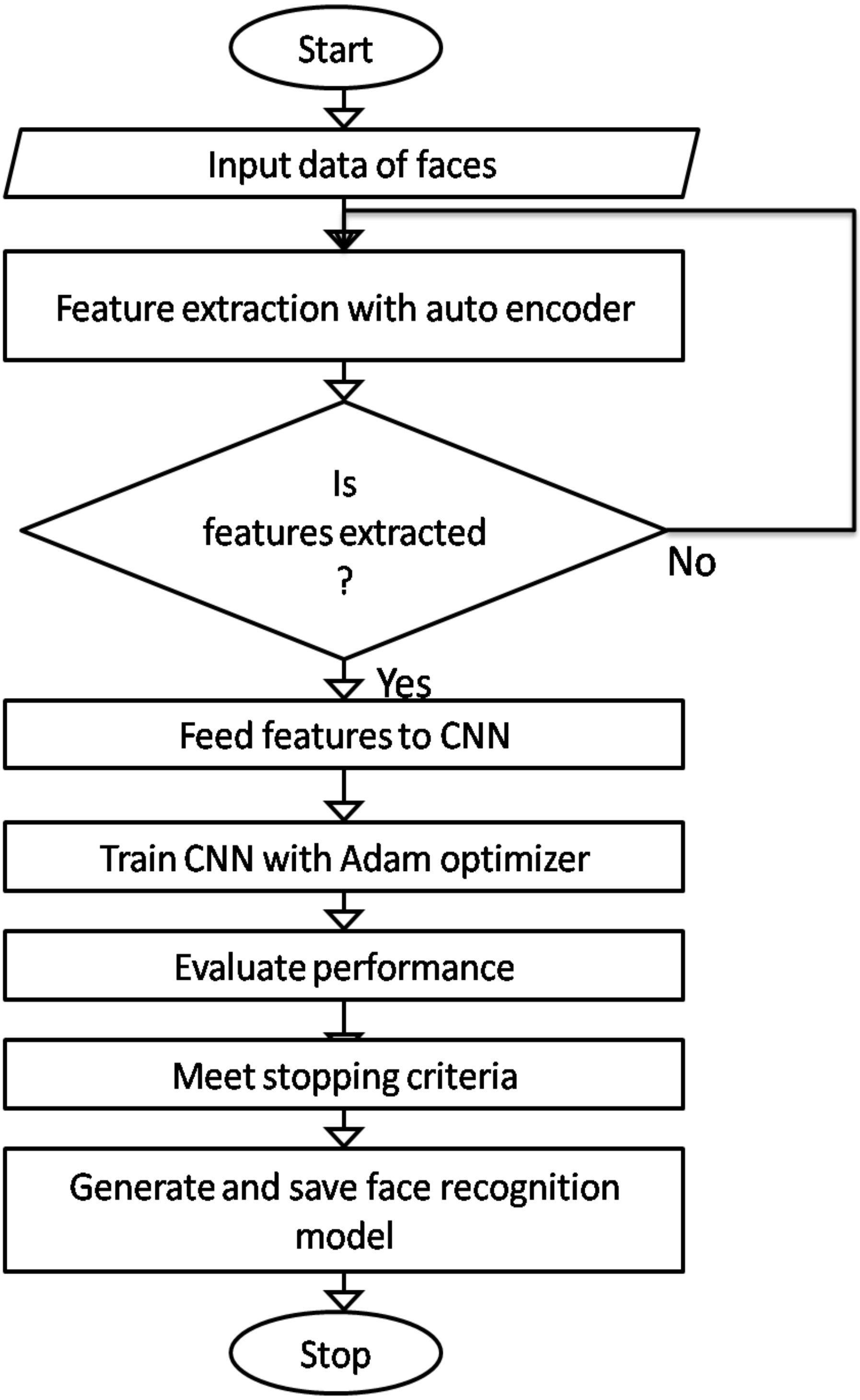
Figure 1: Flowchart of the CNN



CNN Training as Face Recognition Model

The CNN is trained as a supervised learning model to perform face recognition by classifying individuals based on their facial features. After extracting compact and relevant features from the face images using a trained Autoencoder, these features are fed into the CNN model along with their corresponding user identities. During training, the CNN learns to map the input feature vectors to the correct identities by minimizing a classification loss function, typically using SoftMax activation in the final layer. Figure 2 presents the CNN training flowchart.

Figure 2: CNN training flowchart



The CNN as shown in Figure 2 includes multiple convolutional layers to detect hierarchical facial patterns such as edges, eyes, and facial contours, followed by pooling layers to reduce dimensionality. Fully connected layers at the end of the network combine the learned features and pass them to a classifier layer to output the predicted class. The model is trained over several epochs using an optimizer (Adam), and the weights are updated via back-propagation to improve accuracy. Once the CNN achieves satisfactory accuracy, it becomes a reliable face recognition model capable of identifying individuals from input images. The final trained model is then encrypted using a biometric encryption algorithm to ensure secure deployment on the cloud, protecting both the model and user identity data.

Propose biometric Data Encryption Algorithm

The cryptographic algorithm used in this work is the Brakerski-Gentry-Vaikuntanathan (BGV) Algorithm, as described by Huang and Wu (2022). It is a fully homomorphic encryption (FHE) algorithm, which means it allows computations to be performed directly on encrypted data without needing to decrypt it first. The BGV algorithm works based on the Learning with Errors (LWE) principle and uses polynomial rings to represent both the original (plaintext) and encrypted (cipherfeatures) data. This structure enables secure addition and multiplication of encrypted values. The BGV process operates in three main phases: key generation, encryption, and decryption. The key generation phase is defined by four steps which are presented as Equation 1 (Weir, 2013; Huang and Wu, 2022)

$$\varepsilon = (KeyGen, Encrypt, Decrypt, Evaluate) \qquad (1)$$

Where the KeyGen produced the key pairs as(□□, □□), the encryption key used public key to encryption plain features, decryption used secrete key to recover features and the evaluate stage used Boolean function to the cipher features. Table 2 presents parameter description.

Table 2: Parameter description (Huang and Wu, 2022)

Parameter	**Description**
λ	Security level (128-bit)
n	Degree of the polynomial ring
q	Cipherfeatures modulus
p	Plaintext modulus
χ	Noise distribution
s	Secret key vector
A, B	Public key matrices
r, e	Random encryption vectors/noise

Key generation algorithm

This algorithm has an input as the security parameter λ ∈ Z, a number representing the security level (128, 192, 256). The next input is the polynomial ring parameters (degree n, cipher features, q). a noise distribution factor X over the ring A, commonly a discrete Gaussian. From this input, the output is public and secrete key (pk, sk). The sk which is the secrete key contain randomly small norm vector s, and the public key generated using matrix A and a noise vector e and is defined in Equation 2 (Crawford, 2019).

$$Pk = (A, A.s + e)\ mod\ q \qquad (2)$$

This key generation ensures that the hardness of recovering s from the public key reduces to the Ring-LWE problem.

The Encryption

This is applied to encrypt the feature maps (m). To achieve this, the random vector r and noise vector e1, e2 are applied. Then a message polynomial m(x) encoded in the features ring is constructed. The cipher features ct = (c1, c2) as in Equation 3 and 4 (Wang et al., 2004; Huang and Wu, 2022);

$$c_1 = A^t \cdot r + e_1 + m \cdot \lfloor q/p \rfloor \qquad (3)$$

$$c_2 = B^t \cdot r + e_2 \qquad (4)$$

Where q is the cipher features modulus and p is the plaintext modulus. The cipher features is thus a pair of polynomials that hide the message under both encryption randomness and noise.

Decryption

To decrypt the cipher features in Equation 3 and 4, the secret keys was used, the inner produce is defined as Equation 5, while to reduce the modulo (p) and retrieve normal features m is determined with Equation 6

(Bakurov et al., 2022; Huang and Wu, 2022);

Compute $c = c_1 - \ s.c_2)\ mod\ q$ (5)

$$m \ \approx (c_1 - s.c_2) mod\ q\ mod\ p \qquad (6)$$

Decryption succeeds as long as the accumulated noise is below a predefined threshold.

The Evaluation process

The Evaluate process takes a function (arithmetic circuit C) and a set of input cipher features, and performs homomorphic operations in Equation 4.7 (Huang and Wu, 2022);

Decrypt (Evaluate (C, Encrypt(m_1), ..., Encrypt(m))) = C(m_1, ..., m) (7)

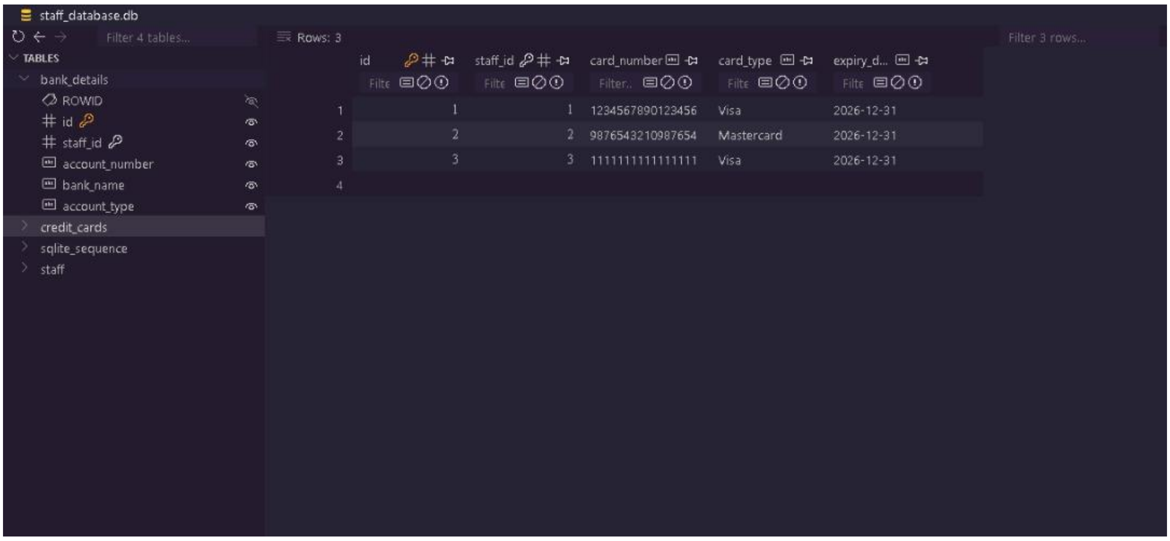
This step supports addition, multiplication, and general function application, enabling secure outsourced computation in the cloud environment.

BGV encryption algorithm (Huang and Wu, 2022)
Key generation %% generate public and secret keys
Encryption key %% public key
Secrete key %% decryption
Encryption %% convert features to cipher features with public key
Decryption %% apply for facial recognition
End

Implementation

The implementation of the biometric cloud data encryption system was carried out in three integrated modules: feature extraction using an Autoencoder (AE), face recognition using a Convolutional Neural Network (CNN), and biometric-bound encryption and decryption using the Brakerski-Gentry-Vaikuntanathan (BGV) homomorphic encryption algorithm.

The system was developed using Python programming language and implemented in Google Colab due to its flexibility, GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) support, and compatibility with machine learning libraries such as TensorFlow, Keras, and PyCryptodome. Figure 3 presents the programming interface.



	id	staff_id	card_number	card_type	expiry_d...
1	1	1	1234567890123456	Visa	2026-12-31
2	2	2	9876543210987654	Mastercard	2026-12-31
3	3	3	1111111111111111	Visa	2026-12-31
4					

Figure 3: The coding environment

The first step involved is building and training an Autoencoder model to extract compressed facial features from input face images. The encoder part of the AE learns to capture latent feature representations of the face, which are compact but retain enough identity-specific information. These features, in matrix form, serve two purposes: they are used to train the CNN model for identity recognition, and they are further passed into the encryption phase as the core biometric key material. In the second phase, the CNN model was trained using a labelled dataset of facial images to classify and recognize individuals. The CNN architecture included convolutional layers, ReLU activations, pooling layers, and fully connected layers. It was trained to learn high-level feature maps of the face, allowing the model to generalize across variations in lighting, orientation, and facial expressions.

Upon testing, the CNN achieved satisfactory performance in terms of accuracy and recognition speed. The third phase integrated the BGV homomorphic encryption scheme, where the facial features served as dynamic encryption keys or inputs for key generation. The encryption process involved key generation, feature-based encryption of sensitive data, and secure storage in a simulated cloud environment. For decryption, the same biometric had to be presented; the decrypted output matched the original data only if the features were authenticated successfully. This ensures that the data remains secure even if intercepted in an untrusted server. Performance evaluations were conducted at each stage to assess feature compression efficiency, CNN recognition accuracy, and encryption/decryption time.

SYSTEM RESULTS

The testing on the system involved the training result of the CNN, AE, encryption algorithm and then experimental validation of the system in real world scenario.

Result of the AE and training

This section presents the result of the AE training. The performance was evaluated with accuracy and loss. The reason for the training is to allow the AE encoder to understand and learn high level features which are necessary for optimal feature extraction process. The training results are reported across 10 epochs and reported in Table 3.

Table 3: Training result of the AE

Epoch	Accuracy	Loss	Val_Accuracy	Val_Loss
1	0.9137	0.2883	0.9783	0.0661
2	0.9847	0.0521	0.9823	0.0512

3	0.9902	0.0326	0.9852	0.0453
4	0.9941	0.0191	0.9855	0.0427
5	0.9959	0.0134	0.9825	0.0555
6	0.9962	0.0107	0.9822	0.0650
7	0.9976	0.0079	0.9853	0.0521
8	0.9972	0.0081	0.9856	0.0576
9	0.9980	0.0054	0.9842	0.0657
10	0.9984	0.0051	0.9859	0.0673

Table 3 summarizes the training results of the AE used for feature extraction in the facial recognition model. During training, the AE achieved a remarkably high accuracy of 99.84% (0.9984) and a very low training loss of 0.0051, indicating that the model learned the underlying patterns in the facial data very effectively. On the validation set, the model maintained a strong performance with an accuracy of 98.59% (0.9859) and a loss of 0.0673.

These results demonstrate that the AE model is highly effective at capturing and learning meaningful facial features. The minimal difference between training and validation metrics suggests that the model is generalizing well and not overfitting the training data. Therefore, the AE can reliably extract high-level and distinctive facial features, which are crucial for accurate and efficient facial recognition in real-world scenarios.

Training of the CNN

This section presents the training process of the CNN model used for facial recognition. The CNN was designed to learn complex patterns and unique features from facial images, allowing it to accurately classify and recognize individual faces. After preprocessing and feature extraction using the AE, the CNN model was trained on the extracted features to further refine and enhance recognition performance. The training process involved feeding the model with labelled image data, adjusting weights through back propagation, and evaluating its performance using loss and accuracy metrics. Table 4 presents the CNN training results.

Table 4: CNN training result

Epoch	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy
1	0.1942	0.1932	0.939777	0.902175
2	0.1931	0.1932	0.954768	0.916849
3	0.1931	0.1932	0.925795	0.93081
4	0.1930	0.1932	0.927815	0.927912
5	0.1930	0.1932	0.944756	0.904864
6	0.1929	0.1932	0.942739	0.912512
7	0.1928	0.1932	0.955992	0.933655
8	0.1927	0.1932	0.94497	0.916262
9	0.1926	0.1933	0.937143	0.92568
10	0.1926	0.1931	0.948248	0.935266
11	0.1925	0.1931	0.955034	0.921922
12	0.1924	0.1931	0.953757	0.934499
13	0.1924	0.1931	0.95223	0.921944

14	0.1924	0.1931	0.959429	0.93111
15	0.1923	0.1931	0.962061	0.942161
16	0.1922	0.1931	0.965795	0.924873
17	0.1922	0.1931	0.960066	0.947047
18	0.1921	0.1931	0.953513	0.944285
19	0.1921	0.1930	0.958256	0.951846
20	0.1920	0.1931	0.97046	0.9504

Table 4 presents the performance results of the CNN trained for facial recognition. The model achieved a training accuracy of 97.05% (0.9705) and a validation accuracy of 95.04% (0.9504). Correspondingly, the training loss was 0.1920, while the validation loss was slightly higher at 0.1931. These results indicate that the CNN was able to effectively learn the relationship between the input features (facial characteristics) and their corresponding labels (identities). The small gap between training and validation performance suggests the model has good generalization ability, meaning it performs well on unseen data. The relatively low loss values further support that the CNN model converged successfully and caSn reliably be used for facial recognition tasks.

Result of the BGV Encryption process

This section presents the outcome of implementing the BGV encryption process for securing the facial feature data during cloud transmission and storage. The facial features extracted using the AE are encrypted using the BGV algorithm before being uploaded on the cloud. The encryption process ensures that even if unauthorized access occurs, the feature data remains unintelligible and unusable without the correct decryption keys. Figure 4 presents the encryption and decryption result.

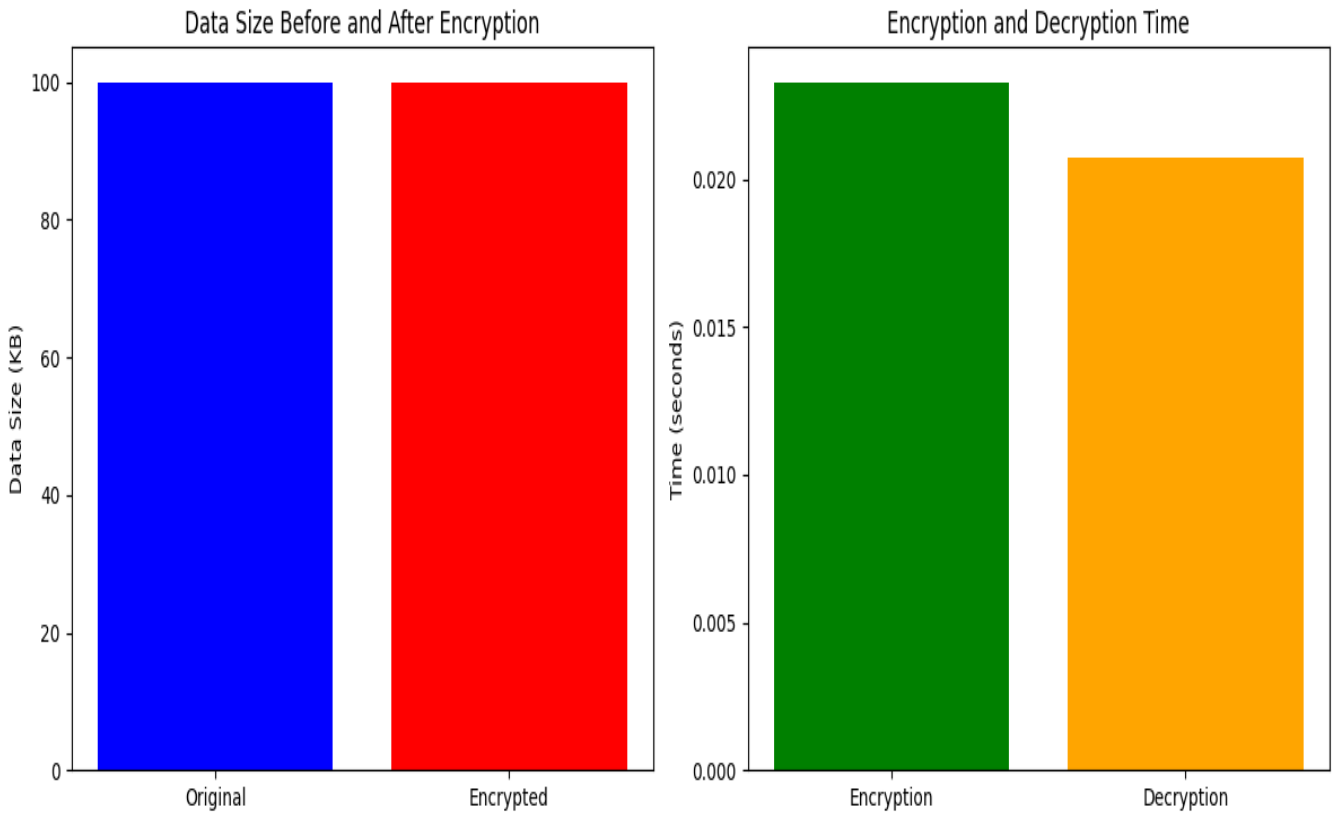


Figure 4: Result of encryption and decryption

The encryption algorithm was evaluated based on key performance parameters, including the data size before and after encryption, as well as the time taken to perform both encryption and decryption processes. The encryption was designed to protect user data through biometric key security, where a unique encryption key was generated using facial features. This key was used to securely encrypt the extracted data. To access the protected data, a decryption key derived from a verified facial signature was required.

The encryption process recorded an average time of 0.023 seconds, while the decryption process took approximately 0.019 seconds. The slightly faster decryption time can be attributed to the fact that encryption involves additional steps such as key generation, feature mapping, and encoding, whereas decryption focuses mainly on reversing those transformations and verifying biometric identity. This efficiency confirms the practicality of the biometric-based encryption scheme in securing cloud data without introducing significant computational overhead.

CONCLUSION

This study developed and implemented a biometric-based facial recognition system integrated with encryption to ensure secure data access in the cloud. The approach combined an AE for feature extraction, a CNN for accurate facial classification, and a BGV encryption scheme to secure biometric templates. The AE demonstrated exceptional performance, achieving a training accuracy of 99.84% and a validation accuracy of 98.59%, indicating its effectiveness in extracting deep facial features. The CNN further reinforced the model's robustness by achieving a training accuracy of 97.05% and validation accuracy of 95.04%, confirming its capability to reliably classify and recognize facial data.

To preserve the privacy and integrity of facial templates, the BGV encryption process was employed. Evaluation showed a minimal overhead, with encryption and decryption times of 0.023s and 0.019s respectively, while maintaining the size integrity of the encoded data. The system effectively ensured that only users with valid facial features could decrypt and access protected data. Overall, the result confirms that the proposed model is effective, reliable, and secures biometric authentication system suitable for modern cloud-based applications where data privacy is paramount.

REFERENCES

1. Babu, P. (2023). *Cloud data security enhancements through the biometric and encryption system* [Unpublished master's thesis]. School of Computing, National College of Ireland.
2. Bagga, P., Mitra, A., Das, A., Vijayakumar, P., Park, Y., & Karuppiah, M. (2022). Secure biometric-based access control scheme for future IoT-enabled cloud-assisted video surveillance system. *Computer Communications, 195*, 27-39. https://doi.org/10.1016/j.comcom.2022.08.003
3. Bakurov, I., Buzzelli, M., Schettini, R., Castelli, M., & Vanneschi, L. (2022). Structural similarity index (SSIM) revisited: A data-driven approach. *Expert Systems with Applications, 189*, Article 116087. https://doi.org/10.1016/j.eswa.2021.116087
4. Crawford, J. L. H. (2019). *Fully homomorphic encryption applications: The strive towards practicality* [Master's thesis, Queen Mary University of London]. Department of Electronic Engineering and Computer Science.
5. Helmy, M., El-Rabaie, E., El-Dokany, I., & El-Samie, F. E. (2023). A novel cancellable biometric recognition system based on Rubik's cube technique for cyber-security applications. *Optik, 285*, Article 170475. https://doi.org/10.1016/j.ijleo.2022.170475
6. Huang, J., & Wu, D. (2022). Cloud storage model based on the BGV fully homomorphic encryption in the blockchain environment. *Security and Communication Networks, 2022*, Article 8541313. https://doi.org/10.1155/2022/8541313
7. Lu, Y., & Zhao, D. (2023). Providing impersonation resistance for biometric-based authentication scheme in mobile cloud computing service. *Computer Communications, 182*, 22-30. https://doi.org/10.1016/j.comcom.2021.10.029
8. Mostafa, A., Ezz, M., Elbashir, M., Alruily, M., Hamouda, E., Alsarhani, M., & Said, W. (2023). Strengthening cloud security: An innovative multi-factor multi-layer authentication framework for cloud user authentication. *Applied Sciences, 13*(19), Article 10871. https://doi.org/10.3390/app131910871
9. Rahman, K., Hridoy, M., Rahman, M., Islam, R., & Banik, S. (2024). Highly secured and effective management of app-based online voting system using RSA encryption and decryption. *Heliyon, 10*(4), Article e25373. https://doi.org/10.1016/j.heliyon.2024.e25373

10. Shruti, Rani, S., & Srivastava, G. (2024). Secure hierarchical fog computing-based architecture for industry 5.0 using an attribute-based encryption scheme. *Expert Systems with Applications, 235*, Article 121180. https://doi.org/10.1016/j.eswa.2023.121180
11. Umar, T., Nadeem, M., & Anwer, F. (2024). Chaos based image encryption scheme to secure sensitive multimedia content in cloud storage. *Expert Systems with Applications, 257*, Article 125050. https://doi.org/10.1016/j.eswa.2024.125050
12. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600-612. https://doi.org/10.1109/TIP.2003.819861
13. Weir, B. (2013). *Homomorphic encryption* [Master's thesis, University of Waterloo].