Iraqi Journal of Science, 2026, Vol. 67, No. 4, pp: 2427- 2439
DOI: 10.24996/ijs.2026.67.4.41

ISSN: 0067-2904

Efficient Searching Technique for Biometric Data Based on Inverted Files

Zahraa Naji Razoqi[1]*, Raheem Ogla[1], Abdul Monem S. Rahma[2]
[1]*Computer Science Department, University of Technology, Baghdad, Iraq.*
[2]*Computer Science Department, College of Science, University of Al-Maarif, Anbar, Iraq.*

Received: 9/12/2024 Accepted: 7/ 4 /2025 Published: 30/4/2026

Abstract

The increase in the amount of biometric data in databases has caused problems for efficient data retrieval and management. Biometric databases face challenges primarily due to the high dimensional nature of biometric features, such as fingerprints, facial recognition data, and iris scans, leading to time consumption in retrieval and matching as well as computationally intensive, and inefficiencies that can hinder real-time applications, especially when using multiple biometric traits. To address these issues, this paper presents a mechanism for searching three-dimensional biometric databases based on inverted files specifically tailored to handle biometric data. As well as designing a 3D database to store data. This method was applied to the proposed multi-biometric recognition system using iris and fingerprint features by storing the biometric data in the database based on identifiers, that are retrieved from the inverted files to locate the vectors of the candidates and retrieve them to perform the matching, thus improve the matching speed without the need to search and match the entire feature vectors in the database. The results showed that less time is consumed in retrieving and matching when applying this technique compared to the traditional method.

Keywords: biometric data, multi-biometric system, inverted files, database indexing, recognition system.

تقنية بحث فعالة للبيانات البيومترية بناءً على الملفات المقلوبة

زهراء ناجي رزوقي [1]*، رحيم عبد الصاحب عكله [1]، عبد المنعم صالح رحمه [2]
[1]قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق.
[2]قسم علوم الحاسوب، كلية العلوم، جامعة المعارف، الأنبار، العراق.

الخلاصة

لقد تسببت الزيادة في كمية البيانات الحيوية في قواعد البيانات في حدوث مشكلات في استرجاع البيانات وإدارتها بكفاءة. تواجه قواعد البيانات الحيوية تحديات كثيرة بسبب الطبيعة عالية الأبعاد للسمات الحيوية، مثل بصمات الأصابع وبيانات التعرف على الوجه والقزحية، مما يؤدي إلى استهلاك الوقت في الاسترجاع والمطابقة بالإضافة إلى العمليات الحسابية المكثفة وعدم الكفاءة التي يمكن أن تعيق التطبيقات في الوقت الفعلي خاصة عند استخدام سمات حيوية متعددة. لمعالجة هذه القضايا، تقدم هذه الورقة آلية للبحث في قواعد البيانات البيومترية ثلاثية الأبعاد بناءً على ملفات مقلوبة مصممة خصيصًا للتعامل مع البيانات البيومترية.

*Email: cs.21.10@grad.uotechnology.edu.iq

بالإضافة إلى تصميم قاعدة بيانات ثلاثية الأبعاد لتخزين البيانات. تم تطبيق هذه الطريقة على نظام التعرف المتعدد البيومتري المقترح باستخدام سمات القزحية وبصمات الأصابع من خلال تخزين البيانات البيومترية في قاعدة البيانات بناءً على المعرفات، والتي يتم استردادها من الملفات المقلوبة لتحديد متجهات المرشحين واسترجاعها لإجراء المطابقة، وبالتالي تحسين سرعة المطابقة دون الحاجة إلى البحث ومطابقة متجهات الميزات بالكامل في قاعدة البيانات. أظهرت النتائج أن الوقت المستغرق في الاسترجاع والمطابقة أقل عند تطبيق هذه التقنية مقارنة بالطريقة التقليدية.

1. Introduction

Biometric characteristics, which include unique physical or behavioral traits such as fingerprints, facial features, iris patterns, and voice prints, are unique to individuals and have a wide range of applications, such as identity verification, commercial, and law enforcement. The uniqueness of biometric data provides a robust mechanism for confirming an individual's identity with a high degree of accuracy, making it a cornerstone of modern security systems [1] [2]. However, the integration of biometric data into practical applications presents significant challenges, particularly concerning data storage, retrieval, and processing efficiency. In practical applications, biometric systems are often required to handle vast amounts of data. For instance, national ID systems, large-scale surveillance, and commercial authentication services can involve millions of individual biometric records. The primary challenge lies in efficiently indexing this high-dimensional data to facilitate rapid and accurate response [3] [4].

Without an effective structure, the retrieval process can become extremely time-consuming, as the system might need to compare a vector against every record in the database, leading to unacceptable delays [5]. To recognize a person, the objective is to narrow down the search space and focus on the most relevant features. Thereby significantly reducing the computational load. This efficiency is crucial not only for enhancing the user experience but also for ensuring the scalability of the biometric system as the volume of data grows [6].

This research aims to address and solve the problems of time-consuming biometric matching and retrieval processes, by designing a technique to speed up the retrieval process based on a specific set of iris features that speed up the retrieval of only candidate IDs from inverted files without searching the entire database and designing a 3D database to store biometric data in the location that determined by the ID of that person.

The rest of this paper is structured as follows: relevant works are included in section 2. The suggested method is explained in section 3. Section 4 explains the metrics that are used to evaluate tasks. The results and discussion are in section 5. Finally, the conclusions and future works are presented in section 6.

2. Related Works

Efficient searching techniques for data are hot topics as the privacy of databases is sensitive in nature. Numerous approaches have already been proposed in various literatures including:

Sahab and Abdul Monem [7] introduced a technique for querying encrypted databases by creating index fields using hash functions. They suggested a data encoding approach to enable immediate querying without exposing index values to unauthorized users. Sultan and Brajendra [8] suggested a bit vector-based model BVM for executing SQL queries for encrypted databases in the cloud that works as an intermediary between users and the cloud provider. In BVM, before the encryption and outsourcing processes, the query manager (QM)

takes each record from the main table, parses it, builds a bit vector for it, and stores it. The BV stores bits, zero and one, and its length equals the total number of sub-columns for all sensitive columns.

Atheer et al. [6] proposed a technique for speeding up query retrieval in encrypted databases by designing inverted tables that retrieve the ID values for each column without repetition and thus decrypt only some of the records. Pawel et al. [9] presented a method for indexing and retrieving biometric data, for workload reduction, and template protection of biometrics using homomorphic encryption. The workload is reduced by utilizing a feature-level fusion of intelligently paired templates, a multi-stage search structure is created. During retrieval, the list of potential candidate identities is successively pre-filtered, thereby reducing the number of template comparisons necessary for a biometric identification transaction.

Kyriaki et al. [10] presented an approach to biometric data protection that utilizes two key modules: Deep Learning Indexer and Comparator. The approach consists of three phases: training, indexing, and searching. In the training phase, specialized deep learning models are used for each biometric modality, including OPQN for facial data and DHN for fingerprints and voice. In the indexing phase, the deep learning indexer creates an index catalog for each modality, including a hash representation and pseudonymized identifier. In the searching phase, the local DL comparator processes queries and employs distinct strategies for modalities. Encrypted data maintains similarity rankings without decryption during comparisons.

The references [6, 7, 8] included improving data retrieval time from encrypted databases. In contrast, [9, 10] worked to minimize the biometric data indexing time, which is closest to the proposed schema in terms of, using biometric data and improving data retrieval.

3. Proposed System

The suggested system consists of two main stages: the enrolment stage (including pre-processing, feature extraction, and template storing in the 3D database, as well as updating the inverted files), and the recognition stage (including pre-processing, feature extraction, search in inverted files for candidates, and matching). Fig. 1 illustrates the diagram of the proposed system architecture.

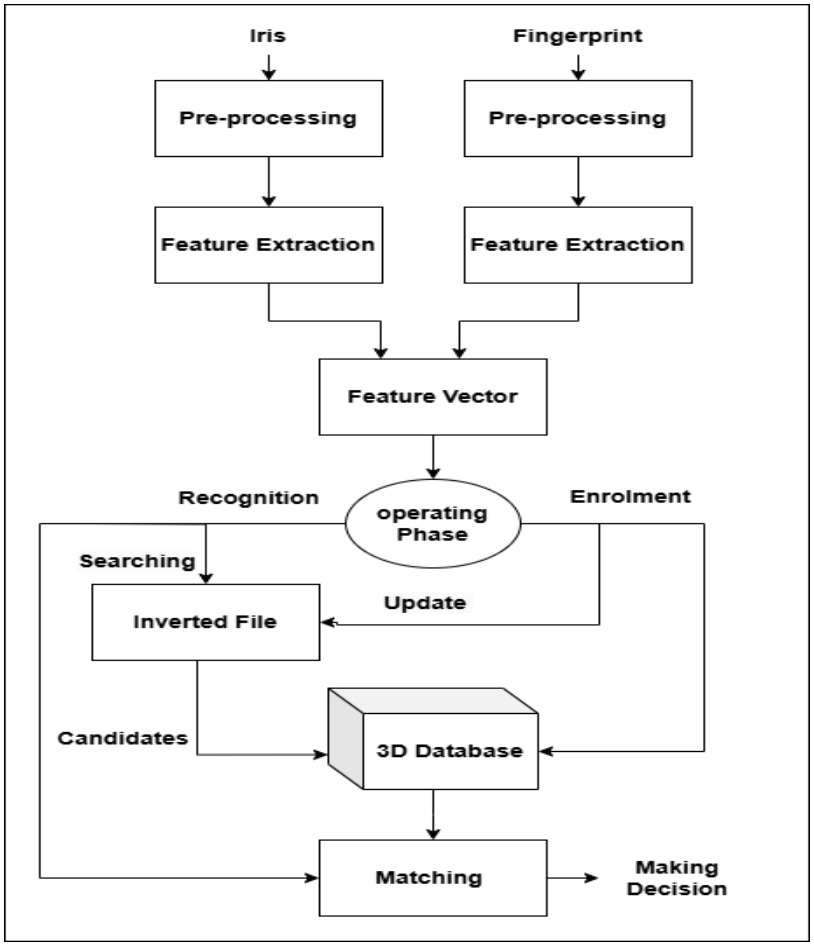

Figure 1: The Block Diagram of the Proposed System.

3.1 Pre-processing

To generate a template of the iris and fingerprint, both samples must be prepared for the features extraction process, due to issues that may arise during the acquisition of images and to detect the Region of interest ROI of the sample. In this work, the pre-processing for the iris image involves:

a. The iris is localized and normalized using the Daugman algorithm [11] that involves finding the center coordinates and the radius of the iris and the pupil and then converting the polar coordinates into cartesian coordinates and isolating only the iris region for further processing.

b. Improving the appearance of the image after detecting the ROI (lower region of the iris in order to decrease the effect of eyelashes and eyelids). Improving the ROI involves stretching its contrast using equation Eq. 1 [12]:

$$\text{stretch I(r, c)} = \left[\frac{\text{I(r,c)} - \text{I(r,c)}_{min}}{\text{I(r,c)}_{max} - \text{I(r,c)}_{min}}\right][max - min] + min \quad (1)$$

c. Divide the iris region into (2x2) nonoverlapping blocks for the next step of feature extraction.

Fig. 2 illustrates the pre-processing steps of the iris sample.

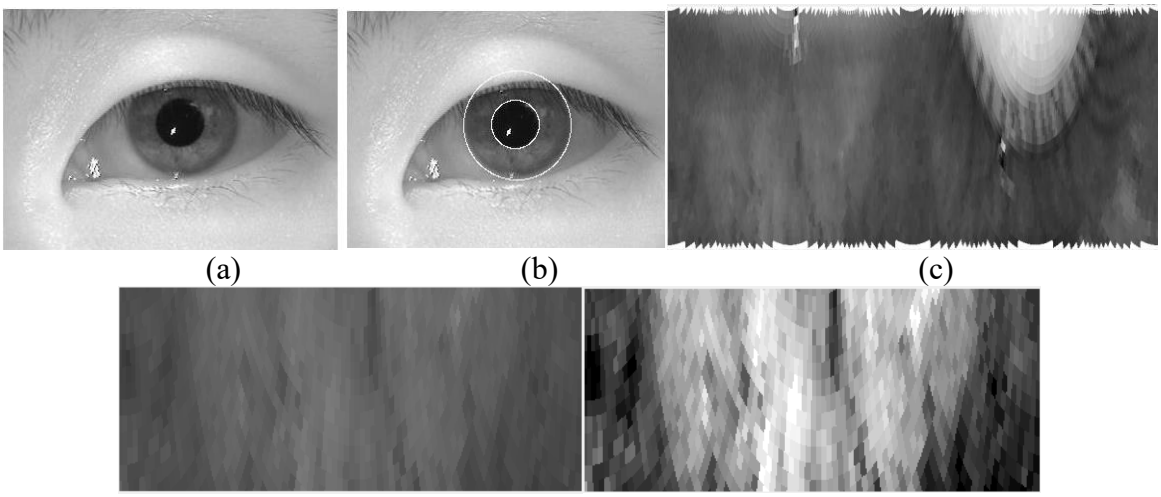

Figure 2: pre-processing of iris sample: (a) Grayscale. (b) Iris localization (c) Iris normalization (d) ROI (e) Adjusting the appearance.

For fingerprint patterns, the steps of pre-processing include:

a. Enhance the intensity of the image by stretching the contrast using equation (1).
b. Apply a canny edge detection method to focus on the ROI of the fingerprint image [13].
c. Merge the edge image with the original image to focus on extracting features from the area surrounding the fingerprint center.

Figure 3 illustrates the pre-processing steps of the fingerprint sample.

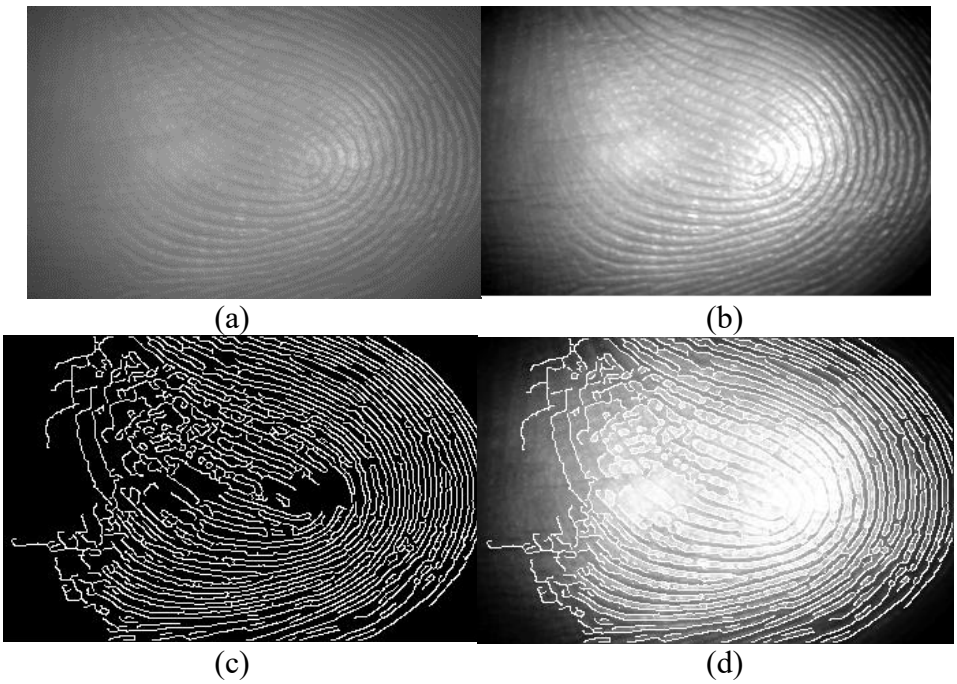

Figure 3: pre-processing of fingerprint sample: (a) fingerprint sample. (b) adjusting the appearance (c) fingerprint edge detection. (d) merge the fingerprint with the edge image.

3.2 Feature Extraction

After pre-processing, the feature extraction step takes place, and the mean, standard deviation, and energy equations [14] are applied for each block of the iris image, which results in 12 values that represent a specific set of features that can be used for inverted file construction.

This work uses the log-Gabor filter [15] for fingerprint feature extraction, resulting in 40 values.

3.3 Template Storing

The proposed system involves designing a database in 3D form (x, y, z), where (x, y) refers to the location and (z) is data. The structure of the 3D database is shown in Fig. 4. In this work, the dimension length is (100, 100, 100), and the maximum storage is 10000 records.

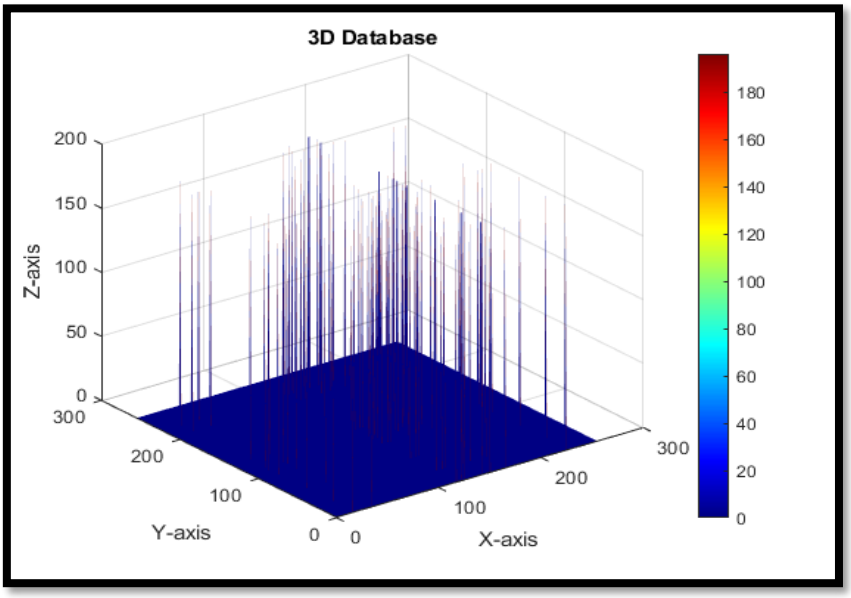

Figure 4: 3D database structure.

Each person added to this database will be given an ID using table (1). The IDs table is pre-constructed, which contains two columns. The first column represents the record number stored in the database, and the second column represents the ID of that record. The ID will cover the location that stores the entire feature vector of that person, where the location is determined using Eq. (1 and 2):

$$x = ID * a1 + b1 \, mod \, (p) \quad \ldots (1)$$

$$y = ID * a2 + b2 \, mod \, (p) \quad \ldots (2)$$

Where x and y refer to the location of the feature vector in the 3D database, a1, b1, a2, and b2 are constant digits that equal 20, 35, 25, and 40, respectively. While p denotes the first prime number less than the maximum storage of the database, which is 97.

Table 1: ID of persons.

No.	ID
1	103
2	45
3	920
.	.
.	.
.	.
9408	4712
9409	807

3.4 Inverted Files

The proposed system includes designing the inverted files that quickly skip most irrelevant data and match only candidates by constructing it based on a specific number of iris features. The number of inverted files depends on the length of the iris feature vector. In this work, 12 inverted files must be constructed. Each file contains a table with two columns, as shown in Fig. 5, the first column contains the range of values similar to the original iris features, and the second column includes the ID numbers.

Range values	ID
0.0-0.1	120 89 11....
0.1-0.2	17 5 96 ....
0.2-0.3	5 41 166 ....
0.3-0.4	22 50 8 ....
0.4-0.5	40 119 27 ....
0.5-0.6	59 21 47 ....
0.6-0.7	18 100 55 ....
0.7-0.8	31 134 65 ....
0.8-0.9	25 10 16 ....
0.9-1.0	1 88 71 ....

Range values	ID
0.0-0.1	27 5 19....
0.1-0.2	22 47 31 ....
0.2-0.3	3 17 21 ....
0.3-0.4	7 44 78 ....
0.4-0.5	4 65 13 ....
0.5-0.6	30 59 24 ....
0.6-0.7	88 102 2 ....
0.7-0.8	13 99 18 ....
0.8-0.9	73 15 1 ....
0.9-1.0	100 27 2 ....

. . .

Figure 5: Inverted file structure.

Two operations are performed on the inverted files: the update operation during the database construction phase and the search operation during the recognition phase. The update operation involves adding the ID of the person (whose biometric data is to be stored in the database) to the inverted files, for future use. Algorithm 1 shows the updated inverted files when new biometric data is stored in the database.

Algorithm 1- Update Inverted files
Input: iris feature vectors, ranges t1 and t2 of the iris features, IDs of each person in the database.
Output: update inverted files
Begin: Step 1: For i=1 to12 // features of iris For j=1 to the length of ranges if vector (i) >= t1(j) & vector(i) <= t2(k) store ID in the inverted file (i) End End End

The search operation in the inverted files involves finding the IDs of the candidates to determine the locations of the vectors that will be matched to complete the recognition

process as shown in Fig. 6. Algorithm 2 shows the searching and matching processes using inverted files.

Algorithm 2- Searching and matching processes using inverted files
Input: A feature vector of 12 values, ranges val1 and val2, inverted files
Output: person identity

Begin:
Step 1: For each value i in the vector
Check inverted file i if the value vector (i) falls within the ranges
val1 and val2 then
retrieved ID
End
Store the retrieved IDs
Step 2: Count the occurrences of each ID
Step 3: Extract IDs that have a count greater than 7 times % (It indicates that more than 7 out of 12 features of the iris are matched with the ranges).
Step 4: For all IDs getting from step 3:
Apply equations 1 and 2 to get the position (x, y) in 3D database.
End for
Step 5: For all candidate's vectors
For i=1 to 12 // refer to the iris features
apply Euclidean distance for matching iris features
if the similarity of iris features ≥ threshold, then
apply Euclidean distance for matching fingerprint features
if the similarity of fingerprint features ≥ threshold then
store the result.
End
End
Step 6: Search for maximum similarity resulting from step 5.
Step 7: Return the person's identity.
End

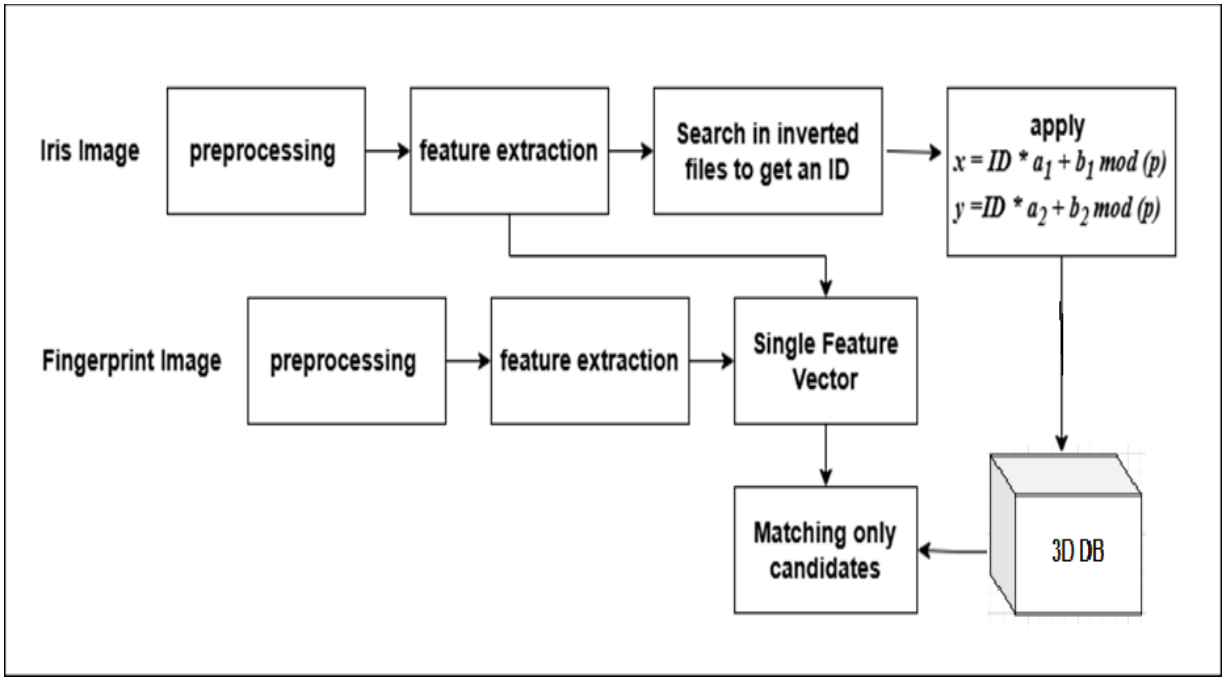

Figure 6: Recognition process.

4. Evaluation Metrics

The Correct Identification Rate (CIR) is a metric used to evaluate the accuracy of an identification task. Eq. 3 provides the CIR [15].

$$CIR = = \frac{Number\ of\ Correct\ Identifications}{Total\ Number\ of\ Identification\ Attempts} \times 100\% \ldots \quad (3)$$

The verification process's performance has been measured using the parameters of the Equal Error Rate (EER) and the Receiver Operating Characteristic (ROC) curve. The error rate is represented by (EER) when FAR equals FRR, which indicates a lower EER value. ROC shows the curve between the FAR and the FRR for different values of the threshold. FAR is the fraction of impostor scores greater than or equal to the threshold value, while FRR is the proportion of genuine scores less than the threshold value. The retrieval time metric was also used, which is an important measure for evaluating the performance of the proposed scheme [16].

5. Results and Discussion

The system's performance was evaluated using samples from the Multi Media University (MMU) dataset [17] to obtain iris samples. While the PolyU dataset [18] is the database from which fingerprint samples were taken. The program was written in the MATLAB (R2020b) programming language. To evaluate the performance, this system was tested on 990 images, 11 images of the iris and fingerprint for each person.

Table 2 illustrates the accuracy of the iris system when dividing the iris domain into different numbers of blocks, which shows the highest recognition rate when divided into (2 x 2) blocks.

Table 2: Accuracy of iris system

Block of iris	Accuracy %
1	30.1
1x2	66.2
2x2	86.3
2x4	85.9
4x4	85

The previous table shows that the 2x2 dividing provides enough complexity in the feature space to distinguish between different iris texture patterns. This complexity might drop when using simpler divisions like 1 block (too general) or 1×2 blocks (limited representation) and give a balance between feature extraction and matching. This balance explains why the accuracy is slightly lower for 2×4 and 4×4 divisions, despite still being high. Table 3 explains that the number of features is affected by the number of scales of the log-Gabor filter, which is reflected in the accuracy of the fingerprint system.

Table 3: Accuracy of fingerprint system

No. of scales	No. of features	Accuracy %
1	8	43.2
2	16	51.6
3	24	70.5
4	32	76
5	40	87.5
6	48	87.5

As shown in the previous table, the number of scales used by the log-Gabor filter impacts the quality and quantity of features extracted. A sufficient number of scales (five in this work) is crucial to achieving optimal accuracy, as they capture enough details without including redundancy or noise. Table 4 shows the effect of applying canny edge detection to improve the appearance of the fingerprint texture, thus increasing the accuracy rate.

Table 4: Effect of canny edge detection on accuracy.

Edge detection	Accuracy %
Without	68.33
With	87.5

Table 5 summarizes the accuracy rate of the recognition system with the different number of features when using only iris, or fingerprint patterns, and when using multi-biometric patterns. The system achieves an accuracy of 94.89 % with only 52 features, which indicates the ability to maintain high accuracy with a specific number of features.

Table 5: The accuracy of the biometric system

Biometric System	CIR %
Iris only	86.2
Fingerprint only	87.5
Iris & Fingerprint	94.89

The ROC curve illustrated in Figure 7, which also shows the lower EER equals 0.13 % at the threshold value of less than 57.

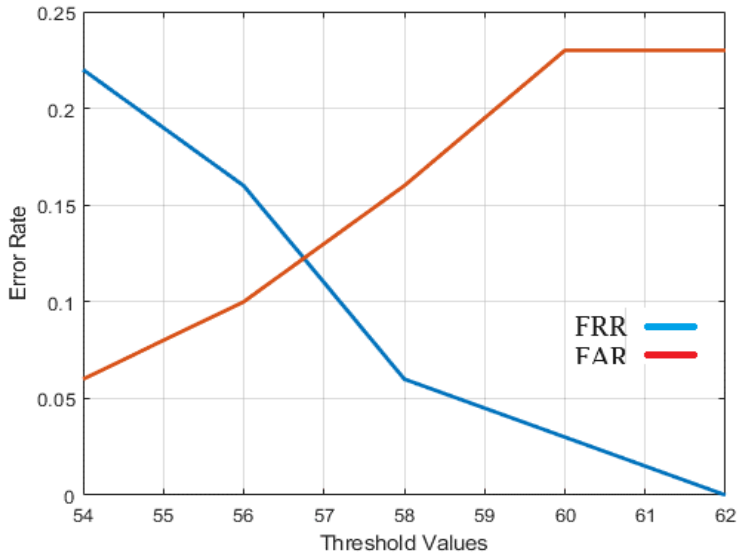

Figure 7: ROC Curve iris system.

The ROC curve is illustrated in Fig. 8, which shows the lower EER equals 0.155 % at the threshold value 61.

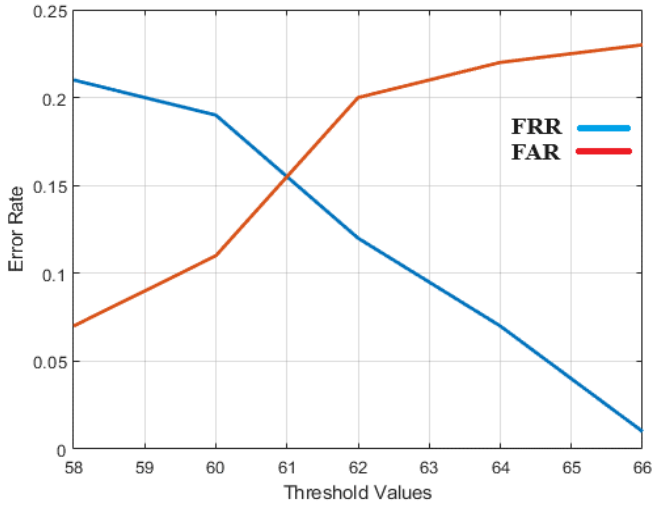

Figure 8: ROC Curve for fingerprint system.

The time complexity of the traditional method is O(N), where N is the number of feature vectors in the database. This linear scaling results in higher retrieval times as the database grows. The proposed method reduces the time complexity to O(k), where k is the number of relevant vectors retrieved from the inverted files. Since k<N, this significantly speeds up retrieval and matching. Table 6 shows the difference in retrieval and matching time between the traditional method, which compares the input vector with all vectors in the database, and when using the proposed inverted files technique.

Table 6: Average time for retrieval and matching **Time (m, s, ms).**

Time with the traditional method	Time with the suggested method
01.13.107	00:27.781

As shown in the previous table, the inverted files are to narrow down the search space and focus on the most relevant features. This is consistent with reducing the time complexity from O(N) in the traditional method to O(k) in the proposed method, where k<N. Thus, speeding up the retrieval and matching processes by avoiding searching the entire database.
The space complexity of the traditional method is O(N×f), where N is the number of vectors in the database, and f is the number of features per vector. The proposed method introduces an overhead for storing inverted files, resulting in a space complexity of O(g+k×f), where g is the storage requirement for the inverted files, and k×f is the storage for the retrieved relevant vectors. This trade-off is small compared to the gains in retrieval efficiency.
Table 7 demonstrates matching time and accuracy according to the threshold that determines the percentage of data retrieved from the database for matching. The threshold refers to the number of iris features of the input vector that matched the ranges of the inverted files.

Table 7: accuracy and time for various thresholds.

Threshold	Data %	Accuracy %	Time (m, s, ms)
5	60	94.89	00:43.864
6	51	94.89	00:37.285
7	38	94.89	00:27.781
8	34	94.51	00:24.856
9	25	94.48	00:18.277

As shown in the previous table, the "Data %" column shows the percentage of data retrieved from the database. As the threshold increases, it results in a smaller portion of the database being retrieved. Increasing the threshold reduces retrieval time by narrowing the search space while maintaining high accuracy. The system effectively balances speed and accuracy by reducing the data retrieved. While the inverted file structure introduces additional storage overhead for indexing, the overall space complexity remains efficient relative to the database size. This demonstrates the system's efficiency and scalability, as it adapts to various thresholds based on performance requirements (speed vs. accuracy).

6. Conclusion and Future Works

This paper presented an efficient searching and matching scheme for the storage and retrieval of biometric records in a multi-biometric recognition system, depending on constructing the inverted files. The system is based on iris and fingerprint patterns that provide (52) features, resulting in less time-consuming retrieval and matching as well as computational simplicity, and efficiency that can be used in real-time applications. The design of the inverted file system is specifically tailored to handle biometric data, significantly enhancing the speed and accuracy of responses. By employing this approach, the system avoids the need to search the entire database, which is traditionally required in many existing methods. Instead, the inverted file mechanism allows for a more immediate and targeted search process. This improvement in the retrieval process is crucial for applications that require rapid and reliable access to biometric data, such as security systems, identity verification, and access control systems. future works, improving the inverted file structure to effectively manage large datasets without losing performance. Additionally, developing dynamic techniques that can adapt to different sizes and types of biometric data would greatly enhance system efficiency.

References

[1] D. Maltoni, D. Maio, A. K. Jain, and J. Feng, (2022), "Handbook of fingerprint recognition: Third edition". doi: 10.1007/978-3-030-83624-5.

[2] Ali, Y. H., & Razuqi, Z. N. (2017). Palm vein recognition based on centerline. *Iraqi J Sci*, 58(2), 726-734.

[3] A. K. Jain and A. Kumar, (2012), Chapter 3 Biometric Recognition: An Overview. doi: 10.1007/978-94-007-3892-8_3.

[4] Al-Ani, L., Altaei, M., & Alwan, A. (2012). Iris Recognition Using Semantic Indexing. *Iraqi Journal of Science*, 53(4Appendix), 1137-1143.

[5] S. Berretti, A. Del Bimbo, and E. Vicario, (2001), "Efficient matching and indexing of graph models in content-based retrieval," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 23, no. 10, pp. 1089–1105, doi: 10.1109/34.954600.

[6] A. M. Abbas, A. M. S. Rahman and N. F. Hassan, (2020), "a new technique by using inverted t ables and 3d b ox for efficient Querying over an encrypted database", *Iraqi Journal for Computers and Informatics*, Vol. 46, Issue 2.

[7] A. M. S. Rahman and S. D. Mohammed, (2018), "Technique for querying over an encrypted database," *Int. J. Eng. Technol*., vol. 7, no. 4, pp. 6951–6955, doi: 10.14419/ijet.v7i4.22068.

[8] S. Almakdi and B. Panda, (2019), "Secure and Efficient Query Processing Technique for Encrypted Databases in Cloud," Proc. - 2019 2nd Int. Conf. Data Intell. Secur. ICDIS 2019, no. September, pp. 120–127, doi: 10.1109/ICDIS.2019.00026.

[9] P. Drozdowski, F. Stockhardt, C. Rathgeb, D. Osorio-Roig, and C. Busch, (2021) "Feature Fusion Methods for Indexing and Retrieval of Biometric Data: Application to Face Recognition with Privacy Protection," IEEE Access, vol. 9, pp. 139361–139378, doi: 10.1109/ACCESS.2021.3118830.

[10] K. Miniadou, A. Leonidis, G. Th. Papadopoulos, and C. Stephanidis, (2024), "Encrypted Biometric Search: A Deep Learning Approach to Scalable and Secure Cross-Border Data

Exchange," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 2794-2800, doi: 10.1109/BigData62323.2024.10825332.

[11] J. Daugman, "How Iris Recognition Works," Essent. Guid. to Image Process., vol. 14, no. 1, pp. 715–739, 2009, doi: 10.1016/B978-0-12-374457-9.00025-1.

[12] A. Agha and L. George, (2014), "Palm Veins Recognition and Verification System: Design and Implementation,".

[13] R. Song, Z. Zhang, and H. Liu, (2017), "Edge Connection based Canny Edge Detection Algorithm," vol. 8, no. 6, pp. 1228–1236.

[14] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, (2020), "Feature Extraction Methods: A Review," J. Phys. Conf. Ser., vol. 1591, no. 1, doi: 10.1088/1742-6596/1591/1/012028.

[15] Fasna K, R. J. Krishna S, P. Scholar, and A. Professor, (2016), "A Review on Iris Feature Extraction Methods," Int. J. Eng. Res. Gen. Sci., vol. 4, no. 2, pp. 663–667, [Online]. Available: www.ijergs.org

[16] D. Petrovska-Delacrétaz, G. Chollet, and B. Dorizzi, (2009) "Guide to biometric reference systems and performance evaluation," Guid. to Biometric Ref. Syst. Perform. Eval., no. September 2015, pp. 1–382, doi: 10.1007/978-1-84800-292-0.

[17] MultiMedia University (MMU) iris dataset, [Online]. Available: MMU iris dataset (kaggle.com)

[18] PolyU fingerprint dataset, [Online]. Available: Ajay Kumar, The Hong Kong Polytechnic University, Hong Kong (polyu.edu.hk)