Fingerprint Recognition System based on Artificial Neural Networks integrated with Machine Learning

Ummadisetty Nagamani[1*], *S.* MohasinaTabassum[1], *B.* Jayasree[1], *V.* Prashanth[1], *S.* Srinath[1], and *S.* Srikanth[1]

[1]Department of Electronics and Communication Engineering, Kandula Sreenivasa Reddy Memorial College of Engineering (KSRMCE), Kadapa, India

Abstract. This study represents a fingerprint recognition approach that combines Artificial Neural Networks (ANN) with Machine Learning (ML) approaches to improve biometric identification in terms of precision, efficiency, and dependability. Due to its distinctiveness and permanence, fingerprint recognition is a widely adopted biometric authentication method in security systems. However, challenges such as noise, partial prints, and varying environmental conditions often degrade performance. system begins by preprocessing fingerprint images to reduce noise and extract distinctive features, such as ridges and minutiae points. These topographies are then used as inputs to a deep neural network model trained on labelled data set of fingerprint samples. ANN architecture is optimized to recognize and classify fingerprint patterns accurately, even when presented with partial or low-quality images. To strengthen system robustness, Machine Learning strategies such as data augmentation are utilized, leading to enhanced performance and dependability. Through extensive testing, our system demonstrates high accuracy in correctly identifying individuals across a variety of conditions, with an emphasis on minimizing false positives and negatives. This fingerprint recognition system has potential applications in secure access control, identity verification, and law enforcement, offering a highly reliable, scalable, and efficient solution for biometric security.

1 Introduction

In the evolving world of security and identity management, biometric systems play an essential role in authenticating individuals using unique biological traits [1]. Among various biometric techniques, fingerprint recognition has long been used due to its uniqueness, ease of capture, and permanence over time [2]. It is widely adopted in applications ranging from smartphone security to large-scale forensic investigations. Despite its reliability, conventional fingerprint recognition systems often face performance challenges when working with real-world images. Variations in image quality, skin conditions, partial fingerprints, and sensor noise can Lowers the performance and reliability of conventional classifiers like KNN (K-Nearest Neighbours), SVM (Support Vector Machines) and ANN (Artificial Neural Networks) [3]. These systems typically rely on handcrafted feature extraction methods, which are sensitive to distortions and may fail to generalize when the input data varies significantly. In order to address these challenges, this study introduces a fingerprint recognition approach that combines CNN-driven deep learning with traditional classifiers, utilizing Gabor filter–based feature extraction. The system is trained and evaluated on large- scale fingerprint data set — SOCOFing — which contains over 9,000 fingerprint images from multiple finger classes. The CNN model is designed to automatically extract and learn distinctive fingerprint features, eliminating by manual feature engineering and improving recognition accuracy, even in challenging conditions. In addition to the deep learning model, the study compares its performance with traditional classifiers using Gabor filter-based features to demonstrate the advantages of CNN-based fingerprint recognition. Results from experiments emphasize the proposed system's reliability, precision, and robustness, confirming its practicality for real-world use cases like digital ID verification, access security, and forensic investigations.

2 Problem Identification

Conventional fingerprint systems frequently face difficulty achieving accurate results with degraded, incomplete or noisy fingerprint data [4]. Handcrafted feature extraction techniques used in conventional machine learning algorithms such as KNN, SVM, and ANN, are sensitive to image variations and may fail to capture complex fingerprint patterns [5].

This creates the need for an automated approach capable of learning distinctive features directly from raw fingerprint images and handling diverse data conditions. To tackle this, the study proposes a CNN-based fingerprint recognition model that provides enhanced precision and reliability over conventional approaches.

* Corresponding author: ummadisetty.nagamani@gmail.com

3 Proposed Frame Work

The proposed framework is represented through a block diagram, incorporating CNN to improve accuracy. SOCOFing data set is used for training and validating model, ensuring dependable and scalable biometric authentication, as illustrated in Figure 1.

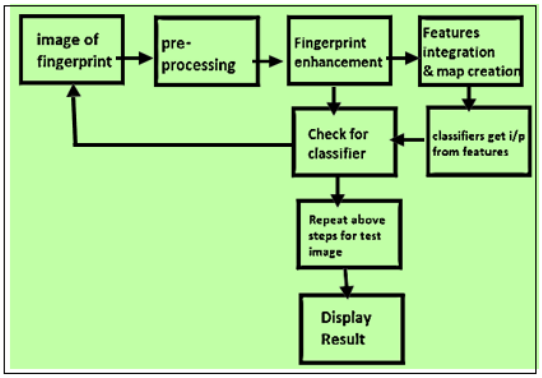

Fig. 1. CNN layer-wise block diagram.

3.1 Preprocessing

Preprocessing is the first step in the fingerprint recognition process, aimed at improving image quality and removing noise [6]. This includes steps such as resizing, normalization, and enhancement techniques to make the fingerprint image more suitable for feature extraction.

3.2 Extraction of Features

Extraction of Features focuses on detecting and isolating essential fingerprint characteristics, including ridge endings, bifurcations, and ridge patterns [7]. This step is essential for distinguishing unique features that can be used for matching.

3.3 CNN Architecture

As shown in Figure 2, The CNN framework is structured to autonomously identify and capture significant fingerprint features from input images[8].This structure has many different layers, which help in classifying the fingerprint pattern based on the features extracted during training.

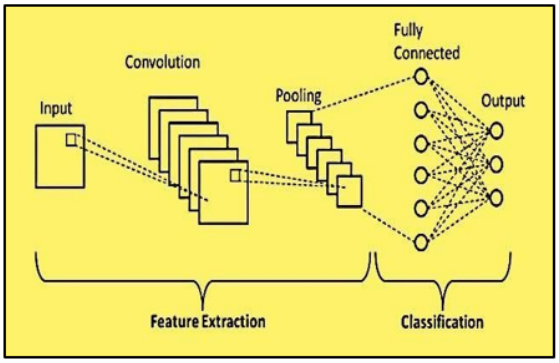

Fig. 2. CNN architecture.

3.3.1 Data Input Layer

The Data input layer receives the fingerprint image, usually pre-processed to a consistent size and normalized to ensure uniform pixel intensity. This step ensures the network works with high-quality input data.

3.3.2 Convolutional Layers

Convolutional layers use kernels to identify key fingerprint characteristics like ridges, branching points, and minutiae. [9]. These layers help the model automatically learn essential patterns from the image without manual feature extraction.

3.3.3 Pooling Layers

This layer reduces dimensions to smaller size, lowering computational effort while preserving critical information to enhance quality and minimize over fitting risks.

3.3.4 Flattening

The extracted features from convolutional and pooling layers are reshaped into single-dimensional vector, making it compatible for further processing in fully connected layers [10].

3.3.5 Fully Linked Layers

The dense layers receive the processed feature vector and analyze its high-level patterns to determine whether a fingerprint matches or not [11].

3.3.6 Output Layer

The output layer classifies the fingerprint at the final stage by assigning it to as specific class using probability scores, typically via a Sigmoid activation. [12].

3.4 Training the Model

At this stage, the CNN is trained using labeled fingerprint images [13]. With the help of back propagation and optimization methods such as gradient descent, the model continuously refines its accuracy by reducing errors in classification.

3.5 Fingerprint Matching

After the model has been trained, it proceeds to fingerprint matching, where newly acquired fingerprint features are compared with stored records to check for a match and to display the results [14].

4 Algorithm

Input**:** Fingerprint imagedataset(TrainingandTesting sets)
Output: Predicted class label for input fingerprint.

Step1:Acquirefingerprintimagesfromthedataset.
Step 2**:** Perform pre-processing [15]:

- Resize images to a uniform dimension.
- Apply normalization and enhancement to reduce noise and improve quality.

Step3:Pre-processed images are fed into the CNN model.
Step4:Apply convolutional layers for automatic feature extraction.

- Detect edges, ridge patterns, bifurcations and minutiae.

Step5:Use combining layers to decrease spatial dimensions of feature maps while preservative most significant features [16].
Step6: Flatten output from pooling layers into one dimensional feature vector.
Step7: Feed flattened vector into fully connected layers to perform final classification.
Step 8: Apply Softmax activation to compute class probabilities.
Step 9:Duringtraining:

- Use labeled data to compute loss using cross-entropy.
- Optimize the model with back propagation and gradient descent.
- Repeat until convergence. Step 10: During testing:
- Input the new fingerprint image.
- Follow Steps2–8.
- Output the predicted fingerprint class label.

5 Problem Definition

Traditional fingerprint recognition techniques are often limited by manual feature extraction, sensitivity to image quality, and lower robustness in real-world scenarios [17]. The challenge lies in developing a system that can automatically extract and classify discriminative fingerprint features with high accuracy even under varying conditions, including distortions, noise, and partial impressions.This research aims to introduce a CNN-based fingerprint system that addresses these issues, boosts classification results, and strengthens biometric protection.

6 Experimental Setup

Performance assessment of the proposed fingerprint recognition method was conducted with the SOCOFing dataset (Sokoto Coventry Fingerprint Dataset). The dataset consists of over 9,000 fingerprint images belonging to 5 distinct classes — index, middle, ring, little, and thumb. The dataset offers a variety of samples with varying image quality, distortion, and rotations, making it ideal for testing the robustness of the proposed Convolutional Neural Network (CNN).

In the experimental setup, 80% of the dataset images were used for training, while the remaining 20% were reserved for testing the performance of the model [18].

The sample images from each finger print class are shown in Figure 3.

Fig. 3. Sample fingerprint images from the SOCOFing data set (one image per class).

All model training and testing were conducted using MATLAB R2018a on a personal computer with Intel Core i3 processor, 4 GB RAM and running Windows 10 operating system. CNN model was implemented and trained using MATLAB's Deep Learning Toolbox.

The training process utilized the following hyper parameters:

- Mini-batchsize:64
- Epochs:10
- Learningrate:0.001
- Optimizer: StochasticGradientDescent (SGD)

7 Performance Metrics

Effectiveness of fingerprint recognition model is assessed by means of standard classification performance metrics.

1. Accuracy: Indicates total proportion of predictions that were correctly classified. Accuracy is defined as shown in Equation (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2. Precision: Ratio of true positive predictions to total predicted positives. Precision is given by Equation (2).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3. Recall (Sensitivity): Reflects how well the model identifies true positive cases. As shown in Equation (3), it is mathematically expressed as follows:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

4. Specificity: The capability of a model to accurately identify true negative cases. It can be calculated as shown in Equation (4).

$$Specificity = \frac{TN}{TN+FP} \quad (4)$$

Where,

- TP = True Positives
- TN = True Negatives

- FP = False Positives
- FN = False Negatives

5. F1-Score: Harmonic mean of precision and recall, offering a balanced amount reflects both false positives then false negatives. This relationship is mathematically represented in Equation (5).

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

8 Experimental results

8.1 Performance of Convolutional Neural Network

8.1.1 Training Progress Plot

Training of the model was performed using a batch size of 64 over 10 epochs with a learning rate set to 0.001, shows steady convergence as observed in training accuracy plot, as shown in Figure 4.

Simultaneously, the loss function showed a marked decrease, reflecting a reduction in prediction errors. This decline in loss alongside rising accuracy suggests that the model parameters were successfully optimized, leading to better performance on the training dataset. The stabilization of both accuracy and loss towards the later epochs indicates that the model reached a point of convergence, where further training yielded minimal improvement.

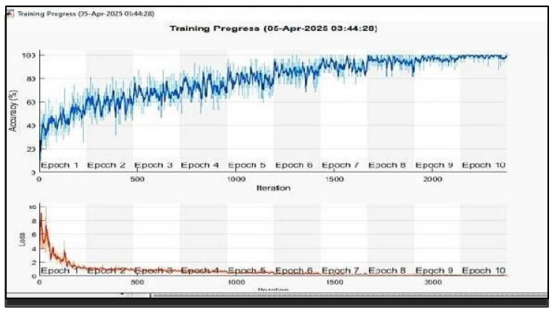

Fig. 4. Training accuracy and loss plot of the CNN model.

8.1.2 Training Table Values

Table1. final training statistics.

Epoch	Iteration	Time	Accuracy	Loss	Learningrate
1	200	00:13:48	43.75%	1.2293	0.0010
2	450	00:27:24	59.38%	0.9501	0.0010
3	700	00:42:41	56.25%	1.0030	0.0010
4	950	00:58:25	62.50%	0.9142	0.0010
5	1150	01:05:50	75.00%	0.7076	0.0010
6	1400	01:15:14	84.38%	0.4814	0.0010
7	1650	01:23:27	93.75%	0.2720	0.0010
8	1900	01:31:06	90.63%	0.1615	0.0010
9	2100	01:37:03	100.00%	0.1306	0.0010
10	2380	01:45:26	100.00%	0.0585	0.0010

The final training statistics for CNN model are summarized in Table 1. These values represent model's performance on training data set over multiple epochs.

8.1.3 Confusion Matrix

To evaluate model's class-wise prediction performance, a confusion matrix was generated, which visualizes correct classifications versus misclassifications across fingerprint classes [19]. Figure 5 shows Confusion matrix.

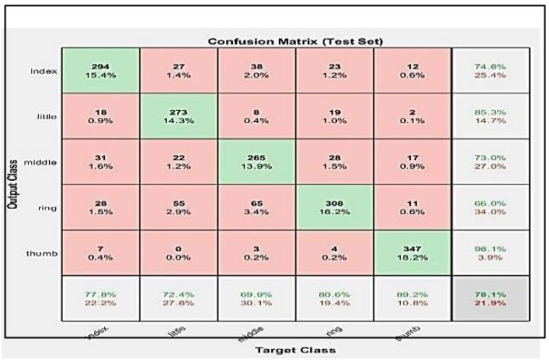

Fig. 5. Confusion matrix plot.

8.1.4 Performance Metrics Table

Table 2. Performance evaluation values.

class	Precision	Recall	F1 Score	Specificity
Index	74.6%	77.8%	76.2%	93.46%
Little	85.3%	72.4%	78.3%	96.14%
Middle	73.0%	69.9%	71.4%	94.20%
Ring	66.0%	80.6%	72.6%	89.57%
Thumb	96.1%	89.2%	92.5%	99.08%

To assess the CNN model's performance, common evaluation metrics such as Accuracy, Precision, Recall, and F1-Score were employed, with the results summarized in Table 2.

8.1.5 Sample outpu timages

Figure 6 shows sample outputs generated by the CNN model on test finger print images.For each input,the predicted class matched the true label, illustrating effectiveness of the model in learning complex finger print features.

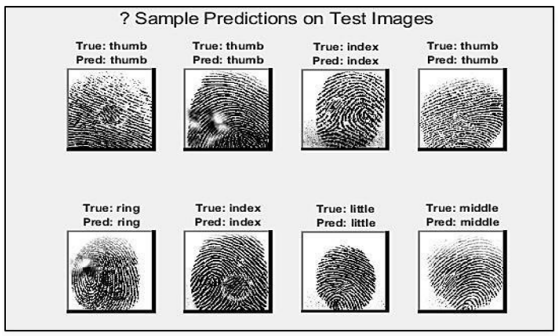

Fig. 6. Sample predictions on test images.

8.2 Performance of Traditional Classifiers

Performance evaluation was carried for three widely used traditional classifiers: K-Nearest Neighbor, Support Vector Machine and Artificial Neural Network. These classifiers were trained using Gabor filter-based feature vectors extracted from fingerprint data set [20]. The performance of the CNN model was further evaluated using standard metrics such as Accuracy, Precision, Recall, and F1-Score. Illustrated in Tables 3, 4 and 5.

Table 3. KNN Evaluation Values.

class	Precision	Recall	F1 Score	Specificity
Index	36.9%	56.6%	44.7%	76.0%
Little	45.9%	52.5%	49.0%	84.8%
Middle	46.8%	33.2%	38.9%	90.6%
Ring	46.0%	35.9%	40.3%	89.4%
Thumb	79.2%	66.6%	72.3%	95.5%

Table 4. SVM Evaluation Values.

class	Precision	Recall	F1 Score	Specificity
Index	16.9%	29.6%	21.6%	64.0%
Little	22.6%	15.6%	18.5%	86.8%
Middle	34.0%	9.5%	14.8%	95.4%
Ring	22.7%	32.5%	26.7%	72.3%
Thumb	50.8%	43.2%	46.7%	89.2%

Table 5. ANN evaluation values.

class	Precision	Recall	F1 Score	Specificity
Index	40.1%	35.7%	37.8%	86.8%
Little	52.0%	66.8%	58.6%	84.9%
Middle	44.2%	36.9%	40.2%	88.4%
Ring	43.1%	35.9%	39.1%	88.1%
Thumb	68.9%	79.7%	73.9%	90.8%

8.2.1 Confusion matrices plots

Figures 7,8, and 9 illustrate confusion matrices obtained for KNN, SVM, and ANN classifiers, respectively.From analysis ANN model showed comparatively better class-wise prediction capability, where as SVM model struggled the most with accurate classification.The KNN model performed moderately, but still showed noticeable misclassification across several classes.

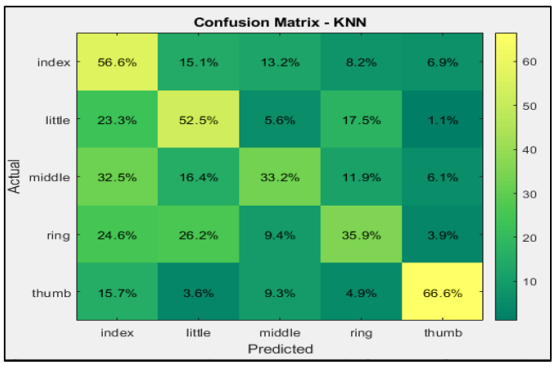

Fig. 7. KNN confusion matrix plot.

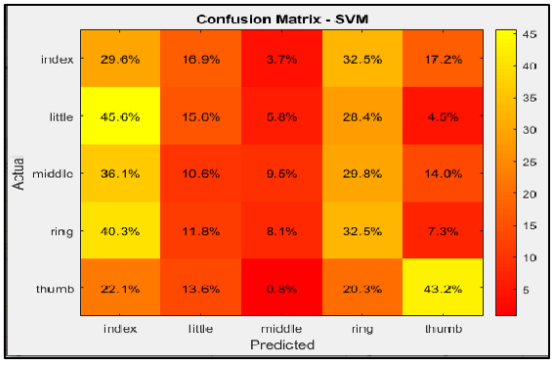

Fig. 8. SVM confusion matrix plot.

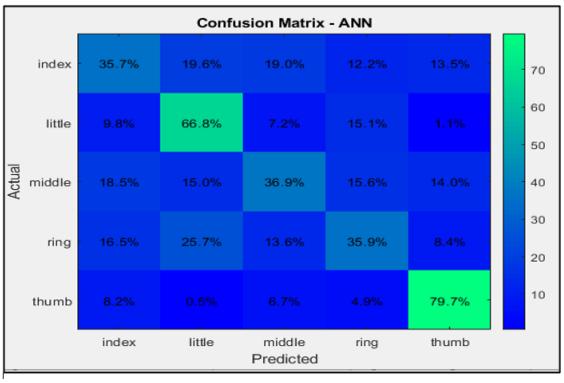

Fig. 9. ANN confusion matrix plot.

8.2.2 Performance Comparison Between CNN and Traditional Classifiers

Table 6. Summary table.

classifier	Accuracy	precision	Recall	F1 score	Specificity
CNN	78.02%	79.0%	77.98%	78.2%	94.49%
KNN	49.02%	50.97%	48.96%	49.04%	87.27%
SVM	26.19%	29.39%	26.08%	25.65%	81.55%
ANN	51.12%	49.67%	51.01%	49.92%	87.78%

Table 6 provides a comparative analysis of the classification performance of four classifiers: CNN, KNN, SVM, and ANN. Among them, the CNN model achieves the highest performance, with an accuracy of 78.02%, precision of 79.0%, recall of 77.98%, F1 score of 78.2%, and specificity of 94.49%. In contrast, the SVM classifier records the lowest performance, attaining only 26.19% accuracy across all metrics. The KNN and ANN classifiers exhibit moderate results, with

accuracies of 49.02% and 51.12%, respectively. Overall, these findings demonstrate that the CNN model significantly outperforms the other classifiers on this dataset and task.

Performance order: CNN>ANN>KNN>SVM

8.2.3 Accuracies plot

Figure 10 illustrates the bar graph comparison of classification accuracies achieved by the CNN, KNN, SVM, and ANN models. Among all models, the CNN achieved highest accuracy of 78.02%, showcasing its robust ability in both feature extraction and classification. In contrast, the traditional classifiers — ANN (51.13%), KNN (49.03%), and SVM (26.19%) — demonstrated comparatively lower performance. This result confirms the effectiveness of deep learning-based approaches over conventional machine learning methods for fingerprint recognition.

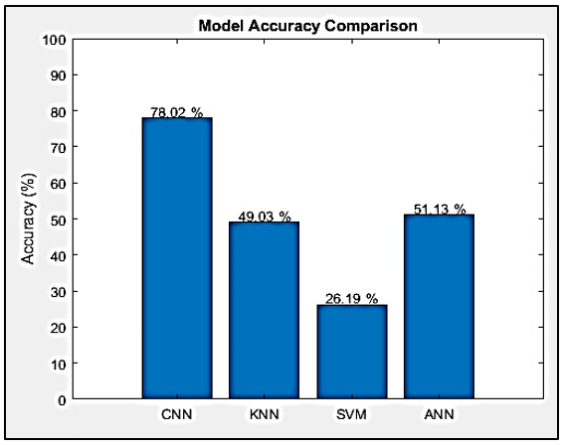

Fig. 10. Bar graph representing accuracies.

9 Conclusion And Future Scope

The study focuses on designing and evaluating a fingerprint recognition system using a combination of deep learning and machine learning approaches.A custom Convolutional Neural Network model was developed and compared with conventional classifiers, including K-Nearest Neighbor, Support Vector Machine and Artificial Neural Network using large-scale fingerprint data set. Experimental findings revealed that CNN model significantly outperformed the traditional approaches, highlighting its enhanced capability to extract then learn distinctive features directly from raw fingerprint images. For future work, the system can be enhanced by exploring advanced deep learning techniques such as transfer learning, hybrid models, or attention-based mechanisms to further improve performance. Additionally, the scope can be extended to real-time fingerprint recognition applications and multi-modal biometric systems by integrating fingerprint data with other biometric traits, which could improve reliability and security in authentication systems.

References

1. T. Vijaya Kumar, Synthesis of palm print in feature fusion techniques for multimodal biometric recognition system online signature. J. Innov. Image Process. **3**, 131-143 (2021). https://doi.org/10.36548/jiip.2021.2.005
2. M. Oloyede, G. Hancke, Unimodal and multimodal biometric sensing systems: A review. IEEE Access **4**, 7532-7555 (2016). 10.1109/ACCESS.2016.2614720
3. E. Rahmawati, M. Listyasari, A. Shidqul Aziz, Digital signature on file using biometric fingerprint with fingerprint sensor on smartphone in engineering technology and applications, In Proceedings of International Conference on Electronics Systems, IEEE, Surabaya, Indonesia December 28 (2017), 1–8.
4. S. Molaei, M.E. Shiri Ahmad Abadi, Maintaining filter structure: A Gabor-based convolutional neural network for image analysis. Appl. Soft Comput. J. **88**, 105960 (2020). https://doi.org/10.1016/j.asoc.2019.105960
5. J. Samuel Manoharan, A novel user layer cloud security model based on chaotic Arnold transformation using fingerprint biometric traits. J. Innov. Image Process. **3**, 36-51 (2021). https://doi.org/10.36548/jiip.2021.1.004
6. H.S. Khalifa, H.I. Wahhab, A.N. Alanssari, M.A.O. Ahmed Khfag, Fingerprint segmentation approach for human identification. Int. J. Appl. Math. Inf. Sci. **13**, 515-521 (2019). http://dx.doi.org/10.18576/amis/130401
7. S. Hemalatha, A systematic review on fingerprint based biometric authentication system, In Proceedings of International Conference on Emerging Trends in Information Technology and Engineering IEEE, Vellore, India, April 27 (2020), 1-4
8. R. Zhang, Y. Zheng, A survey on biometric authentication: Toward secure and privacy-preserving identification. IEEE Access **7**, 5994-6009 (2019). https://doi.org/10.1109/ACCESS.2018.2889996
9. K. Khongkraphan, An efficient fingerprint matching by multiple reference points. J. Inf. Process. Syst. **15**, 22-33 (2019). https://doi.org/10.3745/JIPS.04.0098
10. J. Khodadoust, A.M. Khodadoust, Fingerprint indexing based on minutiae pairs and convex core point. Pattern Recognit. **67**, 110-126 (2017). https://doi.org/10.1016/j.patcog.2017.01.022
11. W. Bian, S. Ding, Y. Xue, Combining weighted linear project analysis with orientation diffusion for fingerprint orientation field reconstruction. Inf. Sci. **396**, 55-71 (2017). https://doi.org/10.1016/j.ins.2017.02.043
12. S.R. Borra, G.J. Reddy, E.S. Reddy, Classification of fingerprint images with the aid of morphological operation and AGNN classifier. Appl. Comput. Inform. **14**, 166-176 (2018). https://doi.org/10.1016/j.aci.2017.07.001

13. C. Xie, A. Kumar, Finger vein identification using convolutional neural network and supervised discrete hashing. Pattern Recognit. Lett. 119, 148-**156** (2019). https://doi.org/10.1016/j.patrec.2017.12.001
14. H.T. Nguyen, Fingerprints classification through image analysis and machine learning method. Algorithms **12**, 241 (2019). https://doi.org/10.3390/a12110241
15. N.A. Alam, M. Ahsan, M.A. Based, J. Haider, M. Kowalski, An intelligent system for automatic fingerprint identification using feature fusion by Gabor filter and deep learning. Comput. Electr. Eng. **95**, 107387 (2021). https://doi.org/10.1016/j.compeleceng.2021.10738 7
16. N.R. Pradeep, J. Ravi, A revolutionary fingerprint authentication approach using Gabor filters for feature extraction and deep learning classification using convolutional neural networks, in Proceedings of International Conference on Networks and Systems, March 13 (2022), 349-360
17. N.R. Pradeep, J. Ravi, An accurate fingerprint recognition algorithm based on histogram oriented gradient (HOG) feature extractor. Int. J. Electr. Eng. Technol. **12**, 19-32 (2021). https://doi.org/10.34218/IJEET.12.2.2021.003
18. J. Ezeobiejesi, B. Bhanu, Latent fingerprint image segmentation using fractal dimension features and weighted extreme learning machine ensemble, in Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR June 26 (2016), 146-154.
19. E. Adam, E.E. Babikir, Evaluation of fingerprint liveness detection by machine learning approach – a systematic view. J. IoT Soc. Mob. Anal. Cloud **3**, 16-30 (2021). https://doi.org/10.36548/jismac.2021.1.002
20. A.E. Omran, R.F. Soliman, A.A. Eisa, Fusion of deep learned and hand-crafted features for cancellable recognition systems. Soft Comput. **24**, 15189-15208 (2020). https://doi.org/10.1007/s00500-020-04856-1