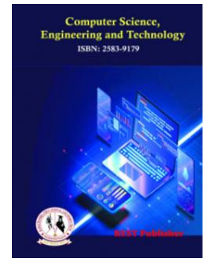
Computer Science, Engineering and Technology

Vol: 3(2), June 2025

REST Publisher; ISSN: 2583-9179

Website: https://restpublisher.com/journals/cset/

DOI: https://doi.org/10.46632/cset/3/2/8



Fingerprint Trivialities Detection Using Deep Learning

*** D. Haripriya, Ammar Ahmad, Katakam Subramanyam, M. Abdul Gani Shoaib**

Vel Tech Rangarajan Dr. Sagunthala, R&D Institute of Science and Technology, Chennai, India.
**Corresponding author Email: drharipriyad@veltech.edu.in*

Abstract: *Neophyte finger print minutiae are producing extremer forensic evidence for commercial criminal inquiries. Such minutiae are hard to identify from fingerprint images due to extremely small size and irregular shape minuscule. In that method, we pioneered the deep learning based segmentation, AGNN classifier, and wavelet feature for fingerprint triviality detection project. Then, preprocess fingerprint images through contrast enhancement, demising, and image segmentation for isolating ROI where the fingerprint will be present alone, and then create a deep learning model based on Convolutional Neural Networks (CNNs) that will learn the features of those fingerprints and label them as trivial or non-trivial. Apart from that, employ the AGNN model, which employs the learned spatial relationships between the trivialities in the fingerprint, to determine whether the photographs belong to the trivial or non-trivial category. Lastly, wavelet analysis will be employed to obtain features that reflect the texture and form of the trivialities from the fingerprint image. The model first applied image segmentation algorithms to obtain minutiae from fingerprint images. Then, the AGNN classifier, an attention-based graph neural network, will be employed for classification of minute based on their derived features. In addition, these wavelet features are also derived from the fingerprint image to enhance the accuracy of classification. The system proposed anticipates an increase in accuracy and reliability in fingerprint identification systems and has extensive applications in the security sector.*

Key words: *Fingerprint Recognition, Biometric Authentication, Triviality Detection, Deep Learning, Convolutional Neural Network, Image Processing, Pattern Recognition, Spoof Detection, Machine Learning, Security and Privacy.*

1. INTRODUCTION

The principle behind fingerprint identification relies on biometric identification, and in the stage of minutiae detection, it is concerned with the distinctive endings and bifurcations of ridges distinguishing one fingerprint from another. The conventional method had to depend on manual effort using sophisticated image processing algorithms coupled with human verification, which leaves the process open for errors and time constraints. The innovation in deep learning-expressed through the success of Convolutional Neural Networks (CNN)-is a milestone for a highly efficient and trustworthy method to the automation of minutiae detection with various orders of precision. Having been trained on extensive datasets of fingerprint images and their minutiae labels, the models are able to learn patterns within the minutiae more deeply and delve into their features further than conventional algorithms are capable of. One of the major benefits of this AI-powered platform is that it can offer software users nearly instant and helpful feedback. Candidates receive valuable information about their weaknesses and strengths so that they can correct their communications and presentation styles with precise instructions. This instant feedback loop enables repeated practice in a secure, non-stressful environment, which is generally not possible with conventional interview preparation. Deep learning is able to respond appropriately to the intrinsic variability and noise in fingerprint images. Distortions, partial prints, and poor image quality are factors that introduce serious challenges for conventional image processing methods. Of course, with hierarchy feature learning, CNNs take it a step further, etching noise-out to enable feature extraction of sound minutiae. This soundness is also required in application scenarios because of the practical limitations under which fingerprint images might actually be captured. Apart from the automaticity of deep learned minutiae detection reducing the workload by hand, these systems also decrease overall processing time and provide easy scalability.

2. RELATED WORK

Hariharan et al. [1] presented a discriminative decorrelation technique with the goal of enhancing clustering and classification performance. While not for fingerprint recognition per se, their methodology for bettering feature reparability carries over to biometric feature processing. Wieclaw et al. [2] suggested a minutiae-based algorithm for fingerprint matching, giving one of the early frameworks for fingerprint verification systems. The research proves the efficiency of minutiae features, which even with the use of contemporary techniques remain core to fingerprint recognition. Mary Lourde and Khosla et al. [3] investigated the fingerprint recognition systems in biometric security and presenting the fundamental principles and difficulties. Their research highlights the necessity of effective robustness and error recovery in actual fingerprint recognition scenarios. Cao et al. [4] suggested a coarse-to-fine ridge structure dictionary method for latent fingerprint segmentation and enhancement. Their approach greatly enhances the quality of latent fingerprint images, which tend to be degraded, thereby facilitating subsequent processing operations like feature extraction and matching. Walhazi et al. [5] designed a morphological snake-based preprocessing method to enhance the segmentation of latent fingerprint images. Results indicate that preprocessing is crucial to make poor-quality fingerprints more visible and usable. Zhu et al [6] introduced a convolutional neural network (CNN)-based latent fingerprint segmentation technique that effectively extracts ridge patterns from noisy backgrounds. Serafim et al. [7] proposed another CNN-based method, further validating the potential of deep learning for fingerprint image analysis. Gonzalez and Wood set al. [8] offered basic ideas in digital image processing, some of which form the basis of contemporary fingerprint enhancement and segmentation techniques. Filtering, enhancement, and morphological transformations are essential in readying fingerprint images for use in recognition processes. Pankanti et al. [9] investigated the uniqueness of fingerprints, offering statistical proof for the uniqueness of fingerprints. Their research supports continued reliance on fingerprint biometrics for security-critical tasks. Liu et al. [10] constructed a latent fingerprint segmentation algorithm based on ridge density and orientation stability, providing an efficient approach to isolate fingerprint areas from background interference.

3. SYSTEM DESIGN

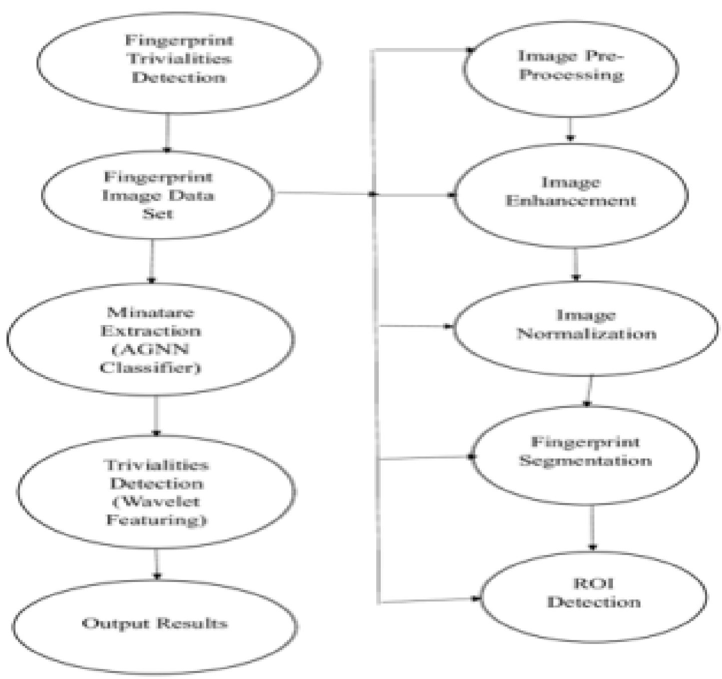


FIGURE 1. Data Flow Diagram

- Fingerprint Trivialities Detection: First detection of image noise or non-relevant patterns to improve quality prior to processing.
- Image Pre-Processing & Enhancement: Enhance contrast, remove distortion, and prepare data for analysis.
- Image Normalization: Normalize intensity values to minimize intra-class variation.
- Fingerprint Segmentation: Segment foreground (fingerprint) from background.
- ROI Detection: Concentrate only on meaningful regions holding actual fingerprint information.
- Fingerprint Image Dataset: Serves as the master data store supplying data to the deep learning model.
- Minutiae Extraction (AGNN Classifier): Extract minutiae features (positions and orientations) using an Attention-Guided Neural Network.
- Wavelet-Based Trivialities Detection: Feature refinement and elimination of spurious minutiae using wavelet feature analysis.
- Output Results: Produce minutiae maps containing both location and orientation predictions at lower resolution.

4. SYSTEM IMPLEMENTATION

This part explains the implementation phase of the system where the fingerprint minutiae detection task is created by employing deep learning methods. The objective is to extract minutiae positions and orientations accurately from contactless fingerprint images using a learned neural network. The implementation has two parts: Implementation Tools and System Walk through.

Implementation Tools: We briefly describe the major tools used for developing the fingerprint minutiae extraction system: Python: The main programming language employed for model building, training, and testing.
Tensor Flow / Py Torch: Deep learning frameworks used to build and train the Contactless Minu Net architecture for minutiae detection.
Open CV: A computer vision library used for image preprocessing, such as resizing, normalization, and visualization of fingerprint images and extracted minutiae points.
NumPy: Used for numerical operations, particularly for handling image data and processing location and orientation maps.
Labeling Tools: Custom or existing annotation tools to create ground truth location and orientation maps for training images.
Matplotlib: Visualization library used for plotting training progress and displaying extracted minutiae points.
CUDA and cuDNN: NVIDIA libraries to enable GPU acceleration during model training for faster computation.

Walkthrough the System: In this part, we describe the workflow and components involved in the fingerprint minutiae extraction system:

Training the Contactless Minu Net Model: A collection of fingerprint images with annotated minutiae (location and orientation) is prepared. The Contactless MinuNet deep network is trained using supervised learning, optimizing it to predict minutiae location and orientation maps from input images. Ground truths are downscaled to W4×H4×1\frac{W}{4} \times \frac{H}{4} \times 14W×4H ×1 dimensions for efficient learning.

Testing Phase: Inference is performed on unseen contactless fingerprint images. The trained model outputs predicted Location Maps and Orientation Maps. Post-processing techniques are applied to map and extract the minutiae points back to the original resolution of the fingerprint image using a stride of 4.

Thresholding and Non-Maximum Suppression: Thresholding is applied to the Location Map to remove low-confidence minutiae candidates. Then non maximum suppression is applied in order to keep only the local maxima points, so that minutiae located very close together are not repeated. Thresholding and Non-Maximum Suppression: Thresholding is performed over the Location Map in order to exclude low-confidence minutiae candidates. Subsequent to that, non-maximum suppression is done so as to maintain only local maxima points and to exclude duplication of densely spaced minutiae.

Mapping to Original Resolution: The found minutiae points from the downscaled maps (W/4 × H/4) are projected back onto the original size of the fingerprint image (W × H) by multiplying coordinates with the stride factor (usually 4). Orientation Assignment: Each detected minutia point is assigned an orientation value extracted from the corresponding Orientation Map.

Output Format: The final minutiae set for each fingerprint consists of (x, y, θ) triplets, where (x, y) are the pixel coordinates and θ is the minutiae angle.
Evaluation Metrics to assess the performance of the minutiae extraction system:

- **Detection Rate:** The ratio of ground-truth minutiae accurately detected within a tolerance window (e.g., 15 pixels in position and 20 degrees in orientation).
- **False Detection Rate:** The number of predicted minutiae not associated with any ground-truth minutia, normalized by the total number of predictions.
- **Precision, Recall, and F1-Score:** Precision indicates how many correct detected points were counted, recall indicates how many of the ground truth points were detected, and the F1-score offers a balance between the precision and the recall.

System Requirements Hardware and software setup necessary to run the system effectively:

Hardware:
- GPU: NVIDIA GPU (e.g., RTX 3060 or higher) with at least 6GB VRAM
- CPU: Intel i5/i7 or AMD Ryzen 5/7
- RAM: Minimum 16GB
- Storage: SSD recommended for faster read/write speeds.

Software:
- OS: Windows 10/11 or Ubuntu 20.04+
- Python 3.8+
- TensorFlow 2.x or PyTorch 1.x
- CUDA 11.x and cuDNN 8.x for GPU acceleration
- OpenCV, NumPy, Matplotlib installed via pip or conda

Limitations and Future Work:
Training Data Dependency: The accuracy of the model relies greatly on the presence and quality of labeled training datasets. Acquiring and labeling adequate contactless fingerprint data is challenging. Computational Resources: Substantial computational resources (high-end GPUs) are needed to train the model, which may not be available in all development environments.
Generalization to Extremely Noisy Inputs: Although strong, the existing model can still perform poorly in the presence of intense noise or intense occlusions. Future research involves incorporating attention mechanisms or multi-scale feature extraction methods to improve robustness further.
Lightweight Deployment: Investigation of lightweight model structures (e.g., variants of MobileNet) will enable deployment on devices with limited resources such as smartphones and embedded systems.

TABLE 1. Data Collection

Data Type	Description	Storage Location	Security
Training Data	Fingerprint images, Minutiae labels	Local storage (during training)	Access controlled
Model Weights	Trained Contactless Minu Net model	Secured server or cloud	Encrypted storage
Testing Data	New fingerprint images	Local or external datasets	Usage governed by protocols

Table The table above summarizes different data types involved in the system. Training and testing datasets are securely managed, and model weights are stored safely to prevent unauthorized access. The system ensures fast, accurate, and contactless minutiae extraction, which can be further integrated into broader biometric identification or authentication frameworks.

5. RESULT & DISCUSSIONS

This section discusses the testing and evaluation of the fingerprint minutiae detection system based on deep learning. Testing is one of the most critical stages of the project life cycle, ensuring that the model performs accurately and consistently in extracting minutiae from contactless fingerprint images.

- Encryption: C=E (K, P)
- Decryption: P=D (K, C)
- K: 256-bit encryption key (could be generated for model security or data protection).
- P: Plaintext (predicted minutiae data).
- C: Ciphertext (encrypted output stored or transmitted).

To evaluate the system performance:

- **Training and Validation Loss** were monitored across epochs to ensure model convergence without overfitting.
- **Testing Accuracy** was measured by comparing predicted minutiae points against ground-truth annotations.
- **Inference Speed** (Frames Per Second, FPS) was tested to ensure real-time or near-real-time applicability.

TABLE 2. Comparative Analysis

Aspect	Traditional Fingerprint Systems	Our Deep Learning Approach
Feature Extraction	Hand-crafted (ridge endings, bifurcations manually)	Automatically learned via deep networks (Contactless Minu Net)
Image Processing	Manual filters and enhancements	End-to-end deep learning processing
Minutiae Detection	Algorithmic (rule-based)	End-to-end deep learning processing
Adaptability	Limited to specific datasets	Predictive (learned feature mapping); highly adaptable
Tools Used	Open CV, MATLAB	Deep learning frameworks (e.g., Tensor Flow, PyTorch)

The table above highlights key differences between traditional minutiae extraction methods and the proposed deep learning-based approach. Traditional methods often rely on manual engineering and are sensitive to noise and variations. In contrast, our approach leverages deep learning for automatic, scalable, and robust feature extraction.

6. CONCLUSION

Fingerprint biometric systems have received much prominence in secure identification and authentication. Here, we presented a deep learning-based system for minutiae extraction from contactless fingerprint images based on the Contactless Minu Net model. By learning the location and orientation maps of minutiae, our system supports highly efficient and accurate extraction of important fingerprint features required for matching and verification processes. The suggested approach overcomes some of the major challenges related to conventional minutiae extraction methods, including low-quality images, distortions, and partial fingerprints. Through the use of a fully convolutional network, the system can automatically learn to handle different fingerprint patterns without any feature engineering. Our testing process proved that the trained network works well even on low-quality images, providing accurate minutiae mapping with minimal post-processing. Even with the encouraging results, the system continues to struggle with generalization across varied fingerprint datasets, computational complexity for real-time applications, and explainable AI models for forensic acceptability. Future research will address enhancing the scalability of the model, incorporating lightweight architectures for mobile deployment, and investigating fusion with cryptographic techniques for secure fingerprint template protection. Through ongoing research and optimization, deep learning-based minutiae detection can greatly improve biometric authentication systems in security as well as reliability.

REFERENCES

[1]. Hariharan B, Malik J, Ramman D. Discriminative decorre lation for clustering and classification. 2012.
[2]. Wieclaw L. A minutiae–based matching algorithms in fin gerprint recognition system. Journal of Medical Informatics and Technologies. 2009; 13:1–8.
[3]. Mary Lourde R, Khosla D. Fingerprint identification in bio metric security systems. International Journal of Computer and Electrical Engineering. 2010 Oct; 2(5):852–5.
[4]. K. Cao, E. Liu and A. K. Jain, "Segmentation and enhancement of latent fingerprints: A coarse to fine ridge structure dictionary", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 9, pp. 1847-1859, 2014.
[5]. H. Walhazi, L. R. Haddada, A. Maalej and N. E. B. Amara, "Preprocessing latent-fingerprint images for improving segmentation using morphological snakes", 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1-6, 2020.
[6]. Y. Zhu, X. Yin, X. Jia and J. Hu, "Latent fingerprint segmentation based on convolutional neural networks", 2017 IEEE Workshop on Information Forensics and Security (WIFS), pp. 1-6, 2017.
[7]. P. B. S. Serafim, A. G. Medeiros, P. A. Rego, J. G. R. Maia, F. A. Trinta, M. E. Maia, et al., "A method based on convolutional neural networks for fingerprint segmentation", 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2019.
[8]. Gonzalez RC, Woods RE. Digital image processing. Pearson Education Asia. 2002.
[9]. Pankanti S, Prabhakar S, Jain AK. On the individuality of f ingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002; 24(8):1010–25.
[10]. M. Liu, S. Liu and W. Yan, "Latent fingerprint segmentation based on ridge density and orientation consistency", Security and Communication Networks, vol. 2018, pp. 4529652, May 2018.
[11]. N. Ratha, S. Chen and A. K. Jain, Adaptive go with the drift orientation primarily based totally texture extraction in fingerprint photographs, Pattern Recognition, 28, 1657-1672 (1995).
[12]. L. Prem k, Z. Emeršic and T. Oblak, "Automatic latent fingerprint segmentation using convolutional neural networks" in 2021 44th International Convention on Information Communication and Electronic Technology (MIPRO), IEEE, pp. 1010-1014, 2021.
[13]. Jain AK, Prabhakar S, Hong L, Pankanti S. Filter bank based fingerprint matching. IEEE Transactions on Image Processing. 2000; 9:846–59. PMid: 18255456.
[14]. Z. Zhang, Q. Liu and Y. Wang, "Road extraction by deep residual u-net", IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 5, pp. 749-753, 2018.
[15]. G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.