Comparative Analysis between Generalized Hough Transform and Neural Networks in Fingerprint Recognition

Ali ZERBO[1] **and Abdoulaye SERE**[2]

[1]PhD student, Burkina Faso
[2]Senior lecturer , Burkina Faso

*E-mail : alizerbo98@gmail.com

Abstract

This study evaluates and compares the performance of two prominent fingerprint recognition methods: the Generalized Hough Transform (GHT) and Siamese neural networks. Using the diverse and well suited CASIA database, we conducted an exhaustive comparison on a representative sample of 628 fingerprints per model. Siamese neural networks demonstrated exceptional accuracy, surpassing 99.2%. Their efficiency in processing large datasets highlights their suitability for biometric applications requiring detailed analysis. In contrast, the Generalized Hough Transform achieved a strong accuracy of 97.7%, proving its reliability even with more constrained data. Our analysis underscores the distinct strengths of each method, particularly the adaptability of Siamese networks to complex fingerprint variations. Their robust generalization capabilities make them ideal for large-scale biometric systems. Performance was assessed using key metrics, including the confusion matrix (visualizing predicted vs. actual matches) and the F1-score (balancing precision and recall). These findings advance our understanding of biometric security challenges and suggest promising directions for future research in this rapidly evolving field

Keywords

Hough Transform; confusion matrix; F1-score

I INTRODUCTION

Fingerprint recognition remains a cornerstone of modern biometric systems, critically supporting identification and authentication across diverse applications ranging from mobile device security to forensic investigations. As reliance on these systems grows, the scientific community continues to pursue enhanced methods that improve both accuracy and operational efficiency. Current approaches span traditional geometric feature-based techniques, such as the Hough transform, to advanced deep learning architectures including convolutional neural networks (CNNs) [12][6] .

Our study focuses on comparing two particularly significant yet fundamentally different approaches: the Generalized Hough Transform (GHT) and Siamese neural networks. This selection is motivated by their complementary advantages in addressing key fingerprint recognition challenges. While conventional CNNs excel at feature extraction, they typically require extensive labeled datasets and are less optimized for pairwise matching tasks. Siamese networks, by contrast, are specifically designed for similarity comparison, making them exceptionally suited

for fingerprint verification where assessing input pairs is essential [7]. The generalized Hough transform offers distinct benefits of computational efficiency and noise robustness, maintaining relevance for applications where interpretability and resource constraints are paramount. This dual evaluation provides unique insights into how traditional and modern methods can address different operational requirements.

Using the comprehensive CASIA database selected for its diversity and scalability, we conduct a rigorous comparative analysis with three primary objectives: first, assessing each method's capability to handle complex fingerprint variations; second, evaluating their accuracy under practical conditions; and third, providing a detailed examination of their respective strengths and limitations.

Our evaluation employs robust metrics including confusion matrices for performance visualization and F1 scores to balance precision and recall, offering a nuanced understanding of each approach's effectiveness.

This systematic comparison not only advances our understanding of fingerprint recognition methodologies but also provides practical guidance for implementing context-appropriate solutions. By bridging traditional and cutting-edge approaches, our findings contribute valuable insights for developing next-generation biometric systems that must balance accuracy, efficiency, and scalability in real world applications.

II METHOD DESCRIPTION

2.1 The database

The choice of database is crucial in our study, as it serves as the primary testbed for evaluating the performance of fingerprint recognition methods. We opted for the CASIA database, renowned for its substantial size and carefully organized diversity.

The CASIA database is extensive, comprising a large collection of fingerprints from a variety of sources. It is organized into 500 folders numbered from "000" to "499," each representing a specific individual. Each subject has two subfolders, "L" (left) and "R" (right), each containing 20 fingerprint images [14]. This is illustrated in the figure below.

In total, the database comprises 20,000 fingerprint images, encompassing a wide range of conditions, subjects, and environments. The scale of CASIA plays a crucial role in obtaining robust and generalizable results, which are essential for evaluating performance across diverse scenarios. Equally important is its diversity, as it reflects the real-world conditions faced by fingerprint recognition systems. The fingerprints cover natural variations in age, gender, and ethnicity. In addition, the database captures different finger positions, varying illumination levels, and other environmental factors that contribute to its richness [13].

By opting for the CASIA database, we aim to ensure a comprehensive and representative evaluation of fingerprint recognition methods. This selection takes into account the complexity of real-world conditions, providing a solid basis for relevant and applicable conclusions.

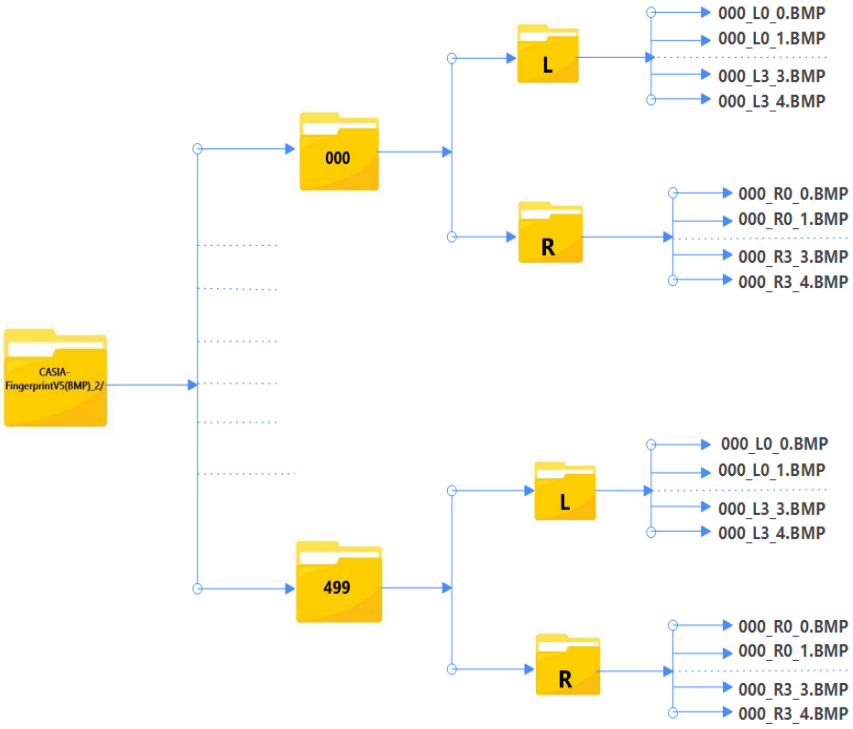

Figure 1: Structure of the CASIA-FingerprintV5(BMP)_2 database.

2.2 Hough Transform

The Generalized Hough Transform (GHT) plays a crucial role in fingerprint recognition by detecting and extracting specific features such as minutiae from an image. It builds on the core principle of the original Hough transform, developed for line detection, and extends it to more complex shapes (e.g., curves and circles), making it particularly suitable for fingerprint analysis [4].

In this section, we detail the implementation of the Generalized Hough Transform and explain how its generality aligns with our study. We begin with data preprocessing, during which fingerprint images undergo intensity normalization, segmentation, and binarization to reduce noise. The Zhang–Suen thinning algorithm is then applied to reduce ridge thickness to a single-pixel width, thereby simplifying minutiae extraction.

a) Data preprocessing

Fingerprint impressions were subjected to several preprocessing techniques [1][8] including intensity normalization, segmentation, and binarization to suppress noise. Subsequently, the Zhang–Suen thinning algorithm [3] was applied to each block to reduce ridge thickness to a single-pixel skeleton, thereby simplifying minutiae extraction.
An overview of the stages of fingerprint image preprocessing is provided below.

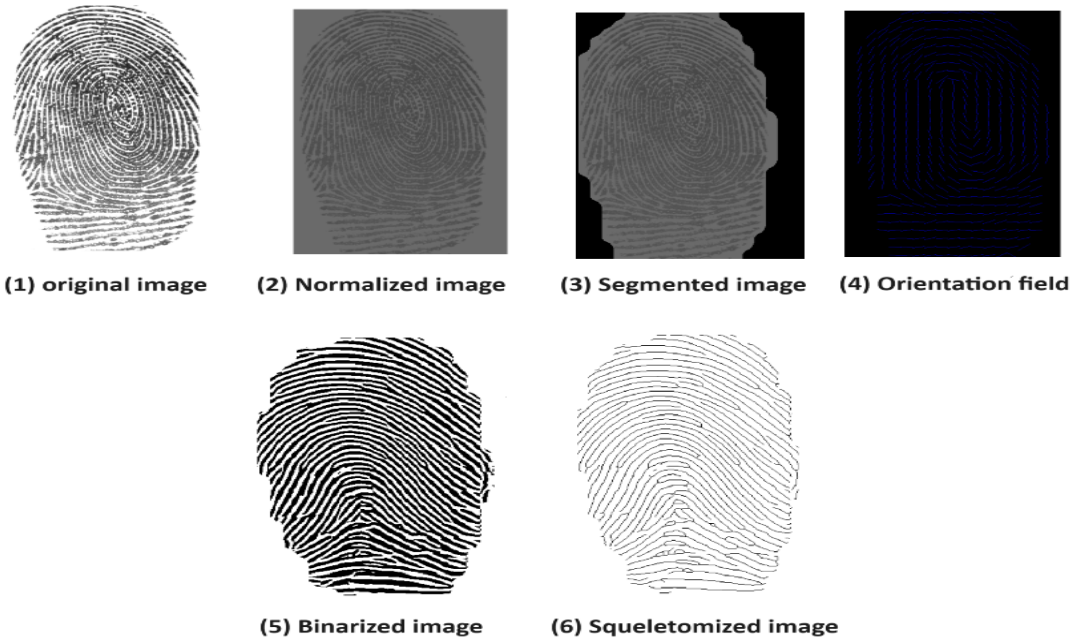

Figure 2: Stages of fingerprint image preprocessing.

b) Minutiae extraction

The minutiae are unique characteristics of the fingerprint that allow for its identification and verification [2]. Each minutia m_i is represented by a quadruplet of coordinates: $m_i = (x_i, y_i, \theta_i, t_i)$ [10]. The coordinates x_i and y_i correspond to the position of the minutia in the image, θ_i represents its direction, which is generally calculated from the local orientation of the ridge, and t_i indicates whether the minutia is a termination or a bifurcation type. Bifurcations and terminations are key points for comparing fingerprints.

For each black pixel in each block of the fingerprint, we calculate the Crossing Number (CN) indicator [2], which is a measure of local topology. The formula for the CN indicator is as follows:

$$CN = \frac{1}{2}\sum_{i=1}^{8}|p_i - p_{i+1}| \qquad \text{with } p_9 = p_1 \tag{1}$$

The neighboring pixels $p_1, ..., p_9$ are arranged in a clockwise order. Using the properties of the CN, as summarized in the table above, each ridge pixel can be classified as either a termination, a bifurcation, or a point without a minutia. Specifically, an isolated pixel will have a CN of 0, a termination will have a CN of 1, a continuous ridge will have a CN of 2, a bifurcation will have a CN of 3, and a crossing will have a CN of 4 [2]. This information is essential for minutiae extraction and fingerprint comparison.

c) Minutia orientation estimation

To estimate the orientation of a minutia m, we seek to find the line that passes through m and best fits the surrounding connected points. We use a simple linear regression method [13]

for this purpose. By then applying the $atan$ function to the slope of the line, we obtain the orientation of the minutia.

d) Comparison of minutiae

After the preprocessing phase of the fingerprint image, we use an adapted Generalized Hough Transform (GHT) algorithm, followed by a matching algorithm to detect and connect the corresponding feature points. The adapted Generalized Hough Transform (GHT) algorithm uses mathematical equations in a parameter space to determine the spatial relationships between the feature points. These parameters describe the characteristics of the spatial relationships, such as the distance and angle between the minutiae .

The algorithms for the adapted Generalized Hough Transform (GHT) and the matching are illustrated below, respectively:

Algorithm 1 Generalized Hough Transform Algorithm

Require: Two sets of minutiae S_1 and S_2
Ensure: Parameter transformation (peak position)
Initialize accumulation matrix $A[x][y] \leftarrow 0$
for each minutia $M_1 \in S_1$ **do**
  for each minutia $M_2 \in S_2$ **do**
    $\delta_{angle} \leftarrow \min(|\theta_{M_1} - \theta_{M_2}|, 360 - |\theta_{M_1} - \theta_{M_2}|)$
    $angle_{diff} \leftarrow 360 - \delta_{angle}$

    $\Delta x \leftarrow x_{M_1} - (x_{M_2} \cdot \cos(angle_{diff}) - y_{M_2} \cdot \sin(angle_{diff}))$
    $\Delta y \leftarrow y_{M_1} - (x_{M_2} \cdot \sin(angle_{diff}) + y_{M_2} \cdot \cos(angle_{diff}))$

    $A[\Delta x][\Delta y] \leftarrow A[\Delta x][\Delta y] + 1$
  end for
end for
return $\arg\max_{x,y} A[x][y]$

Algorithm 2 Minutiae matching

Require: Two minutiae sets E_1 and E_2, thresholds τ_1 and τ_2
Ensure: List of minutiae pairs $\mathcal{P}$
Initialize $\mathcal{P} \leftarrow \emptyset$, *count* $\leftarrow 0$
for each $M_1 \in E_1$ **do**
  for each $M_2 \in E_2$ **do**
    if $|\Delta x| < \tau_1$ **and** $|\Delta y| < \tau_1$ **and** $\theta_{\text{diff}} < \tau_2$ **then**
      count $\leftarrow$ *count* $+ 1$
      $\mathcal{P}$[*count*] $\leftarrow (M_1, M_2)$
    end if
  end for
end for
return $\mathcal{P}$

III NEURAL NETWORK MODEL FOR IMAGE COMPARISON

Remarkable advances in artificial intelligence have produced a wide range of neural network architectures specialized for specific tasks. For fingerprint verification and image comparison, Siamese models have emerged as a particularly effective approach [4,6]. The model described below follows this paradigm, adopting a Siamese architecture for image comparison and thereby leveraging convolutional feature learning.

Our methodology employs a neural network specifically designed for fingerprint recognition. The choice of this model detailed below stems from considerations aimed at optimizing recognition performance across diverse conditions. Before delving into the network's characteristics, we emphasize the central role of methodology in evaluating fingerprint recognition systems.

To make the methodological approach more accessible, a detailed schematic diagram is provided below.

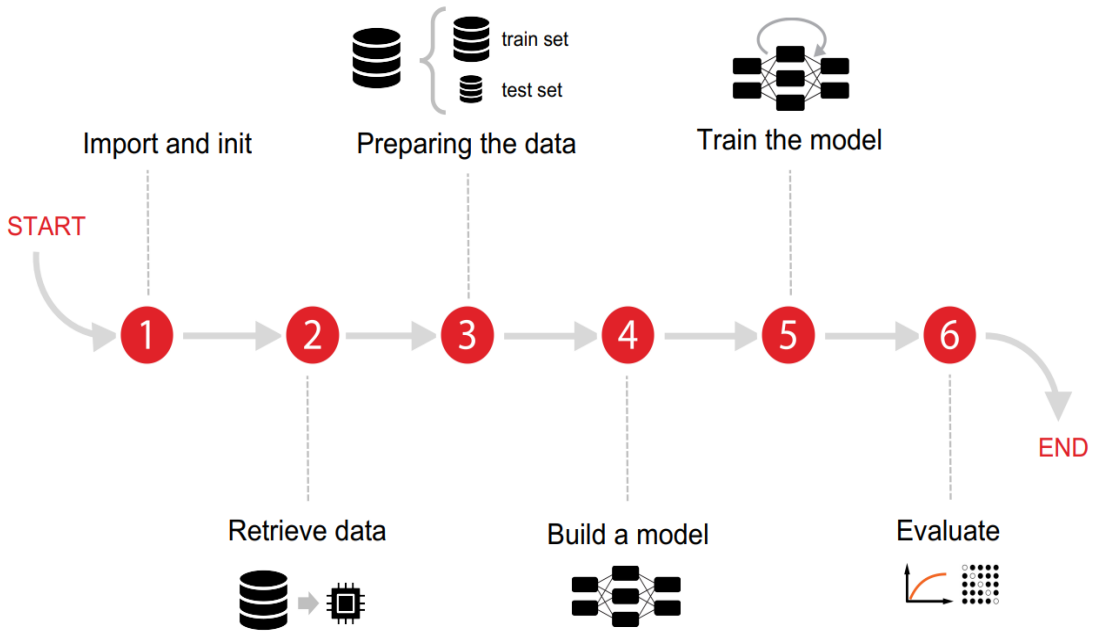

Figure 3: Diagram.

This diagram gives readers a clear view of the stages of the process from data preparation through the crucial minutiae matching stage. Including the diagram offers a high-level overview that clarifies the interactions among the components of our methodology.

3.1 Model architecture

The fingerprint-matching model adopts a Siamese architecture with two weight-sharing branches that process a pair of fingerprint images (Figure 4). Each branch encodes its input into a feature embedding, and a similarity measure computed from the two embeddings determines whether the prints match, with sensitivity to subtle differences [7].

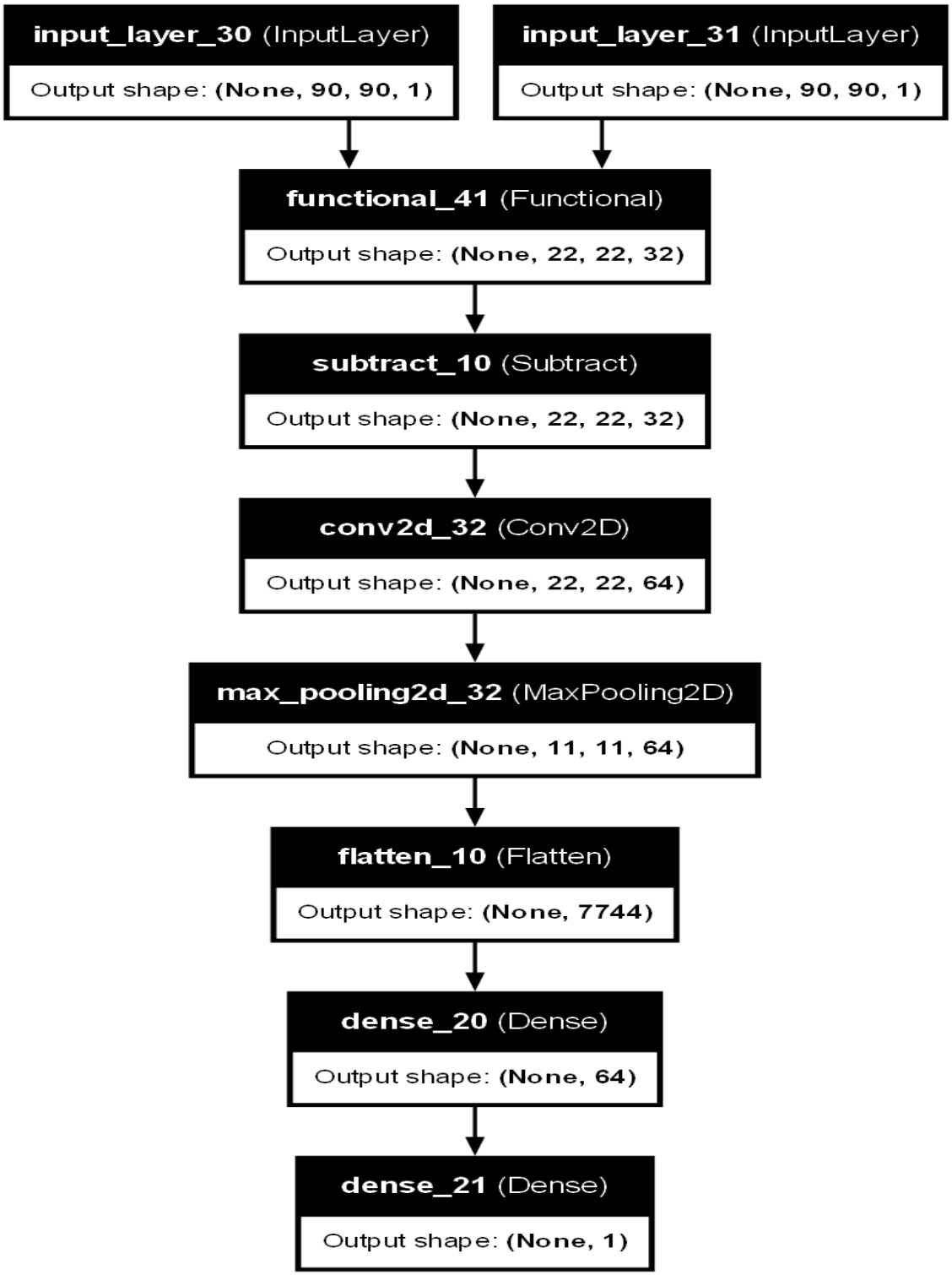

Figure 4: Model architecture.

3.2 Feature extraction

The first stage of the model involves feature extraction from fingerprint images. This phase uses convolution and max-pooling operations to detect significant patterns while reducing the dimensionality of the data. The special feature lies in the sharing of weights between the two input branches, ensuring consistent feature extraction.

3.3 Feature Subtraction

Once features have been extracted, an element-wise subtraction is applied between the two branches. This step reveals pixelwise differences between the fingerprints, thereby emphasizing regions of divergence.

3.4 Classification

The resulting features are then passed through classification layers, including additional convolutional operations. These decision layers estimate the probability of similarity between the fingerprints. The sigmoid activation in the output layer produces a value in $[0, 1]$ that represents the similarity score.

3.5 Data Augmentation Training

To improve robustness, we integrated a data generator during training. This generator applies various augmentation techniques such as gaussian blur and geometric transformations to diversify the dataset. In our setup, we increased the number of images fivefold per fingerprint. This enlargement further enriches the dataset, giving the model greater variety and helping it generalize in the presence of fingerprint variations, which in turn improves robustness and recognition performance in real world conditions. An illustration of the augmentation pipeline is shown in Figure 5.

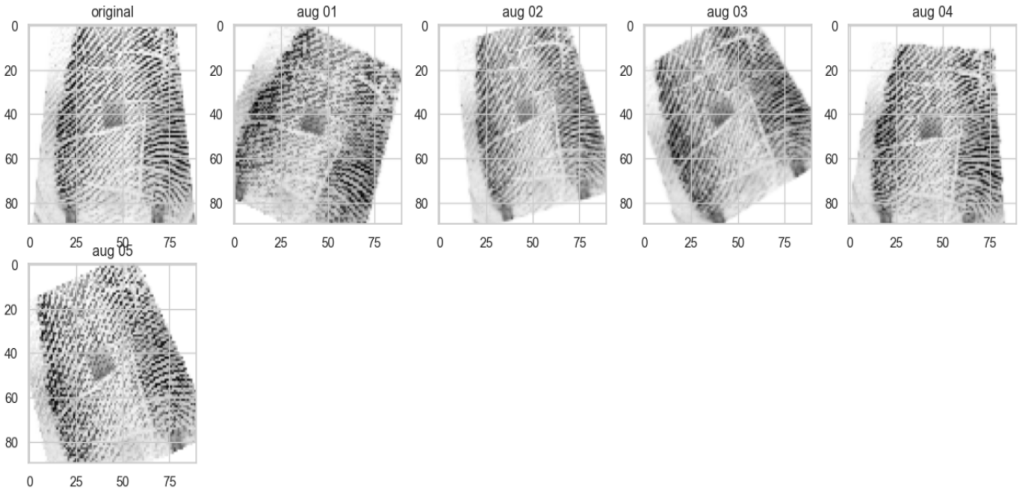

Figure 5: Data Augmentation.

With regard to the model parameters, the activation function used is ReLU (Rectified Linear Unit) for the convolutional layers, except for the output layer, which uses the sigmoid activation function. The optimizer used is Adam, an algorithm commonly used in deep learning, with the default learning rate. For the loss function, we opted for the binary cross-entropy loss, adapted to binary classification tasks. Finally, the batch size used for training is 563, determining the number of samples processed simultaneously before the model weights are updated. We also trained the model over 6 epochs, a number chosen to ensure adequate convergence while avoiding overfitting. Summaries of the model architecture and hyperparameters are provided below (Tables 1–3).

Layer	Type	Size	Output	Params
Input	InputLayer	-	90×90×1	0
Conv2D_1	Conv2D	3×3	90×90×32	320
MaxPool_1	MaxPool2D	2×2	45×45×32	0
Conv2D_2	Conv2D	3×3	45×45×32	9,248
MaxPool_2	MaxPool2D	2×2	22×22×32	0
			Total:	**9,568**

Table 1: Architecture of the shared feature extraction block

Stage	Description	Output Shape	Parameters
Inputs	Two 90×90 grayscale images	(None, 90, 90, 1) × 2	0
Feature Extraction	Shared `feature_model` application	(None, 22, 22, 32) × 2	9,568
Subtraction	Element-wise difference	(None, 22, 22, 32)	0
Conv2D	Convolution on difference	(None, 22, 22, 64)	18,496
MaxPooling	Spatial reduction	(None, 11, 11, 64)	0
Flatten	Vectorization	(None, 7,744)	0
Dense_1	Intermediate dense layer	(None, 64)	495,680
Dense_2	Binary output	(None, 1)	65
		Total parameters:	**523,809**

Table 2: Complete Siamese neural network architecture

Category		Value
Architecture	Input size	90×90×1
	Convolution kernels	3×3
	Pooling	2×2 MaxPooling
	Filters	$32 \rightarrow 32 \rightarrow 64$
	Padding	'same'
Training	Loss function	binary_crossentropy
	Optimizer	adam
	Metric	accuracy

Table 3: Siamese network hyperparameters

IV RESULTS

In our study, we used two distinct methods to analyse an image dataset: Generalized Hough Transform (GHT) and the neural network.

For Generalized Hough Transform, we randomly selected 628 images from our main dataset. This selection was used to create two subsets of the data, Data1 and Data2, containing 500 and 128 images respectively. Note that 64 images are common to these two subsets. We then compared the fingerprints in Data2 with those in Data1.

For the neural network approach, we divided our dataset (20,000 fingerprint images) by allocating 90% of the images to model training and the remaining 10% to validation. After training, a random sample of 128 images was selected for further testing. These tests were used to compare the results obtained with those of the Hough transformation.

These evaluations were carried out on a Lenovo Ideapad Gaming 3 laptop, equipped with 16 GB of RAM and a 10th-generation Intel Core i5 processor running at 4.5 GHz. Python (version 3.11.7) was used as the main development environment for implementing the machine learning

models. TensorFlow (version 2.15.0) was used to create, train and evaluate neural networks, in particular for the Siamese neural network method. Anaconda was used as the development environment, with Jupyter Notebook for executing and presenting the Python code. All these operations were carried out on the same machine to ensure consistency of experimental conditions.

Thanks to these rigorous methodological approaches, we have obtained precise and reliable results, highlighting the distinct performance of each analysis method. The results are presented in the table 1.

These results are also illustrated using a confusion matrix and a diagram, providing a clear visualisation of the performance of each method.

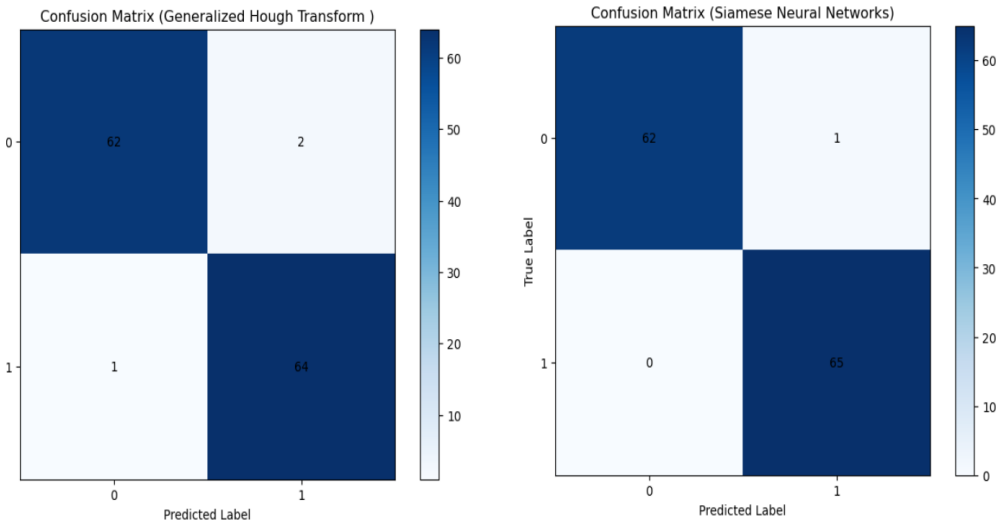

Figure 6: Confusion matrix.

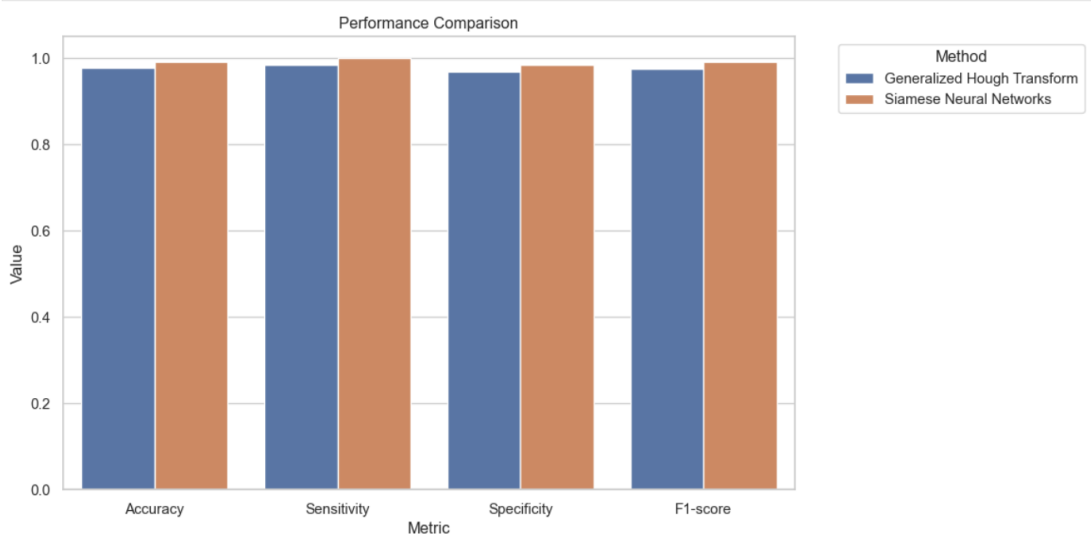

Figure 7: grouped bar graph of analysis method performance.

	Accuracy	Sensitivity	Specificity	F1-score
Generalized Hough Transform	0.977	0.984	0.969	0.976
Siamese Neural Networks	0.992	1.0	0.984	0.992

Table 4: Performance comparison of analyzed methods.

V DISCUSSION

5.1 Performance comparison

In this section, we compare the performance of the Generalized Hough Transform and the Siamese network based on the results obtained.

Our findings indicate that the Siamese network slightly outperforms the Generalized Hough Transform. It achieves higher accuracy (0.992 vs. 0.977) and perfect sensitivity (1.000 vs. 0.984), reflecting an excellent ability to correctly identify positive matches. Specificity remains comparable (0.984 for the Siamese model vs. 0.969 for Hough). Finally, the F1 score which balances precision and recall is also higher for the Siamese network (0.992 vs. 0.976).

Although the Siamese network is generally superior, the optimal choice depends on constraints related to data, computational resources, and interpretability requirements. Generalized Hough Transform (GHT) does not require labeled data, has a very small memory footprint, runs efficiently on CPU only hardware, and offers strong interpretability (votes/peaks in parameter space) all crucial for auditability and deterministic behavior. Conversely, a Siamese network requires labeled pairs and a machine-learning runtime, but typically provides greater robustness to noise, occlusions, and pose variations especially when trained with data augmentation. It also scales better to large databases by comparing compact vector representations (embeddings) via approximate nearest-neighbor search. In practice, prefer GHT under stringent resource or transparency constraints, and favor the Siamese approach when accuracy under diverse real-world conditions and scalability are paramount. A pragmatic compromise is a cascaded design: use fast geometric pre-filtering (GHT) to reject obvious non-matches or estimate coarse alignment, then apply the Siamese verifier to the remaining candidates to optimize the compute accuracy trade-off.

Under real-time constraints (decisions within a few tens of milliseconds) and limited resources (CPU-only, battery constrained), GHT remains highly competitive: predictable compute cost, minimal memory, no ML runtime, and deterministic behavior. In practice, one bounds the parameter space (angles, distances/radii), restricts processing to a region of interest (ROI), and applies early reject heuristics to meet strict service level objectives (SLOs). A Siamese network can also satisfy real-time requirements provided it is optimized: quantization (e.g., INT8), reduced input resolution, lightweight architecture (depthwise-separable convolutions), and early exit thresholds. With an embedded GPU/NPU, inference becomes very fast; on CPU, latency remains acceptable if the model is compact.

In structured environments (standardized acquisition: slap/roll scanners, glass platen, stable illumination, controlled pose and resolution), ridge geometry is predictable and intrinsic image quality is high. GHT then benefits fully from tightened parameter ranges, which reduces the accumulator size, accelerates decision-making, and strengthens interpretability (sharp peaks in the voting space), with low maintenance (no training or recalibration). This setting suits access kiosks, administrative counters, or industrial devices. Siamese networks often retain a

slight accuracy edge, but the gap may become marginal in highly controlled conditions; that gain should be weighed against deployment complexity (ML runtime, model updates). Thus, in highly structured environments with strong auditability requirements, GHT is preferable; where residual variability is present (partial/wet fingers, micro shifts) or when maximal accuracy is required, the Siamese network is the better choice.

5.2 Limitations of the study

o interpret our results appropriately, it is important to consider the study's limitations. First, although the CASIA-FingerprintV5 (BMP) dataset contains 20,000 fingerprint images, it captures only a fraction of the diversity found in human fingerprints. Models trained here may therefore exhibit biases and may not fully generalize to more heterogeneous datasets. For example, certain demographic groups or acquisition conditions may be underrepresented. To mitigate this, one could employ data augmentation (e.g., rotation, elastic distortion, contrast variation), incorporate more diverse datasets, and perform cross-dataset validation.

Second, we evaluated only two techniques (GHT and Siamese networks). Other approaches such as triplet-based networks, graph based methods, or Transformer-based architectures may offer different or complementary perspectives. Our conclusions should therefore not be assumed to generalize to all fingerprint identification or verification methods.

Third, the observed performance can be sensitive to training choices: hyperparameters, loss functions, optimization methods, the strategy for generating pairs (positive/negative), and the selection of a decision threshold. A systematic exploration (sensitivity analysis, hyperparameter search) would be necessary to more fully characterize model robustness.

Finally, experimental conditions hardware, software libraries, framework versions, and random seeds can also affect results. These factors should be accounted for when interpreting performance and assessing real world applicability. Using reproducible environments (e.g., containers), standardized benchmarks, fixing random seeds, and reporting confidence intervals can further strengthen the reliability of the conclusions.

VI CONCLUSION

In conclusion, this comparative study examined the performance of the Generalized Hough Transform (GHT) and Siamese neural networks for fingerprint recognition and highlighted significant results. The Siamese model achieved higher accuracy (0.992 vs. 0.977) and a higher F1 score (0.992 vs. 0.976) than GHT, underscoring its effectiveness in this setting.

This comparison offers practical guidance by clarifying the strengths and limitations of each approach, which can inform the design and adoption of more accurate and robust biometric systems that enhance security and privacy across applications. Nonetheless, challenges remain improving robustness to variable fingerprint quality and scaling to large data volumes among them so further work is needed to address these issues and deliver more effective solutions.

To move beyond an either or choice, we propose a cascaded hybrid. GHT first provides a fast, interpretable pre selection of candidates (bounded parameter space, geometric score) and, if needed, a light geometric realignment; the shortlisted pairs are then refined by the Siamese network, which outputs a match probability. Final decisions can use dual thresholds (reject/accept/gray zone) and late score fusion (e.g., weighted or logistic calibration of the GHT score

with the network output). This strategy preserves GHT's frugality and transparency while leveraging the learned model's robustness. At very large scales, the order can be inverted: retrieve top-K candidates in the Siamese embedding space and validate them geometrically with GHT. Future evaluations should quantify the overall benefit (F1, ROC–AUC/EER) alongside latency and resource constraints.

VII GLOSSARY OF TECHNICAL TERMS

Term	Description
Minutiae	Local ridge characteristics (e.g., terminations, bifurcations) used to describe and compare fingerprints.
Termination / Bifurcation	Types of minutiae: a *termination* (end of ridge) or a *bifurcation* (division into two branches).
Ridge orientation (theta)	Local direction of ridge flow around a point/minutia; used to guide matching and normalization.
Crossing Number (CN)	Local topological measure on a 3×3 neighborhood to classify pixels (non-minutia, termination, bifurcation, etc.). Formula: CN $\approx \frac{1}{2}\sum_{i=1}^{8} \lvert p_i - p_{i+1} \rvert$ with $p_9 = p_1$.
Generalized Hough Transform (GHT)	Voting-based method accumulating geometric evidence in a parameter space (Hough) to detect shapes or correspondences.
Accumulator (Hough space)	Discrete parameter grid where votes are tallied; peaks indicate the most probable geometric alignments.
Region of Interest (ROI)	Sub-part of the image containing the relevant fingerprint area, used to focus computations.
Siamese neural network	Two-branch network sharing the same weights, producing comparable *embeddings* to estimate similarity.
Embedding	Compact feature vector produced by a network to represent an image (for comparison or indexing).
Absolute difference / Subtraction	Elementary operations between two feature maps: $\lvert A - B \rvert$ (order-invariant) or $A - B$ (order-sensitive).
Precision	$\frac{\text{TP}}{\text{TP+FP}}$: proportion of correctly predicted matches.
Recall (Sensitivity)	$\frac{\text{TP}}{\text{TP+FN}}$: proportion of true matches correctly detected.

Specificity	$\frac{\text{TN}}{\text{TN+FP}}$: proportion of non-matches correctly rejected.
F1-score	Harmonic mean of precision and recall: $F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.
ROC–AUC / EER	**ROC–AUC**: area under the ROC curve (threshold-independent performance). **EER**: equal error rate (false acceptances = false rejections).
Approximate Nearest Neighbor (ANN) search	Fast search for the K nearest neighbors in large databases (sub-linear time).
Quantization (INT8)	Representation of weights/activations on 8 bits to reduce memory and latency (minor precision loss).
NPU	Neural Processing Unit; hardware accelerator optimized for neural network inference on embedded devices.
Data augmentation	Transformations applied to training images (blur, rotation, contrast, etc.) to improve generalization.
Batch size / Epoch	**Batch size**: number of samples processed before an update. **Epoch**: one complete pass through the training set.
ReLU / Sigmoid	Activation functions: ReLU for hidden layers, Sigmoid for binary outputs (match probability).
Binary cross-entropy	Cost function for binary classification, comparing predicted probabilities to actual labels.
Adam optimizer	Adaptive stochastic gradient method commonly used for neural network training.

Table 5: Glossary of technical terms in fingerprint recognition and machine learning.

REFERENCES

Publications

[1] K. J. Anil, P. Salil, and P. Sharath. "Twin Test: On Discriminability of Fingerprints". In: *International Conference on Audio- and Video-Based Biometric Person Authentication* 2091.1629 (2001).

[2] W. Łukasz. "A minutae based matching algorithms in fingerprint recognition". In: *In medical informatics and technologies* 13.6 (2009).

[3] R. W. Anastasia. "Comparing Hilditch, Rosenfeld, Zhang-Suen,and Nagendraprasad Wang Gupta Thinning". In: *World Academy of Science, Engineering and Technology* 78.6 (2011).

[4] P. Mukhopadhyay and B. B. Chaudhuri. "A survey of Hough Transform". In: *Pattern Recognition* 48.1–3 (2015).

[5] V. C. Jane, P. Jianmin, N. R. Shesh, and G. Susan. "ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves". In: *Surgery* 159.6 (2016).

[6] A. Mouad, H. M. Vivek, P. Yannawar, and A. T. Gaikwad. "Overview of Fingerprint Recognition System". In: *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (2016).

[7] D. Chicco. "Siamese Neural Networks: An Overview". In: *Artificial Neural Networks(Methods in Molecular Biology)* 2190 (2020).

[8] G. M. Rashmi, K. Deepti, and G. Rubén. "Fingerprint image enhancement and reconstruction using the orientation and phase reconstruction". In: *Information Sciences* 530 (2020).

[9] J. W. Wong and S. Lai. "Multi-task CNN for restoring corrupted fingerprint images". In: *Pattern Recognition* 101.2 (2020).

[10] K. Abdullah, L. Asif, and A. Shafique. "Machine Learning in Computer Vision: A Review". In: *EAI Endorsed Transactions on Scalable Information Systems* 8.32 (2021).

[11] G. Jochen, H. Fred, M. Dominik, W. Kanit, R. Donghao, N. Rahul, K. Marc, and P. Kayur. "Neo: Generalizing Confusion Matrix Visualization to Hierarchical and Multi-Output Labels". In: *CHI '22: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* 408 (2022).

[12] C. Mahmud and I. Masudul. "Contactless Fingerprint Recognition Using Deep Learning—A Systematic Review". In: *Journal of Cybersecurity and Privacy* 2.1–3 (2022), pages 714–730.

[13] S. Sujana and K. S. v. Reddy. "Multi-modal Biometric System for Face and Fingerprint using Convolutional Neural Network". In: *International Conference on Applied Electromagnetics, Signal Processing and Communication (AESPC)* (2022).

[14] *Database (CASIA benchmark)*. `http://www.idealtest.org/`. Accédé : 2024-01-29.