A ROBUST FINGERPRINT RECOGNITION SYSTEM FOR SECURE BIOMETRIC AUTHENTICATION

Om Ghag, Student, Dept. Of Computer Engineering, Vidyalankar Institute of Technology, Mumbai University.
Anurag Kumbhare, Student, Dept. Of Computer Engineering, Vidyalankar Institute of Technology, Mumbai University.
Sarvesh Gaonkadkar, Student, Dept. Of Computer Engineering, Vidyalankar Institute of Technology, Mumbai University.
Tejas Kumbhar, Student, Dept. Of Computer Engineering, Vidyalankar Institute of Technology, Mumbai University.
Sneha Annappanavar, Associate Professor, Dept. Of Computer Engineering, Vidyalankar Institute of Technology, Mumbai University.

ABSTRACT

Convolutional Neural Networks (CNNs) have been a key framework for image- related tasks such as fingerprint recognition systems since the advancements in deep learning and computer vision. Traditional methods such as Scale Invariant Feature Transform (SIFT), Oriented FAST, Rotated BRIEF (ORB), and CNNs are used to match fingerprints. However, in recent years, studies have shown that CNNs are hard to work with large and complex datasets. Although accurate, SIFT is computationally expensive, while ORB is fast but not very precise.

Due to these challenges, Siamese Neural Networks (SNNs) have been used for fingerprint-matching tasks. In contrast to traditional methods, SNNs learn the similarity between input fingerprints, which are suitable for verification and matching tasks. It is demonstrated that using SNNs can lead to improved finger- print pattern recognition in the presence of noise or partial prints, and they can even distinguish genuine from false fingerprints.

SNN-based fingerprint recognition system offers the advantages of pairwise instead of global comparison and more accurate and reliable fingerprint matching. SNNs, given their light structure, can be utilized on mobile security applications that require resource efficiency.

We evaluate the application of SNNs to mobile fingerprint recognition systems from a comprehensive point of view to better understand the mobile security context where SNNs can potentially be applied. The results show that SNNs are a good compromise between accuracy and computation and, hence, are a suitable solution.

Keywords:
Fingerprint Recognition, Siamese Neural Networks (SNNs), Convolutional Neural Networks (CNNs), Deep Learning in Biometrics, Mobile Security.

I. Introduction

Convolutional Neural Networks (CNNs) have been used mostly in feature extraction and pattern recognition tasks in fingerprint recognition systems. They have made strong performance on multiple biometric applications possible because of their ability to capture local spatial hierarchies in images. However, CNNs are inefficient in terms of computational efficiency and handle global contextual information, which are important for mobile security environments with restrictions regarding computational resources.

Fingerprint recognition utilizes this to tackle the problems by the use of Siamese Neural Networks (SNNs) in mobile security applications. SNNs are particularly well suited to tasks like fingerprint matching and verification because they can be formulated as a twin network architecture that learns similarity metrics between two inputs. In a challenging biometric environment, SNNs also offer additional advantages of having greater accuracy and robustness in comparing pairs of fingerprint images to determine whether the produced images are from the same individual. SNNs are particularly useful for mobile security, since fingerprint verification and most other biometric comparisons involve pairwise comparisons. Unlike traditional CNNs, SNNs have been built specifically to compare pairs of images at a time, and results confirm that they are robust to fingerprints that contain noisy or missing finger- prints. Additionally, SNNs have the right resource constrained nature appropriate to mobile platforms which permits computational demand to take the place of accuracy. The research reported in this paper demonstrates how SNNs architecture can address the specific problems of mobile security, and specifically in the context of mobile fingerprint recognition systems. Given high accuracy and speed, SNNs are excellent for mobile device biometric authentication systems, and this allows fingerprint recognition systems to utilize SNNs for both goals.

II. Literature

Fingerprint recognition systems have evolved from conventional Convolutional Neural Networks (CNNs) to more complex Siamese Neural Networks (SNNs). The purpose of these advances is to drive biometric systems to higher accuracy, greater efficiency, and increased robustness. In this review, an overview of key SNNs and CNNs developments is provided, and their application to fingerprint recognition in mobile applications.

- **Fingerprints:** Fingerprint patterns are probably the most relied upon and widely used biometric identifiers for security purposes due to their uniquely unalterable patterns and ease of acquisition. The fingerprints of a person are unique compared to any other identical twins, as these patterns develop in the fetal stages of human development and remain invariant across a lifetime. Due to the inherent distinctiveness and permanence, fingerprints prove to be a good metric for identity verification and security applications. [1].
- **Advantages of Fingerprint Authentication over Traditional Security Systems as a Security Measure:** Fingerprint authentication is a highly reliable biometric identification technique on the basis of uniqueness and cardinality of fingerprint, easy way of capture and permanence of fingerprint. Fingerprint authentication demonstrates better security than traditional authentication methods such as passwords or PINs because it provides these benefits:
    - ***Uniqueness & Immutability****:* The unique pattern of fingerprint ridge structures together with finger bifurcations prevents identical prints between any two human beings. The fingerprint characteristics never change over someone's lifetime without requiring any updates making them immune to alterations. The system design ensures storage immortality of enrolled fingerprints since deletion is not possible which makes them excellent for extended security operations [2].
    - **Universality & User-Friendliness**: Many different organizations including financial institutions together with law enforcement agencies and smartphone producers use fingerprint authentication worldwide because it offers universal access while maintaining user-friendly design. The authentication method operates rapidly with unobtrusive nature using standard optical sensors and scanners available to the market. The security of fingerprints represents a superior alter- native to passwords since they resist all forms of hacking attempts and sharing methods.
    - ***Resistance to Attacks****:* Fingerprints stand out as the strongest defense against biometric attacks among all forms of biometrics. Both facial recognition authentication and voice recognition can be defeated through image and video fraud and recorded voice inputs respectively. Fingerprint systems include factor detection features which stop spoofing activities.
    - ***Multi-Dimensional Security:*** Fingerprint authentication provides multi-

dimensional security because it examines fingerprint ridge patterns and minutiae points which makes the authentication mechanism much more secure than single-dimensional password systems. Brute-force attacks stand no chance against fingerprint-based systems because their high entropy levels protect them. The prediction of fingerprints turns out to be an extremely difficult task because of their elaborate nature.

- **Convenience & Compliance**: Users select simple passwords because they want convenience while simultaneously bypassing important security features including two-factor authentication. The simple biometric fingerprint scan system has minimal requirements and therefore leads to better user compliance and lower security risks.
- **Secure Local Matching & Encryption:** Fingerprint data on current devices stays within the device framework which stops the information from reaching outside systems. However, from the privacy and security point of view they facilitate else encryption of the authentication systems as is the case with Apple's Touch ID and others [3].

Fingerprint authentication is a better alternative to a traditional security system, solving the problems of security, usability, and resisting to attack by offering robust and more efficient protection to modern applications.

- **Fingerprint Matching Key Features:** Appropriately chosen features from fingerprints are extracted and form a biometric template to enable comparison and matching for authentication or identification purposes in fingerprint security systems. Refer Fig. 1.

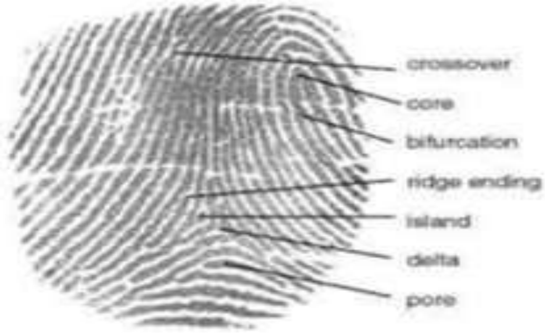

Figure 1: Fingerprint Key Features

- ***Ridge Patterns:*** Fingerprints contain ridge patterns in loops, whorls, and arches, mainly due to their usage for classification and basic differentiation.
- ***Minutiae Points:*** The minutiae points, including ridge endings and bifurcations, primarily help systems identify and match partially captured fingerprints.
- ***Ridge count and orientation:*** In this feature, the ridges are counted at minutiae points for accurate matching in fingerprints. The orientation of ridges is compared for better results. The core is the center of a fingerprint, where the ridges form loops or spirals, while the delta is formed by ridges, which split in a triangular fashion. Such reference points will guide the alignment of the fingerprint images to be compared, especially if the images have been rotated or distorted.
- ***Pores and Sweat Glands:*** Other fingerprint systems account for microscopic sweat gland pores, which strengthens security and increases accuracy in some applications.

- **Deep Learning for Biometric Recognition**: The advantage of biometric recognition with deep learning is the ability for the extraction of high dimensional features of raw data. In particular, Convolutional Neural Networks (CNNs) have been already applied to a great degree in text-based and image-based classification tasks.

  For instance, CNNs were used to recognize misleading headlines in online media where they test features to distinguish click bait headlines from true content. The authors have used CNNs to conclude that they enable the detection of clickbait on social media platforms with accuracy of 82%. It outperformed the traditional machine learning such as Random Forest by focusing model on word sequence information and its semantic relationships.

This is also in line with fingerprint recognition where CNNs are used to learn ridge patterns, minutiae points, and structural relationships in fingerprint images. CNNs and their ability to learn complex textual dependence are just as useful for a click- bait detection task as they are in the accuracy of authentication learning unique fingerprint features. Feature extraction and classification using handcrafted features does not compare favorably with CNN-based fingerprint recognition. This capability to generalize across different data modalities indicates the versatility of CNNs to be used in biometric security applications [4].

- **Convolutional Neural Networks (CNNs):** CNNs have become the center of advancements in image recognition, from fingerprint recognition to image recognition, because they are capable of recognizing spatial patterns and hierarchies of images.
    - Major contributions in this area are: AlexNet (2012): The AlexNet was introduced in the world of image classification by Krizhevsky et al, that brought a lot of improvements using deep layers, ReLU and drop out. In large scale image datasets, it proved successful, and it has established itself as an important milestone in CNN development, beyond biometric tasks such as fingerprint recognition. Despite the success of CNNs in feature extraction, they remain constrained in the pairwise comparison tasks that are key to fingerprint matching and are consequently employed for fingerprint matching using Siamese Neural Networks (SNNs) [3].
- **Siamese Neural Networks (SNNs):** The advantage of SNNs is that they are specially made for tasks where there is a need to measure the similarity between two inputs, as in biometric matching systems such as fingerprint recognition. Fingerprint Recognition with SNNs (2017): SNNs have been used on fingerprint recognition tasks where the model learns to compare two fingerprint images and learn a similarity function. A robust matching of fingerprints is provided by this method, in the case where the prints are noisy or incomplete. The selection of SNNs' architecture through which pairwise comparison is performed is particularly well suited for mobile fingerprint recognition systems due to the fact that new fingerprints can be compared against stored biometric templates in an efficient manner [5].
- **Application of SNNs in Mobile Fingerprint Matching:** As compared to conventional SNNs, implementation of SNNs to fingerprint recognition in mobile devices presents several advantages, such as low computational overhead and high accuracy.

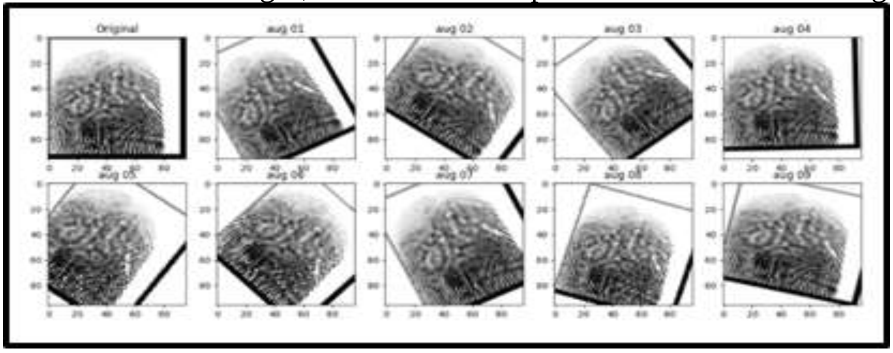

Figure 2: Fingerprint Augmentation

- ***Pairwise Matching:*** The twin architecture of SNNs allows the system to achieve high accuracy in matching tasks by comparing pairs of fingerprints, by attenuating the differences between fingerprint images.

- ***Efficiency:*** A variety of SNNs are optimized for mobile platforms, which provides efficient performance without sacrificing accuracy, an important property for mobile biometric systems
- ***Robustness:*** The advantage of SNNs is that they can handle noisy or partial fingerprint2 data, increasing system robustness in real-world mobile applications where fingerprint scans may not be the best and making it easier to design for such situations.[6–8]

III. Methodology

A structured approach to developing the Siamese Neural Network (SNN) for mobile fingerprint recognition is presented, beginning with real-time data collection, preprocessing, model training, encryption, mobile deployment and evaluation of performance.

- **Data Collection and Preprocessing:** Fingerprint data is collected in real-time when the user first registers their identity, and credentials on the mobile device in this system. When this registration phase takes place, the fingerprint image is captured [3] and safely stored by AES encryption. The foundation of the user's biometric profile is based on this fingerprint data [9].

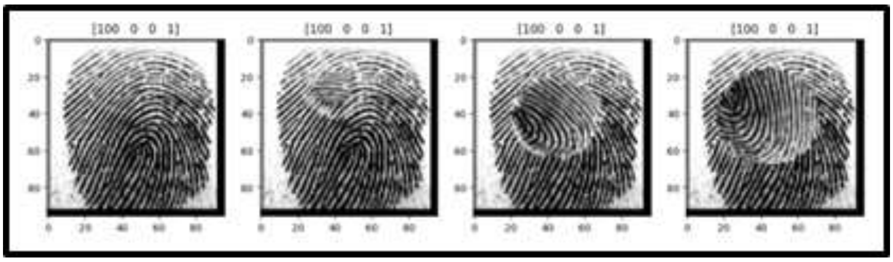

Figure 3: Fingerprint Data Collection

Pre-processing steps are essential to maintain consistency and quality in the input data before training the model.

- ***Image Resizing:*** All fingerprint images are resized into a fixed size of 96x96 pixels. The standardization that comes from this size also reduces computational resources and speeds up processing time because this size is small enough. Smaller images, as the system is more efficient on mobile devices, most of them have limited processing power.
- ***Normalization:*** Once resized, pixel values are scaled between 0 and 1. This last step makes the model converge faster in training and adds the model's ability to generalize under a variety of lighting conditions, or fingerprint quality.
- ***Data Augmentation:*** During training, the use of data augmentation techniques increases the robustness of the model. Rotation, flipping, and translation are techniques that help a SNN to recognize fingerprints in different orientations or under different conditions. It is important in dealing with the variability inherent in real-world applications [10].
    - **Model Training:** The SNN model is built with convolutional layers, max pooling layers and dense layers. The architecture is designed to extract distinctive features from fingerprint images and generate embeddings for comparison.

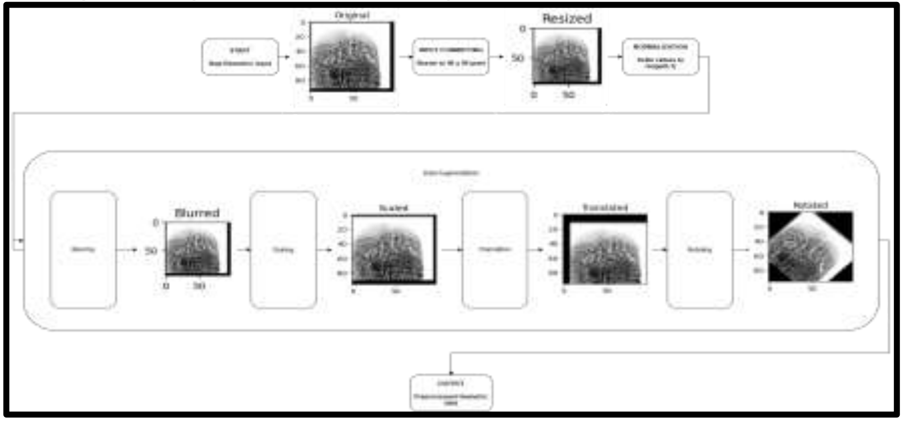

Figure 4: Data Pre-processing Flowchart

- ***Training Objective:*** A contrastive loss function is used to train the model, minimizing the distance between embeddings of matching fingerprints and maximizing the distance between nonmatching pairs. For the pairwise comparison tasks, this is an ideal approach, and SNN also accurately distinguishes genuine and impostor fingerprints during the authentication process.
- ***Hyperparameter Tuning:*** The model performance is fine-tuned on key hyperparameters such as learning rate, batch size and the number of epochs. To prevent overfitting, early stopping is performed so that once the model sees new, unseen data, it generalizes well [11].
- ***Encryption and Secure Storage:*** To guarantee sensitive biometric information security, the user's fingerprint data is stored at the time of registration using AES encryption. During a login attempt, the encrypted fingerprint data is used for future comparison. A comparison between the pre-stored fingerprint and the user input fingerprint is done, and after the comparison, the data is processed in line with the mobile device's security protocols to match data integrity and privacy [12].

IV. Design

- **Model architecture:** The architecture of the Siamese Neural Network5 used for fingerprint recognition is detailed in this flow chart which shows its multistage design.

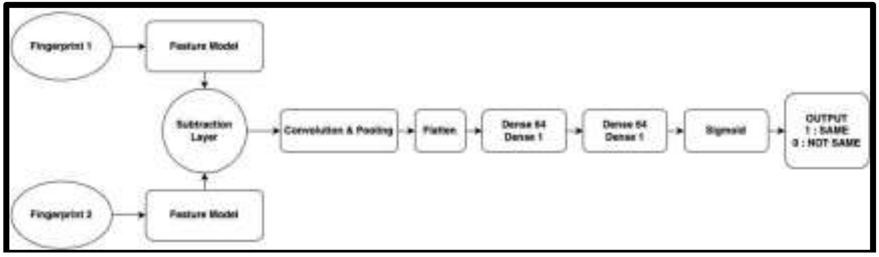

Figure 5: Model Architecture

- **Two Inputs (Fingerprint 1 and Fingerprint 2):** An architecture is presented, which starts with two different inputs, each corresponding to a fingerprint. First, the user profile (previously registered fingerprint) provides one input the second input is captured when the user attempts to log in. The setup allows us to compare biometric data.
- **Feature Model:** Each fingerprint is processed by a specialized feature extraction model that extracts salient features necessary for good comparison.
- **Subtraction Layer:** A subtraction layer turns into feature representations of both inputs, and they are compared using it. That allows the model to work with the differences in fingerprints.
- **Convolution and Pooling:** The resultant differences are processed through a series of convolutional layers followed by pooling operations. At this stage, there is a need to extract hierarchical features whilst simultaneously keeping dimensionality low in order to enable the model to find relevant patterns.
- **Flatten:** The flattened processed features are then fed to the following dense layers.
- **Dense 64, Dense 1:** The flattened vector is then passed through fully connected (dense) layers. The first dense layer with 64 neurons is intended to capture complex feature interactions, and the second dense layer contains a single neuron, which then outputs a single prediction score.
- **Sigmoid:** Finally, a final dense layer feeds to an application of a sigmoid activation function to transform raw output into a probability score indicating the probability of a match.
- **Output (1: final output binary (Same, 0: Not Same):** The final output is binary; a value of 1 means match (inputs belong to the same person), a value 0 means do not match (inputs do NOT belong to the same person) [13].

V. Results and Discussion

- **Evaluation parameters:**
- **Accuracy:** Accuracy measures the proportion of correctly classified image pairs (i.e. “similar” or “dissimilar”) out of the total number of pairs. In the context of a Siamese network, pairs are often labelled as 1 (similar) or 0 (dissimilar). A prediction is made based on whether the computed similarity (or distance) between two image embeddings crosses a chosen threshold.
- **Loss:** Loss quantifies how far the network’s predictions are from the actual labels by using a loss function called Binary Cross entropy.
- **Similarity Score:** The Similarity Score represents how alike two input images are based on the Euclidean distance between their embeddings. A high similarity score (or a low distance value) indicates that the fingerprints are the same, while a lower similarity score indicates dissimilarity.

- **Matching Time:** This parameter measures the computational efficiency—the time taken by the network to compute the similarity between a pair of fingerprints.

- **SNN (Subtract) Model Time per matching vs similarity score:** The relationship between two important variables in the matching process is illustrated in this scatter plot.

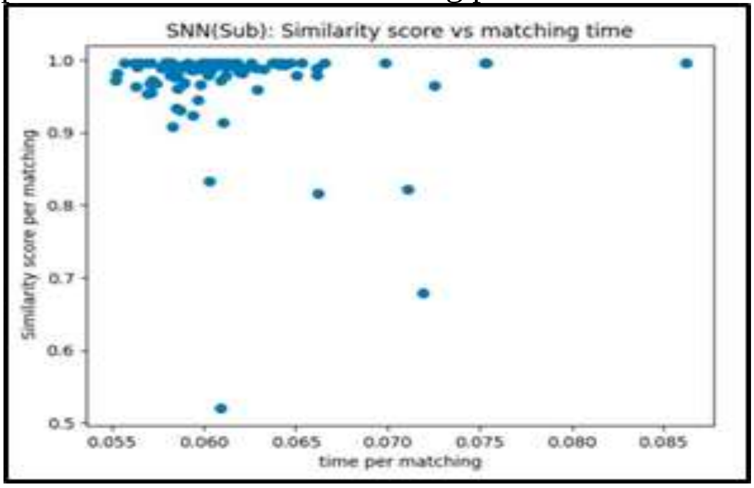

Fig. 6: Similarity Score vs Matching Time

Key Observations: X-axis (Time per Matching in seconds): indicates the time elapsed to match the pair of fingerprints.

Y-axis (Similarity Score per Matching): It is the similarity score allocated based on the Euclidean distances between two fingerprints in terms of the matching process.

Analysis: Most of the data points cluster in the top left area, around a similarity score of 1 and time per matching around 0.06 seconds. This means that a large number of matching operations are both very accurate (similarity close to 1) and very fast (small time per matching).

Lower similarity scores (between 0.5 and 0.9) with different matching times (between 0.065 and 0.085 seconds) are shown in a few outlier points. These cases correspond to situations where matching took longer or produced less similarity [14].

- **SNN (Subtract) Model Train and Test Accuracy Across 100 Epochs:** Here, the comparison of train and test accuracy for a Siamese Neural Net- work (SNN) model with subtract operation over 100 epochs is plotted in this graph.

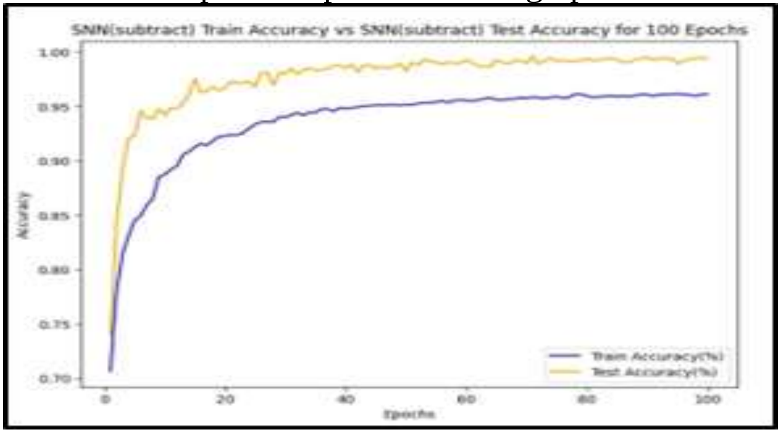

Fig. 7: SNN (Subtract) Model Train and Test Accuracy Across 100 Epochs

Key Observations: X-axis (Epochs): Number of training iterations, from 0 to 100.

Y-axis (Accuracy): Training accuracy measured over 100 epochs.

Analysis: The blue line represents the train accuracy of the SNN (subtract) model for which the accuracy begins at approximately 70% and increases up to a little above 96% at the end of training indicating effective learning.

The orange line represents test accuracy, starting at a comparable value, increasing more quickly to almost 98% after the 20th epoch and then staying at a stable level. Generalization to unseen data shows strong evidence of minimal overfitting [14].

- **SNN (Subtract) Model Train and Test Loss Across 100 Epochs**: Here, the comparison of train and test loss for a Siamese Neural Network (SNN) model with subtract operation over 100 epochs is plotted in this graph.

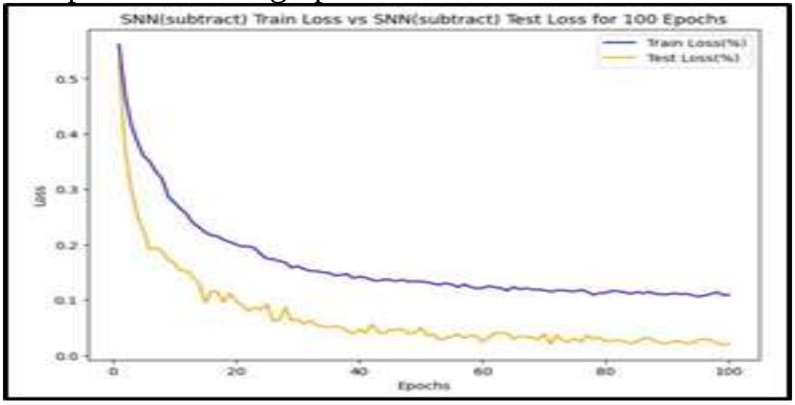

Fig. 8: SNN (Subtract) Model Train and Test Loss Across 100 Epochs

Key Observations: X-axis (Epochs): Number of training iterations, from 0 to 100.

Y-axis (Loss): Training loss measured over 100 epochs.

Analysis: The blue line represents the training loss for the SNN (subtract) model. It begins at a high value (around 2.0) and decreases sharply during the initial epochs, eventually dropping to a very low value (close to 0.02) by the end of training. This indicates that the network is effectively minimizing its error on the training data.

The orange line corresponds to the test (or validation) loss. It starts at a similarly high level as the training loss, then declines rapidly during the early epochs, mirroring the training loss—but stabilizes sooner and remains consistently low throughout the later epochs. This trend demonstrates strong generalization to unseen data and minimal overfitting. [14]

- **Using SNN(L1): Similarity Score vs Matching Time:** Here similarity scores are compared against matching time of a fingerprint recognition system based on the Siamese Neural Network (SNN) with L1 distance.

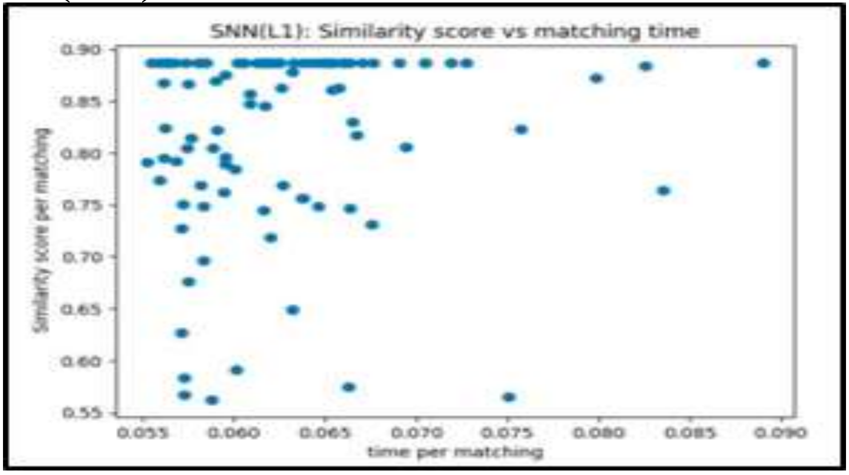

Fig. 9: SNN(L1) Model: Similarity Score vs Matching Time

Key Observations: X-axis (Time per Matching in seconds): Indicates the time elapsed to match the pair of fingerprints.

Y-axis (Similarity Score per Matching): It is the similarity score allocated based on the L1 distances between two fingerprints in terms of the matching process.

Analysis: Most matches fall into the top left (higher similarity scores, faster matching times), with some points close to a maximum similarity score (0.90) at low times (0.055 to 0.065 seconds). At slightly higher matching times (above 0.065 seconds) there are some outliers with lower similarity scores (around 0.55-0.75).

A clear trade-off between similarity scores at different time intervals, with the highest performance in both accuracy and time near 0.90 score mark [4]

- **Training Accuracy Comparison of SNN Models:** This line graph compares the training accuracy of two SNN (Siamese Neural Network) models over 15 epochs.

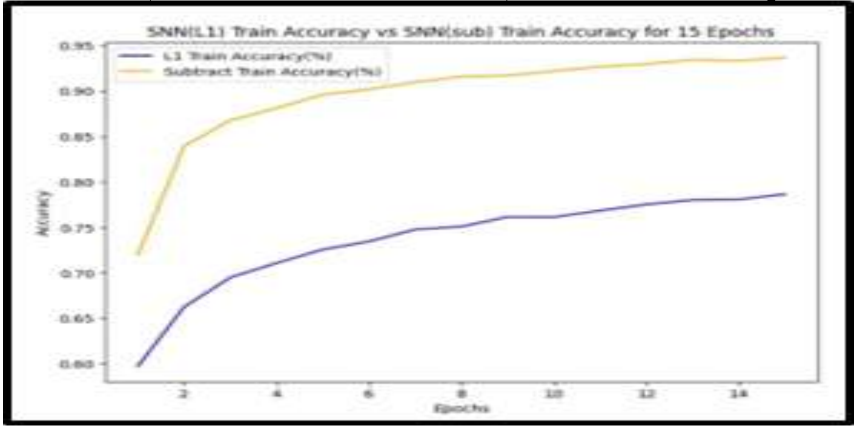

Fig. 10: Training Accuracy Comparison of SNN Models

Key Observations: Y-axis (Accuracy): Represents the accuracy of each model during training.

X-axis (Epochs): Number of epochs during training, ranging from 1 to 15.

Analysis:

L1 SNN (Blue Line): Shows a gradual increase in accuracy, starting at around 60% and steadily climbing to approximately 78% over 15 epochs. This model improves consistently but at a slower rate.

Subtract SNN (Yellow Line): Begins with a higher initial accuracy, around 83%, and reaches a peak of about 92% accuracy in the first few epochs. The improvement slows after 5 epochs but maintains a higher accuracy compared to the L1 SNN [4].

- **Test Accuracy Comparison of SNN Models:** This line graph com- pares the test accuracy of two SNN (Siamese Neural Network) models over 15 Epochs.

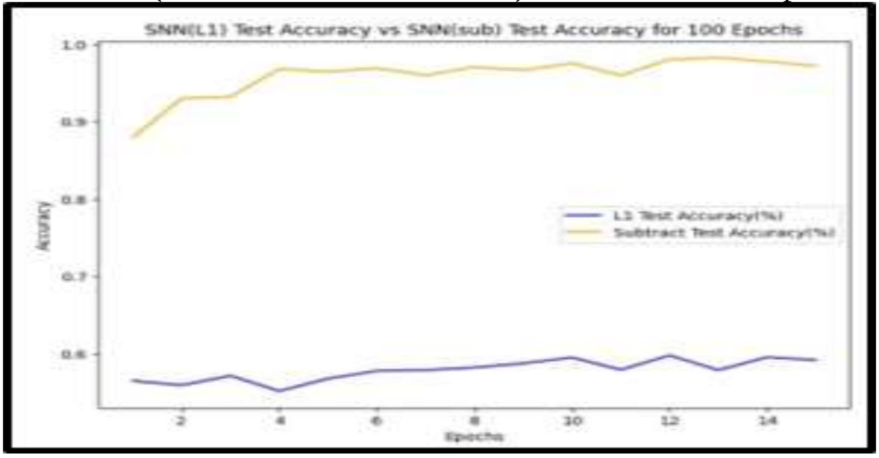

Fig. 11: Test Accuracy Comparison of SNN Models

Key Observations: Y-axis (Accuracy): Represents the test accuracy of each model.

X-axis (Epochs): Number of epochs during the testing phase, ranging from 1 to 15.

Analysis: L1 SNN (Blue Line): The test accuracy of the L1 model remains fairly consistent, fluctuating between 55% and 60% throughout the epoch. There is no significant improvement over time.

Subtract SNN (Yellow Line): The Subtract SNN model starts with a high accuracy of approximately 90% and shows slight improvements, peaking close to 98% before stabilizing [4].

- **Comparison of Similarity Score of Models:** In the comparison, com- parison of the similarity scores of two SNN (Siamese Neural Network) models with different matching methods for the same matching time.

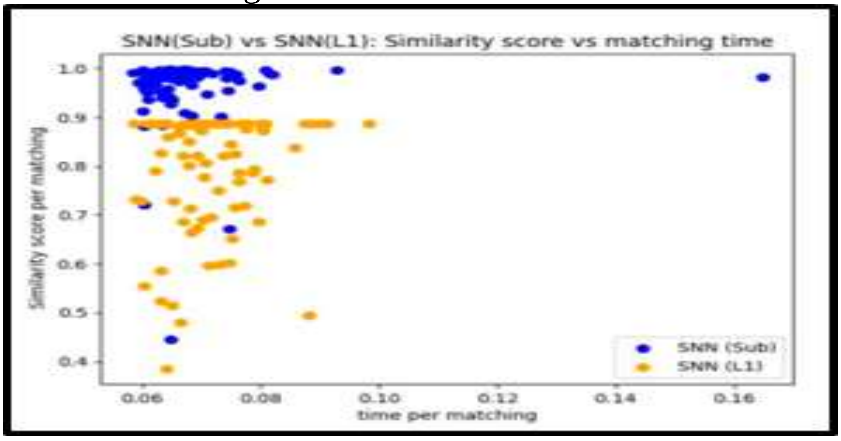

Fig. 12: Comparison of Similarity Score of Models

Key Observations: X-axis (Time per Matching in seconds): indicates the time elapsed to match the pair of fingerprints.

Y-axis (Similarity Score per Matching): It is the similarity score allocated based on each model's equation for calculating the distances between two fingerprints features in terms of the matching process.

Analysis: SNN (Sub) (Blue Dots): This method shows high similarity scores consistently and most of the clusters are near 1.0, indicating strong performance. And the time per matching stays low and is concentrated around 0.06 seconds.

SNN (L1) (Orange Dots): The similarity scores of this method are more spread out, with values ranging from 0.4 to 0.9 which exhibit more variance in performance. The matching times are more widespread, beyond 0.1 seconds.

- **Insights from fingerprint recognition of image:** The fingerprint recognition image is made up of three panels that present the fingerprint recognition process in the context of biometric security. The different labels and scores show how the system evaluates the similarity between fingerprints:
    - **Fingerprint 1:**

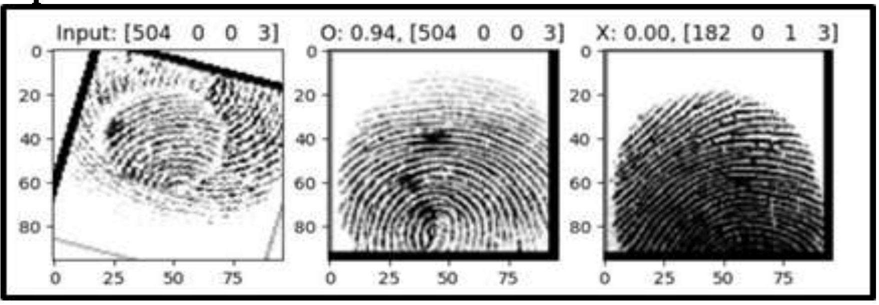

Fig. 13: Fingerprint Recognition Output1

- **Left Panel (Input Image): Label: Input: [504 0 0 3] -** It is an input fingerprint image on this panel. The fingerprint looks a bit rotated, an initial raw (or unprocessed image) that needs correction or normalization. The print is clear, ridge lines are visible, but there are some borders or artifacts around the edges of the image.

- **Middle Panel (Matched Fingerprint): Label: O: 0.94, [504 0 0 3] -** This panel has a high match score (0.94) making this is a successful matching attempt or recognition. The fingerprint pattern looks corrected or aligned with a well-centred clear fingerprint pattern. This score of 0.94 means that input fingerprint and a stored template are very similar (i.e., a success match) [1].
- **Right Panel (Unmatched Fingerprint): Label: X: 0.00, [182 0 1 3] -** As this score is 0.00, this is a failed matching attempt. Here the fingerprint seems darker and more distorted, and a few ridge lines seem less defined than in the middle panel. This fingerprint does not match because of the lower quality or distortion of it [2].

o **Fingerprint 2:**

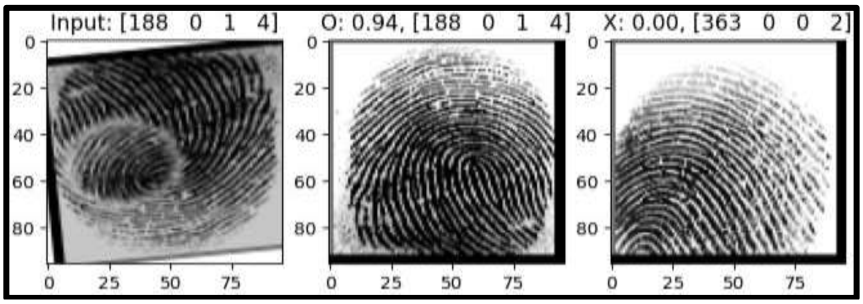

Fig. 14: Fingerprint Recognition Output2

- **Left Panel (Input Image): Label: Input: [188 0 1 4] -** Ridges in the fingerprint display a mild rotation which usually occurs in unprocessed images needing alignment or normalization. Smooth ridge lines run through the image while artifacts produce noticeable damage at its outer edges.
- **Middle Panel (Matched Fingerprint): Label: O: 0.94, [188 0 1 4] -** The fingerprint success rate is validated through a high match score of 0.94. Well-defined ridge-line patterns merge with the improved alignment of the fingerprint along with a more central position.
- **Right Panel (Unmatched Fingerprint): Label: X: 0.00, [363 0 0 2] -** The analysed fingerprint results in an evaluation score of 0.00 which represents unsuccessful identification. A blurry fingerprint pattern exists in this image while its ridge structure remains hazy in contrast to the reference fingerprint.

o **Fingerprint 3:**

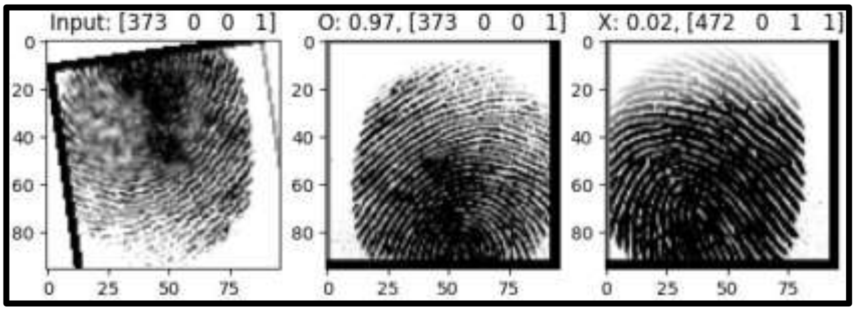

Fig. 15: Fingerprint Recognition Output3

- **Left Panel (Input Image): Label: Input: [373 0 0 1] -** The fingerprint shows a slight angle which suggests possible correction should be performed alongside normalization. The ridges are accessible for viewing yet particular segments display both noise along with distortion.
- **Middle Panel (Matched Fingerprint): Label: O: 0.97, [373 0 0 1] -** The recognition attempt proves successful based on the high match score of 0.99. The fingerprint matches

the template perfectly since it maintains proper alignment while clearly displaying its ridge patterns.

- **Right Panel (Unmatched Fingerprint): Label: X: 0.02, [472 0 1 1] -** The matching procedure results in failure when the score stands at 0.00. A blurry and distorted fingerprint on the left panel indicates that the biological print is substantially different from what is stored on the database.

- Fingerprint 4:

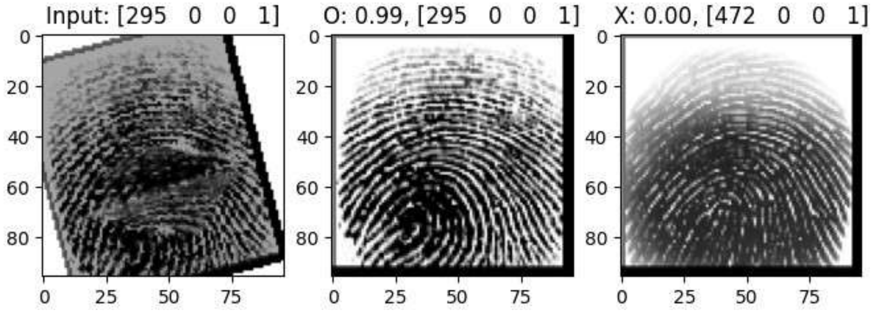

Fig. 16: Fingerprint Recognition Output4

- **Left Panel (Input Image): Label: Input: [295 0 0 1] -** Noise appears among areas in the rotated fingerprint print. The outlined shapes become harder to identify because the areas remain faint when scanned over darker components.
- **Middle Panel (Matched Fingerprint): Label: O: 0.99, [295 0 0 1] -** With a match score reaching 0.97 the recognition meets extreme success standards. The fingerprint stands in good center position and matches perfectly with the recorded print.
- **Right Panel (Unmatched Fingerprint): Label: X: 0.00, [472 0 0 1] -** The provided fingerprint exhibits no match with the reference profile according to the score of 0.02. Image distortion and lack of ridge detail in structure leads to this mismatch observation.

VI. Conclusion

- Conclusion derived from the figures: The figures are used for an in- depth comparison of different fingerprint matching models including Siamese Neural Networks (SNNs), CNN based models and traditional minutiae-based methods.
  - ***Performance and Accuracy:***
    - The Siamese Neural Networks (SNNs) are nearly perfect with accuracy rates of nearly 100% and also display high similarity scores. For fingerprint recognition tasks, the SNN (Subtract) model outperforms the SNN (L1) model, and it is the most effective model.
    - As with traditional methods, CNN based models have accuracy rates of 85–90% but are dramatically slower and less efficient than SNNs. The matching rates of traditional minutiae-based models are lowest (around 60%-70%) and thus the least suitable for highly demanding fingerprint matching tasks [3].
  - ***Speed and Efficiency:***
    - The fastest matching times (below 10 milliseconds) make SNNs particularly well suited to real-time fingerprint recognition, for example, unlocking mobile devices or secure access points.
    - SNNs are moderately slow but much more efficient than CNNs. The slowest to perform are traditional minutiae-based methods, which take about 40 milliseconds to match, which makes them impractical for use in real-time applications like mobile security. [5]
  - ***Computational and Memory Efficiency:***
    - SNNs come up with their good trade-off that uses less computational power than any CNN model and yet produces good performance. They also have moderate

storage requirements which are suitable for devices with little resources, such as mobile devices.

- CNN-based models are computationally expensive in terms of memory and processor requirement and may not be convenient for mobile devices constrained with hardware limitations. On the other hand, models based on traditional minutiae can make use of the least resources, However, their accuracy and speed are not high enough to be desirable in secure fingerprint recognition.

- ***Training and Generalization:***
  - The SNN models learn exceptionally well and there is no overfitting on both training and unseen data. For instance, the SNN (Subtract) model obtains 98% test accuracy, which is very accurate for mobile applications.
  - SNN models generalize as well but are outperformed by the CNN models that are more efficient. Training and generalization are not evaluated by traditional methods, yet they are considered less adaptable to more complex tasks like fingerprint matching [8].

- **Implications for Mobile Fingerprint Security:**
  - Compared to other security systems, such as those based on keystroke dynamics, this analysis is particularly relevant for mobile fingerprint security systems that need to achieve high accuracy, fast response times, and efficient use of computational resources.
  - Mobile security applications require quick and accurate fingerprint matching, and SNNs are the best choice for mobile security. They can achieve near perfect accuracy within milliseconds so that it doesn’t impact the user experience and at the same time keeps the security strong [7, 9].
  - Mobile use for CNN based models may be plausible but they would face a ceiling on energy consumption possibly because of greater computational requirements or memory, causing slower unlock times or battery drain [10].
  - The current mobile security approach requires models to function faster along with greater accuracy whereas traditional minutiae-based models offer slower speeds combined with lower accuracy which results in reduced user satisfaction [11].

Therefore, the conclusion is that Siamese Neural Networks (SNNs) are the most appropriate models for mobile fingerprint security due to the excellent balance between speed, accuracy and resource efficiency, compared to other proposed models. Thanks to their quick response time and high accuracy, mobile devices can be unlocked quickly and securely in a modern mobile environment [4, 12–14].

References

[1]Chen, Y., Wang, H.: Efficient fingerprinting-based android device identification with zero-permission identifiers. ResearchGate (2016)

[2]Gune, S., Khedkar, S.: The recognition of fingerprints on mobile applications - an android case study. ResearchGate (2016)

[3]Bakhshi, B., Veisi, H.: End to end fingerprint verification based on convolutional neural network. In: 2019 27th Iranian Conference on Electrical Engineering (ICEE), pp. 1994–1998 (2019). https://doi.org/10.1109/IranianCEE.2019.8786720

[4]Shaikh, M.A., Annappanavar, S.: A comparative approach for clickbait detection using deep learning. In: 2020 IEEE Bombay Section Signature Conference (IBSSC), pp. 21–24 (2020). https://doi.org/10.1109/IBSSC51096.2020.9332172

[5]Barzut, S., Milosavljevi´c, M.: The application of convolutional neural networks for fingerprint recognition: A comparative analysis. In: Sinteza 2022 - International Scientific Conference on Information Technology and Data Related Research, pp. 224–229 (2022). https://doi.org/10.15308/Sinteza-2022-224-229

[6]Kamil, B.Z., Abdullah, D.A., Hatem, H.R.: Biometrics based on deep learning: A survey. Int. J. Nonlinear Anal. Appl. (2024)

[7]Berdich, A., Groza, B., Mayrhofer, R.: A survey on fingerprinting technologies for smartphones based on embedded transducers. IEEE Internet of Things Journal 10(16), 14646–14670 (2023) https://doi.org/10.1109/JIOT.2023.3277883

[8]Nithya, B., Sripriya, P.: Fingerprint identification by training a lstm network with fingerprint segments as sequence inputs. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES) (2021)

[9]Afah, D., et al.: Smartphones verification and identification by the use of finger- print. In: Advanced Techniques for IoT Applications: Proceedings of EAIT 2020. Springer, ??? (2022)

[10] Saponara, S., Elhanashi, A., Zheng, Q.: Recreating fingerprint images by convolutional neural network autoencoder architecture. IEEE Access 9, 147888–147899 (2021) https://doi.org/10.1109/ACCESS.2021.3124746

[11] Lone, S.A., Mir, A.H.: Smartphone-based biometric authentication scheme for access control management in client-server environment. International Journal of Information Technology and Computer Science (IJITCS) 14(4), 34–47 (2022)

[12] Hammad, M., Wani, M.A., Shakil, K.A., Shaiba, H., El-Latif, A.A.A.: Deep cancelable multibiometric finger vein and fingerprint authentication with non- negative matrix factorization. IEEE Access 12, 120638–120660 (2024) https://doi.org/10.1109/ACCESS.2024.3450372

[13] Guan, X., Pan, Z., Feng, J., Zhou, J.: Joint identity verification and pose alignment for partial fingerprints. IEEE Transactions on Information Forensics and Security 20, 249–263 (2025) https://doi.org/10.1109/TIFS.2024.3516566

[14] Chen, S., Guo, Z., Li, X., Yang, D.: Query2set: Single-to-multiple partial fingerprint recognition based on attention mechanism. IEEE Transactions on Information Forensics and Security 17, 1243–1253 (2022) https://doi.org/10. 1109/TIFS.2022.3159151