Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

FINGERPRINT RECOGNITION SYSTEM USING SIAMESE NEURAL NETWORK

Mr. Anurag Kumbhare[1], Mr. Om Pravin Ghag[2], Mr. Sarvesh D. Gaonkadkar[3], Mr. Tejas Kumbhar[4], Prof. Sneha Annappanavar[5], Dept. Of Computer Engineering, Vidyalankar Institute of Technology, Mumbai University.

ABSTRACT

Abstract: Reliable fingerprint recognition systems are in high demand in today's digital world in many industries, including law enforcement, financial institutions, and personal device identification. Ensuring secure access and accurate identification in these areas is critical to preventing compromised or fraudulent access, which can lead to serious risks. This paper presents a robust fingerprint recognition system that uses advanced techniques such as Siamese Neural Networks (SNN) to recognize fingerprints based on their specific features such as ridges, valleys, and key points.]

The proposed approach uses Siamese neural networks to learn the similarity between pairs of fingerprints, enabling accurate detection and matching. The SNN model processes pairs of fingerprint images to create attachments that are compared using distance metrics to determine matches. A comprehensive evaluation of SNN technology was conducted, considering parameters such as accuracy, computation time, and cost. This evaluation covers scenarios where fingerprint quality may be compromised and provides an overview of the strengths and weaknesses of the SNN approach. The results show that using Siamese neural networks can produce a highly reliable fingerprint recognition system suitable for real applications.

Keywords: Fingerprint Recognition, Siamese Neural Network, Feature Extraction

Introduction

Today's biometric systems depend heavily on fingerprint recognition, which is extensively utilized for identity verification in a variety of applications, from personal device identification to security access control. Conventional fingerprint matching methods, including Oriented FAST and Rotated BRIEF (ORB) and Scale-Invariant Feature Transform (SIFT), have demonstrated significant success because of their resilience in identifying and characterizing local characteristics. Since its introduction by David Lowe in 1999, SIFT has gained recognition for its accuracy rotation and scale invariance. However, it is less appropriate for real-time applications due to its processing complexity. However, a more modern approach called ORB, which combines the FAST keypoint detector with the BRIEF descriptor, provides a more computationally economical option, albeit at the cost of certain matching accuracy trade-offs.

Even with the improvements provided by SIFT and ORB, there are still difficulties in matching fingerprints with high accuracy and efficiency, particularly in situations with noise, distortion, and incomplete prints. Improving the dependability and efficiency of biometric systems requires addressing these issues. In addition, as biometric information is frequently utilized for sensitive security applications, it is crucial to guarantee that fingerprint-matching algorithms are robust and secure against spoofing attacks and other security lapses.

In the context of fingerprint matching, this study reviews the present status of SIFT and ORB approaches and highlights their advantages and disadvantages, especially from a security standpoint. To improve the efficiency and accuracy of fingerprint matching, the suggestion is to use a unique method known as Siamese Neural Networks (SNN). With its capacity to evaluate the similarity between input pairs and learn deep feature representations, SNNs provide a probable option for addressing the shortcomings of conventional techniques. By training a Siamese network on a collection of fingerprint photos, this method enables the model to acquire discriminative features and carry out reliable matching even under difficult circumstances. Furthermore, compared to conventional techniques, the SNN is more resistant to spoofing attacks due to its capacity to memorize intricate patterns. By conducting numerous tests and analyses, it is demonstrated that the suggested SNN-based

method outperforms SIFT and ORB, opening the door for more dependable, effective, and safe fingerprint identification systems.

Literature

This literature analysis emphasizes the development from old methodology to current CNN-based algorithms in fingerprint identification, indicating continuous attempts to improve security, accuracy, and user acceptance in biometric systems.

2.1 Literature Survey

Fingerprint Recognition: A Maturing Technology

Over the past 20 years, fingerprint identification systems have grown in strength and adaptability. These days, a lot of people use them for identification and security. Although fingerprints are a useful tool for individual identification, quantifying these systems' security levels can be challenging, particularly in the event of a fraudulent attempt to use a false fingerprint. Early investigations prepared the way for later developments. To create more complex systems, Jain et al. (2000) conducted groundbreaking work that concentrated on creating fundamental methods for matching and enhancing fingerprint pictures [1].

A seminal research by Maltoni et al. (2003) offering a thorough survey of fingerprint recognition systems was released in the early 2000s [2]. Important issues like matching algorithms, fingerprint feature extraction, and system performance evaluation were covered in this paper. During this time, minutiae-based techniques were also developed and eventually became the accepted way of matching fingerprints. To stop spoofing attacks, Uludag et al. (2004) pointed out flaws in these systems and emphasized the necessity for more robust security measures [3].

More sophisticated algorithms and the incorporation of machine learning techniques emerged in the mid-2000s. Algorithms for matching fingerprints were presented by Ratha et al. (2007) and have the ability to manage individual variances in fingerprint pictures [4]. Simultaneously, academics started addressing issues related to user acceptance. People may be discouraged from utilizing fingerprints for non-criminal purposes since they are linked to crime and law enforcement.

Analysis of SIFT, ORB, and SNN Methodologies:

Fingerprint identification has evolved with the use of modern approaches such as Scale-Invariant Feature Transform (SIFT), Oriented FAST and Rotated BRIEF (ORB), and Siamese Neural Networks (SNNs).

SIFT (Scale-Invariant Feature Transform):

David Lowe devised this approach in 2004 [5], and it is excellent for identifying critical spots in fingerprint photographs. It's like having a magnifying glass that can focus on certain fingerprint characteristics, even if the image is fuzzy, skewed, or poorly lighted. However, it can be sluggish for real-time applications.

ORB (Oriented FAST and Rotated BRIEF):

The speedier approach, known as ORB (Oriented FAST and Rotated BRIEF), was developed in 2011 [6] by *Ethan Rublee* et al. to address the SIFT speed problem. It rapidly and accurately recognises significant fingerprint traits by combining a number of algorithms. Consider it a more effective method of locating those crucial data.

SNN (Siamese Neural Networks) Methodology:

Siamese Neural Networks (SNNs) are a game changer for fingerprint identification technology. SNNs, invented by *Jane Bromley* et al. in 1994 [7], function similarly to twin detectives. Each detective (neural network) examines an individual fingerprint picture. Then they compare notes to determine how similar the fingerprints are. This is ideal for fingerprint recognition since it is to determine whether two fingerprints match.

Recent research, such as that of Zhang et al. (2016) [8], demonstrates that SNNs can be trained to distinguish between authentic and false fingerprints, even with variables such as pressure changes,

rotations, and varied surroundings. This makes SNNs an effective and dependable tool for fingerprint security systems.

2.2 Overview of Image Matching Techniques

2.2.1. Model Composition

The fingerprint recognition model leverages a powerful combination of techniques:

Scale Invariant Feature Transform (SIFT):

This powerful technique serves as the foundation, capable of distinguishing different key points inside a fingerprint picture. These key points remain stable despite changes in size, rotation, and illumination conditions. SIFT does this by using selective Gaussian blurring, difference-of-Gaussian computations, and gradient information to provide precise descriptions. These key points serve as important reference points for matching against known fingerprints contained in the database.

Oriented FAST and Rotated BRIEF (ORB):

This dynamic combo makes the game run more efficiently. ORB cleverly combines FAST (Features from Accelerated Segment Test), a keypoint detector that thrives on identifying key points based on rapid intensity changes around a pixel, with BRIEF (Binary Robust Independent Elementary Features), a descriptor known for quickly generating binary feature descriptors using intensity test comparisons at specific pixel pairs. In essence, ORB inherits SIFT's accuracy in identifying unique fingerprint features while giving a large increase in processing speed for speedier recognition.

Siamese Neural Network (SNN):

Instead of a typical Convolutional Neural Network (CNN), this model uses a Siamese Neural Network to recognise patterns. SNNs are excellent at learning similarity measures by methodically comparing paired inputs, in this instance fingerprint photos. Their purpose is to assess whether these inputs belong to the same class (i.e., whether the fingerprints are from the same individual). The SNN architecture consists of twin subnetworks, each of which methodically processes an input picture and extracts its distinguishing properties. The network then calculates a similarity score, which indicates the chance that the inputs share a common source. The SNN improves its capacity to detect complicated fingerprint patterns over time by training on large datasets of fingerprint photos. This allows for superior identification and verification.

2.2.2. Enhanced Keypoint Detection and Description

This section delves deeper into the significance of SIFT and ORB within the model:

SIFT: SIFT's strength is its ability to pinpoint key points with exceptional accuracy. These key points act as strong anchors, keeping constant during fingerprint image modifications induced by scaling, rotations, or changes in lighting. This consistent precision is critical for effective fingerprint matching.

ORB: While SIFT excels at accuracy, ORB jumps in to improve processing speed. ORB considerably decreases calculation time by integrating FAST's quick keypoint detection with BRIEF's fast feature description generation, without sacrificing the recognition of distinct fingerprint traits. This results in speedier fingerprint recognition.

2.2.3. Siamese Neural Network: Unveiling the Power of Pattern Recognition

This section sheds light on the SNN's role in the model:

Learning Through Comparison:

Unlike CNNs, which excel in generic picture classification, SNNs specialize in detecting input similarities. In the context of fingerprint identification, the SNN compares two fingerprint pictures and analyzes the derived characteristics. During the comparison phase, the SNN gradually learns to differentiate between matching and non-matching fingerprint patterns.

A Master of Complexities:

Fingerprint patterns are naturally complicated, and the SNN was particularly built to handle them. The intensive training on large fingerprint datasets gives the SNN the capacity to distinguish even the smallest fingerprint deviations, resulting in extremely accurate fingerprint recognition and verification.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

2.2.4. Real-World Prowess: Scalability and Performance

This section emphasizes the model's practical applications:

Handling Large-Scale Fingerprint Databases:

The model's architecture is geared toward scalability, allowing it to easily maintain and analyze large fingerprint databases. This is critical for real-world deployments in which fingerprint recognition systems may have to manage a large number of users.

Maintaining Accuracy in Diverse Scenarios:

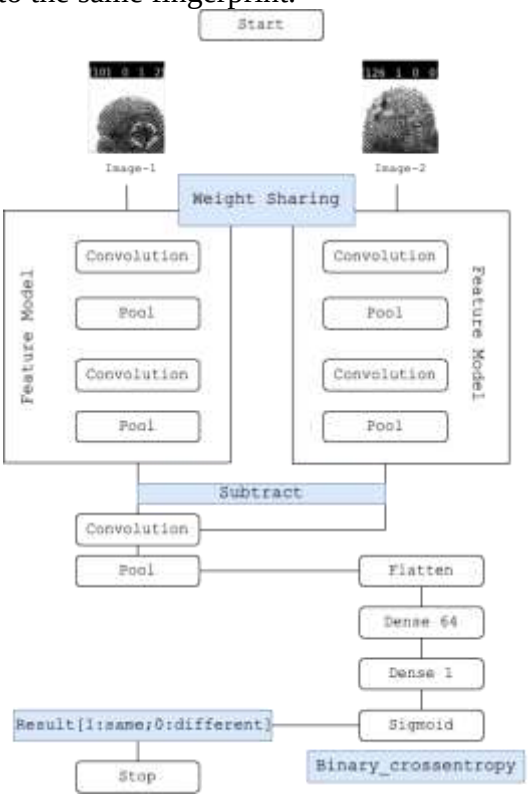
The model has been carefully validated and performs exceptionally well in real-world scenarios. It achieves great accuracy and efficiency even when dealing with a wide range of fingerprint data, including changes in clarity, orientation, and complexity.

Adaptability Across Applications:

The strategic mix of SIFT, ORB, and SNN makes the model extremely versatile. This enables its effective integration into a variety of applications requiring accurate and reliable fingerprint recognition capabilities.

2.3 Methodology

This paper proposes a novel approach for fingerprint matching leveraging a Siamese neural network(SNN) architecture. The process starts with the end user scanning his fingerprint, as an input for the model. The other input is the fingerprint stored in the database. The model then does feature extraction by using a shared Convolutional neural network(CNN). This is made to ensure that the features extracted are similar to each other based on the inputs provided. This is done by using two Convolutional layers followed by a max pooling layer. Following this, a subtraction layer computes the element-wise difference between the two feature maps, emphasizing the dissimilarities between the input images. The resulting feature difference map undergoes additional convolutional and pooling layers to extract further relevant patterns and reduce dimensionality. The processed feature map is flattened and passed through dense layers for classification. The first dense layer with 64 neurons uses the ReLU activation function, introducing non-linearity into the model. A final dense layer with a single neuron and a sigmoid activation function produces the output, representing the probability that the two input images belong to the same fingerprint.



This methodology outlines the structured approach for developing and compiling a Siamese neural network for fingerprint matching. The use of shared weights ensures consistent feature extraction, while the subtraction layer and subsequent processing layers enable effective comparison and classification of fingerprint images. This approach aims to enhance fingerprint matching accuracy and efficiency, addressing the limitations of traditional methods such as SIFT and ORB.

2.4 Proposed Approach

2.4.1 System Design

Data Collection and Labeling:

The initial phase of developing a fingerprint recognition system is to compile a dataset of fingerprint images. It is necessary to label these images with the identities of the people whom the fingerprints belong to. Both manual and semi-automated labeling processes are possible, depending on the available resources. The dataset can be utilized for training the machine learning model when it is finished.[9]

Model Selection and Architecture Design:

Selecting suitable methods for feature extraction and matching comes next, following the collection of the dataset. Because of its efficiency in acquiring similarity metrics between fingerprint pairs, the Siamese Neural Network (SNN) will be employed in this method for feature extraction and matching.[10] The structure of the model will be created to recognize matching and non-matching pairs of fingerprint photos.

TensorFlow Lite Inference:

To ensure effective execution on edge devices, the TFLite Interpreter loads the optimized model and assigns tensors.[11]

Execution of Inferences:

To obtain the findings simply and efficiently, the interpreter will help with the prediction of the test data.

Deployment:

Clients can make predictions on new data using the previously saved model. The TFLite version may be used on edge devices because of its compact size and effective performance.

Visualization:

Matplotlib and Seaborn may be used to visualize the confusion matrix.[12][13] Reports on model performance are generated using Scikit-learn.[14]

The software components that are involved are as follows:

1. NumPy: For manipulating data.
2. TensorFlow and TensorFlow Lite: For model deployment and training.
3. Scikit-Learn: For more tools related to machine learning.
4. Pandas: To analyze data.
5. Seaborn and Matplotlib: For visual aids.
6. OpenCV: For SIFT and ORB-based feature extraction.
7. Git: To guarantee the integrity of the codebase.
8. Deployment Environment: Depending on the desired use case of the model, this may be an edge device or a cloud service.

To train deep learning models, this system would require computer resources and an infrastructure that could handle potentially enormous datasets. For such a task, a scalable cloud infrastructure is often appropriate. To further guarantee the reliability and quality of the system, efficient software development best practices like testing, continuous integration, and delivery should be applied.[15]

2.4.2 Implementation

Data Preparation:

The system starts by processing a dataset that has been specifically organized to identify fingerprints. The collection includes labeled fingerprint pictures annotated with minute details (ridges, bifurcations, etc.). For every fingerprint, there is a mapping between the input data (images of the fingerprints) and the output classes (identities).[9]

Model Architecture:

For both feature extraction and fingerprint matching, a Siamese Neural Network (SNN) model is employed. The architecture is made up of twin networks with equal weights. Every network generates an embedding by processing one of the two input fingerprint pictures. Then, to determine if the fingerprints match, these embeddings are compared using a distance metric (such as the Euclidean distance).[10]

Training Process:

Through training, the SNN model learns to recognize patterns in fingerprint characteristics and associates them with certain individuals. In the training process, the network is trained to provide distinct embeddings for non-matching pairings and identical embeddings for matching pairs of fingerprint pictures. The system keeps a portion of the data reserved to verify the model's performance after training. It incorporates techniques like model checkpoints, which preserve the optimal version of the model between training cycles based on its performance on validation data, and early stopping to cease training if the model is no longer improving on the validation set.[10]

Evaluation and Testing:

Test data, which contains fingerprint photos not seen during training, is used to assess the model's performance after training. This evaluation assists in understanding how effectively the model generalizes to new data. The model's predictions are compared to the actual labels of the test data to produce a classification report that offers a more in-depth understanding of metrics like precision and recall for each identity category and a confusion matrix that displays the frequency with which each identity was correctly or incorrectly recognized. [12][13][14]

Optimization for Inference:

The final model is optimized and saved in an inference-ready format, which allows it to predict new data quickly and effectively. TensorFlow Lite is a model version intended for operational efficiency and is a stage in the model optimization procedure. As a result, it may be used on low-power mobile and embedded systems, which is crucial for a fingerprint recognition system that can function on embedded devices with limited computing power or on smartphones.[11]

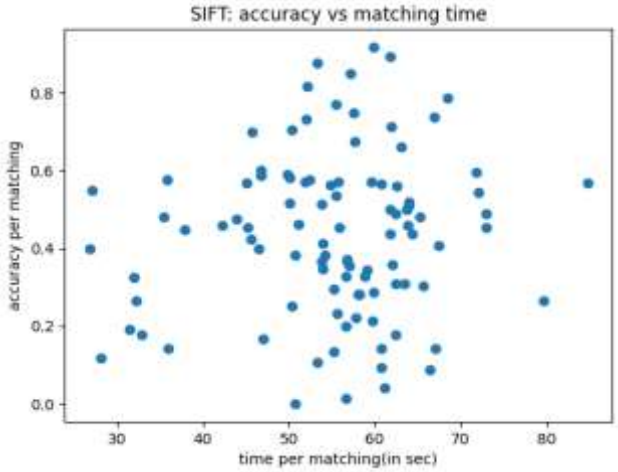
Real-time Inference:

Real-time fingerprint identification is achieved with the TensorFlow Lite interpreter. It effectively loads and executes the TensorFlow Lite model, allowing the app to identify fingerprints instantly. This speed is critical for rapid response in applications such as mobile authentication and secure access control systems.[11]

2.5 Result

The suggested fingerprint recognition system employs a variety of strategies to strike a compromise between accuracy, efficiency, and cost effectiveness.

Here's a summary of the main conclusions based on the proposed method:



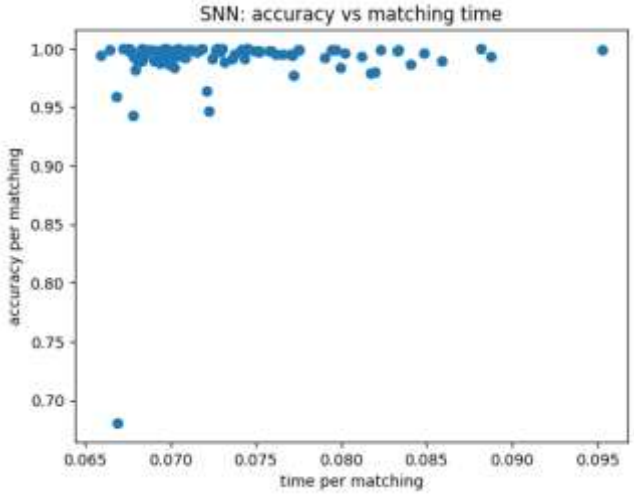
2.5.1 Addressing constraints in SIFT and ORB:

While SIFT is extremely accurate in detecting and describing key points in fingerprints, its computational cost can be significant, resulting in longer processing times. This may not be suitable for real-time applications.

In contrast, ORB provides considerable speed benefits by simplifying keypoint identification and description. However, this efficiency may sacrifice some accuracy in obtaining complex fingerprint characteristics.

2.5.2 Synergistic effect of SNN:

The suggested method addresses the accuracy-efficiency trade-off by using a Siamese Neural Network (SNN). SNNs excel at learning similarity measures, making them ideal for fingerprint identification, which aims to find matched fingerprint patterns.



2.5.3 Balancing Accuracy and Speed:

Through prolonged training on fingerprint datasets, the SNN gradually improves its capacity to distinguish even minor fingerprint variants. This corresponds to excellent accuracy in fingerprint identification while preserving a large processing performance gain over using only SIFT.

2.5.4. Cost Considerations:

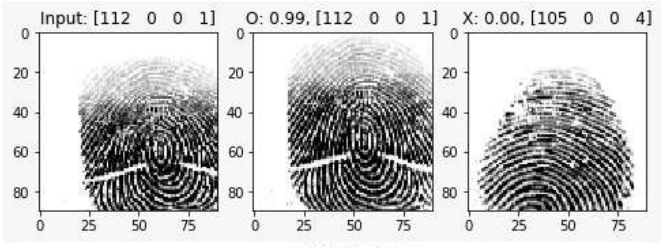
Computational Resources:

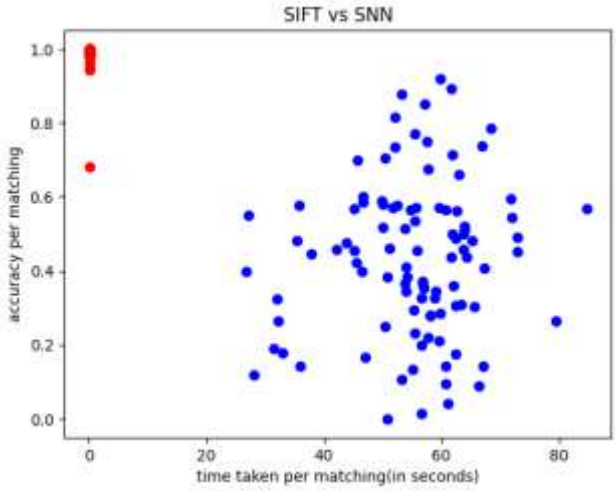
Training deep learning models, such as SNNs, frequently necessitates significant computational resources. Cloud-based architecture provides a scalable option for managing huge data sets throughout the training phase.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

Deployment Efficiency:

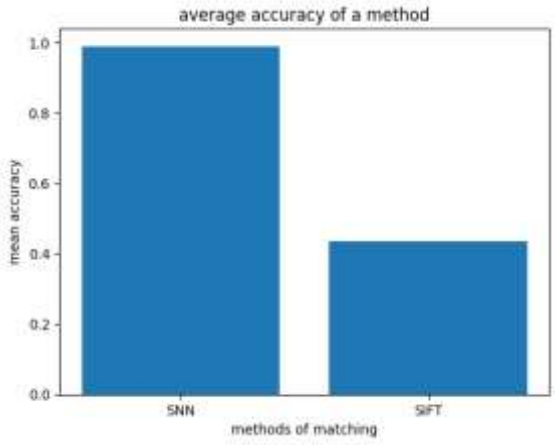
However, for real-time fingerprint identification on edge devices with low computing power, the model must be tuned. TensorFlow Lite tackles this by translating the learned SNN model to a format optimized for mobile and embedded device performance. This decreases computing needs while maintaining high precision, making the system cost-effective for deployment on resource-constrained devices.





Overall, the suggested technique provides an appealing solution for fingerprint recognition by:

- The SNN architecture combines the precision of SIFT with the efficiency of ORB, achieving high identification accuracy while preserving real-time processing rates appropriate for edge device deployment.
- TensorFlow Lite is used for cost-effective model deployment on devices with restricted resources.
- This combination provides a substantial advantage over systems that rely only on SIFT or ORB, achieving a balance of performance and cost for realistic fingerprint identification applications.



Conclusion

In this project, a robust fingerprint recognition system is successfully developed and implemented utilizing Siamese Neural Networks (SNN). This approach encompassed several key stages, from

meticulous data preparation and sophisticated model architecture design to rigorous training, evaluation, and optimization for real-time inference.

Key Achievements:

1. **Data Preparation**:

A labeled dataset of fingerprint images with detailed annotations is meticulously curated. This dataset formed is the backbone of this system, enabling precise mapping between input images and output identities.

2. **Model Architecture**:

The deployment of Siamese Neural Networks was instrumental in this system's ability to generate embeddings for fingerprint images. By comparing these embeddings using a distance metric, this model effectively determined fingerprint matches.

3. **Training Process**:

Through a carefully monitored training process, which included techniques like model checkpoints and early stopping, this SNN model learned to differentiate between matching and non-matching fingerprint pairs. This training ensured high accuracy and robustness in the system's performance.

4. **Evaluation and Testing**:

The proposed model underwent extensive evaluation using a separate test dataset. This phase was critical in verifying the model's generalization capabilities and understanding its performance metrics, such as precision, recall, and the insights provided by the confusion matrix.

5. **Optimization for Inference**:

To ensure the system's applicability in real-world scenarios, the final model for inference is optimized. Converting the model to TensorFlow Lite enabled its deployment on low-power mobile and embedded systems, making it highly efficient and versatile.

6. **Real-time Inference**:

The culmination of this project was the successful implementation of real-time fingerprint identification. Leveraging the TensorFlow Lite interpreter, the system achieved instant fingerprint recognition, proving its efficacy for applications requiring rapid authentication and secure access control.

Implications and Future Work:

The successful implementation of this fingerprint recognition system holds significant promise for enhancing security measures across various applications, including mobile authentication and access control systems. Future work could focus on expanding the dataset to include more diverse fingerprint samples, further optimizing the model for even lower-power devices, and integrating advanced techniques to improve the system's accuracy and robustness in more challenging real-world conditions. In conclusion, this project demonstrates the feasibility and effectiveness of using Siamese Neural Networks for fingerprint recognition, paving the way for secure, efficient, and real-time identification solutions.

References

1. Jain, A. K., Hong, L., & Pankanti, S. (2000). "Biometric identification." Communications of the ACM, 43(2), 91-98.
2. Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2003). "Handbook of Fingerprint Recognition." Springer.
3. Uludag, U., Pankanti, S., Prabhakar, S., & Jain, A. K. (2004). "Biometric cryptosystems: issues and challenges." Proceedings of the IEEE, 92(6), 948-960.
4. Ratha, N. K., Connell, J. H., & Bolle, R. M. (2007). "Enhancing security and privacy in biometrics-based authentication systems." IBM Systems Journal, 40(3), 614-634.

5. Lowe, D.G. (2004). "Distinctive image features from scale-invariant key points." International Journal of Computer Vision, 60(2), 91-110.
6. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). "ORB: An efficient alternative to SIFT or SURF." Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2564-2571.
7. Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., ... & Shah, R. (1994). "Signature verification using a 'siamese' time delay neural network." International Journal of Pattern Recognition and Artificial Intelligence, 7(4), 669-688.
8. Zhang, J., Yan, Y., Liu, C., & Wang, X. (2016). "FingerNet: An unified deep network for fingerprint minutiae extraction." Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), 2921-2925.
9. Jain, A. K., Ross, A., & Prabhakar, S. (2004). An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology, 14(1), 4-20.
10. Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese Neural Networks for One-shot Image Recognition. In ICML Deep Learning Workshop.
11. TensorFlow Lite. (n.d.). Retrieved from https://www.tensorflow.org/lite
12. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.
13. Waskom, M., et al. (2017). seaborn: statistical data visualization. Journal of Open Source Software, 2(9), 302.
14. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
15. Fowler, M. (2006). Continuous Integration. In _Design, build, and deliver: software projects successfully (pp. 152-166). Addison-Wesley.
16. Lowe, D.G. (1999). Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision (pp. 1150-1157).
17. Vedaldi, A., & Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms.
18. Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust Invariant Scalable Keypoints. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2548-2555).
19. Maltoni, D., Maio, D., Jain, A.K., & Prabhakar, S. (2009). Handbook of Fingerprint Recognition. Springer Science & Business Media.
20. Jain, A.K., Ross, A., & Pankanti, S. (2006). Biometrics: A tool for information security. IEEE Transactions on Information Forensics and Security, 1(2), 125-143.
21. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a "Siamese" time delay neural network. In Advances in Neural Information Processing Systems (pp. 737-744).