Development of Deep Learning Models Using CNN Methods for Fingerprint Biometric Identification in Attendance Systems

Mohammad Tio Rezky[1][4242201063], Peter Jordan Pontoh[2][4242201088], Iman Fahruzi[3][100007]

[123] Politeknik Negeri Batam, Department of Electrical Engineering, Batam 29461, Indonesia
iman@polibatam.ac.id

Abstract. With the development of technology, fingerprint biometric systems are increasingly being used because they have unique and permanent patterns that are effective for identifying individuals. However, most attendance systems still rely on template matching methods, which have limitations, such as sensitivity to noise, fingerprint position variations, lack of scalability, and inability to detect fake fingerprints. As a solution, this study developed a deep learning model based on Convolutional Neural Network (CNN) to improve the accuracy and reliability of attendance systems. The methods used include fingerprint data collection and preprocessing, CNN architecture design, and model training with an adequate dataset. Evaluation was conducted using accuracy, precision, recall, and F1-score. The test results show that the designed CNN model is capable of achieving an accuracy rate of 91.68%, making the resulting attendance system more secure, efficient, and accurate than conventional methods, and can be applied on campuses and in workplaces.

Keywords: Deep Learning, Convolutional Neural Network (CNN), Biometric Identification, Fingerprint, Attendance System, Image Processing

1 Introduction

Fingerprints are one of the most common and widely used forms of physiological biometrics in identification systems. Human fingerprints have very distinctive, complex patterns that are unique to each individual, making them one of the most reliable forms of biometrics in the identification process [1]. This uniqueness is permanent throughout life, making it difficult to imitate or forge [2]. In addition, the latest fingerprint sensors show high accuracy and adaptability to various scanning conditions, making them increasingly secure for various security and administrative applications [3]. With technological advances, the use of fingerprints has been applied more modernly in security, access control, and administration systems, especially in biometric attendance systems. This system enables automatic attendance recording through the process of scanning and matching fingerprints with a database, which has proven to be more efficient and secure than manual methods because it minimizes fraud such as proxy attendance and reduces the potential for recording errors [4], [5].

In a previous study entitled Designing Student Attendance Using Fingerprints on Raspberry Pi Based on the Internet of Things, a fingerprint-based attendance monitoring system was developed using the ID template method. In this study, the system was able to match scanned fingerprints with ID templates stored in the database, and the results were automatically recorded in the system. The use of ID templates enables an efficient identification [6]. However, the ID template-based approach still has limitations in terms of flexibility regarding data noise, variations in fingerprint position during scanning, scalability in large-scale data, and inability to recognize fake fingerprints.

In response to these limitations, various studies have begun to direct system development towards a deep learning-based approach. One of these was conducted [7], who implemented the Convolutional Neural Network method using TensorFlow for the fingerprint recognition process. CNN has proven to be capable of classifying fingerprint images more accurately due to its ability to recognize complex patterns through an automatic feature learning process. Meanwhile [8]. developed a CNN-based fingerprint recognition framework with a transfer learning approach using the ResNet50 architecture. This model is designed to recognize fingerprints directly from images without the need for manual feature extraction, and remains effective even when training data per class is limited. This approach shows that the use of CNN not only improves the accuracy of the system, but also makes it more flexible and scalable in various input conditions.

In contrast to previous studies, this work offers the distinct advantage of combining a deep learning approach with the design of a fully integrated biometric attendance system. The CNN model is developed not only for image classification but also for direct integration with hardware components such as the ESP32, AS608 fingerprint sensor, and a web-based monitoring platform. Furthermore, this research employs a larger and more representative SOCOFing dataset, supported by advanced preprocessing techniques using OpenCV, enabling the system to effectively recognize fingerprint patterns under diverse conditions [9], [10]. Model evaluation is also carried out comprehensively using accuracy, precision, recall, and F1-score metrics to ensure system reliability. The expected result of this research is the creation of an attendance system that is more secure, efficient, and accurate than conventional methods. By combining CNN, IoT, and biometric sensor technologies directly, this system is expected to reduce the rate of identification errors, detect attendance in real-time, and strengthen the integrity of attendance data.

1.1 Design System

The implementation stages in this system are arranged in a systematic workflow as shown in the flowchart. The process begins with a literature study as a basis for strengthening the theory. Next, fingerprint datasets are collected, then expanded through augmentation and divided into training data and testing data. The next stage involves preprocessing and configuring the CNN model so that the data is ready for use in training. The model is trained and evaluated repeatedly until good accuracy is achieved. Once the model has been proven reliable, system development and integration on the website is carried out, along with the connection of the fingerprint sensor module to support the main functions of the system. The final stage involves comprehensive testing to ensure the reliability and consistency of the system's performance, after which the entire process is documented in full.

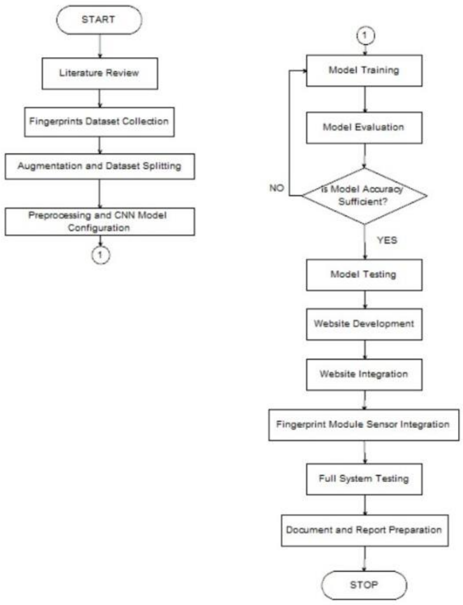

Fig. 1. Flowchart Design Implementation

1.2 Implementation

The architecture of the fingerprint-based attendance system is structured into three main stages: preprocessing, CNN model training, and system implementation. The workflow starts when the user places a finger on the sensor, after which the captured image is transmitted via the ESP32 module that functions as a bridge to the server for subsequent processing. During the preprocessing phase, several operations are applied to enhance image quality and ensure uniformity, including grayscale conversion, normalization, noise reduction, and binarization [11]. The processed images are then fed into the CNN model, where convolution layers extract relevant fingerprint features, supported by activation, normalization, dropout, and pooling layers to improve model stability and minimize overfitting (similar to the method Once the training phase is completed, the trained model is embedded into the attendance system so that every fingerprint scan is instantly compared with the learned feature representations. If a match is detected, the attendance record is saved in the database and displayed through an IoT-enabled web interface. Conversely, if no match is found, the system simply outputs the raw image without confirmation, ending the process. This workflow reflects practical implementations of biometric attendance systems discussed in recent literature [12].

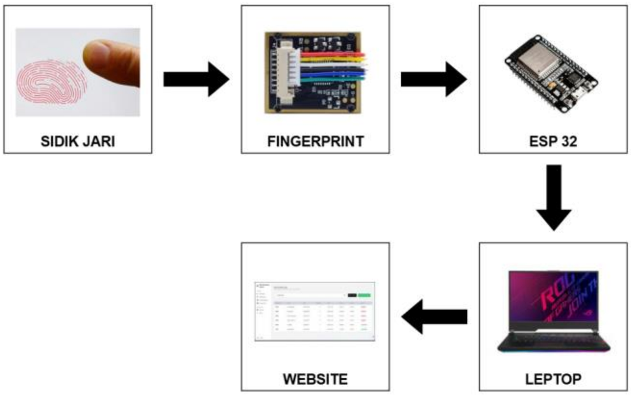

Fig. 2. Block Diagram

1.3 Dataset

The initial stage in the CNN model training process begins with the compilation of the dataset to be used. In this study, the data used was sourced from the Sokoto Coventry Fingerprint Dataset (SOCOFing), which is a collection of biometric fingerprint images developed specifically for academic research purposes. This dataset includes 6,000 fingerprint images from 600 individuals of African descent, and is supplemented with additional information such as user ID, gender, hand type, and finger name. In addition to the original data, SOCOFing also provides a version of the images that have been synthetically modified with three types of deformation, namely obliteration, central rotation, and Z-Cut. These modifications were made using the STRANGE toolbox to simulate interference or damage to the images while maintaining a resolution of 500 dpi. The image capture process was carried out using two types of sensors, namely Hamster Plus HSDU03PTM and SecuGen SDU03PTM, producing a total of 55,273 fingerprint images, each measuring 96 × 103 pixels and having one channel [13].

Fig. 3. Dataset

1.4 Preprocessing and Augmentation Data

Preprocessing is an important process because it involves preparing images before they are used for CNN model training. In this experiment, preprocessing was performed using the OpenCV library with a series of main steps aimed at improving image quality, starting with reading images in grayscale format. The images were then resized to 128×128 pixels to even out the dimensions so that they would be compatible with CNN model training. Next, the image undergoes a denoising process using the Non-Local Means (NLM) method, which serves to reduce noise in the image without removing important details in the fingerprint pattern. After the denoising process, image segmentation is performed using the inverse Otsu Thresholding method to convert the image into binary, making the fingerprint patterns appear more contrasting and sharp. The final stage is normalization, which is used to convert the pixel value range to a smaller range (0.0–1.0) in float32 format, reducing computation complexity and stabilizing CNN training convergence [14].

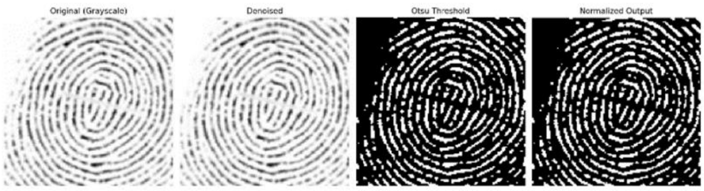

Fig. 4. Preprocessing

There is also data augmentation to multiply and vary the training dataset without having to collect new data. This technique is very important to use in CNN model training because it helps overcome limited data and prevents overfitting in the model. By performing augmentation, the model can learn from variations such as geometric transformations (rotation, zoom in/out, shift, shear) and photometric transformations (lighting adjustment and noise addition). These augmentation techniques play an important role in improving model performance and generalization, making the CNN more adaptive to unseen fingerprint variations [15].

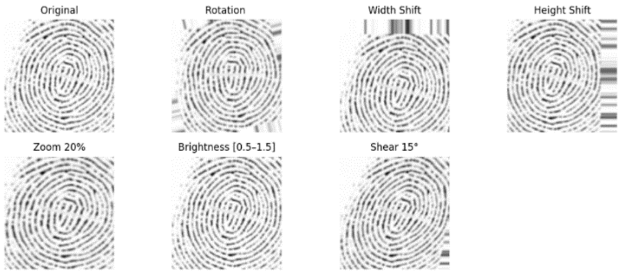

Fig. 5. Data Augmentation

1.5 Training Model CNN

The training process of the CNN model training is an important stage that aims to teach artificial neural networks to automatically recognize patterns and characteristics specific to fingerprint images. At this stage, fingerprint images first undergo a preprocessing stage, such as conversion to grayscale, normalization, noise removal, and binarization to optimize data quality. Next, the preprocessed images are processed through a CNN architecture consisting of a convolution layer to extract important features, followed by an activation layer, batch normalization, dropout, and pooling to improve stability and prevent overfitting. The training process is carried out through two main mechanisms, namely forward propagation to distribute input data to all layers of the network, and backpropagation to update weights so that the model can learn gradually. This model was trained in batches of 32 images for 100 epochs using the Adam optimizer algorithm, which is known to be efficient in handling small gradients and noise, and combined with a learning rate of 0.0001 to ensure stable weight updates. After the training process was complete, the resulting model was able to recognize fingerprint patterns better and was ready to be integrated into a real-time biometric attendance system.

2 Results

2.1 Training Result

In training the CNN model on this system, a dataset was used with an 80% split for training data and 20% for validation data, and went through a preprocessing stage involving conversion to grayscale, normalization, noise removal, and binarization. The model was trained in batches of 32 images for 100 epochs with the Adam optimizer and a learning rate of 0.0001, assisted by the EarlyStopping and ReduceLROnPlateau callbacks to avoid overfitting and smooth the learning process. The training results showed a consistent decrease in training loss and validation loss until the end of the epoch, while the training and validation accuracy graphs increased steadily to approach the maximum value, indicating that the training process was effective and stable. The confusion matrix shows a balanced prediction distribution with low classification errors, while the final evaluation resulted in an accuracy of 95.62% with an average precision, recall, and F1-score of 0.96 for both Real and Altered classes. This proves that the trained CNN model is capable of effectively extracting fingerprint features and has reliable performance for real-time fingerprint-based biometric identification systems.

```
📊 Final Evaluation:
Test Accuracy: 0.9562
Test Loss: 0.1214
214/214 ━━━━━━━━━━━━━━━━━━━━ 13s 61ms/step

Classification Report:
              precision    recall  f1-score   support

        Real       0.97      0.94      0.96      3414
     Altered       0.94      0.97      0.96      3413

    accuracy                           0.96      6827
   macro avg       0.96      0.96      0.96      6827
weighted avg       0.96      0.96      0.96      6827
```

Fig. 6. Final Evaluation Results Model

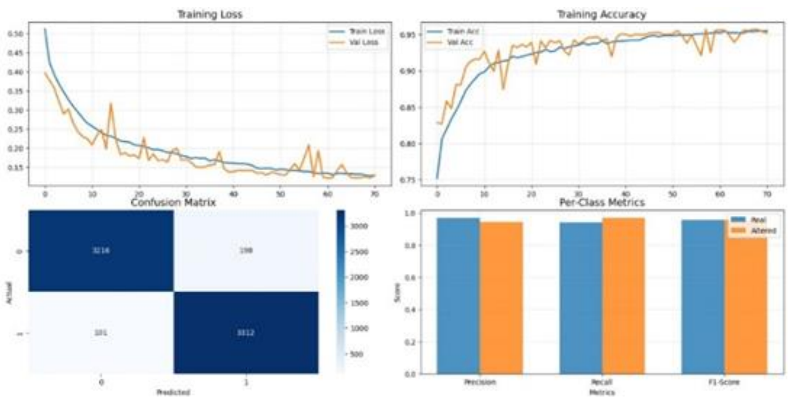

Fig. 7. Model Metric Evaluation

2.2 Testing System

The system testing was conducted using a fingerprint image dataset that had been divided from the outset, with the testing data prepared separately from the training and validation data. The dataset consists of 6,000 images, allocated as 80% for training and 20% for validation. In addition, the system testing also uses another fingerprint image dataset. The experimental results demonstrate that the system is capable of extracting fingerprint image features with good accuracy and presenting the features, confidence scores, and quality scores. Furthermore, the system was successfully integrated with a web-based IoT monitoring platform, and all results were recorded in the database.

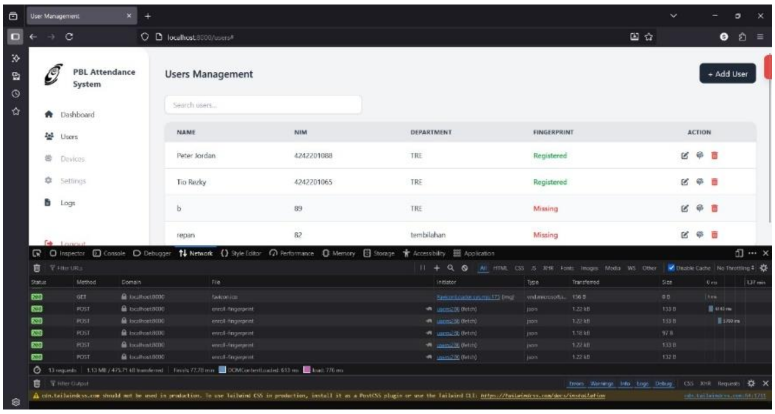

Fig. 8. Testing results

Table 1. Fingerprint Enrollment Results

Parameter	Value
Success	True
Feature Length	128
Quality	10.4017
Confidence	80%
Processing Time	0.1675 second
User ID	11
Template Name	null
Features	[0.0, 0.2879, 1.4095, ..., 1.8382]

Table 1. Final Evaluation Resul

Class	Precision	Recall	F1-Score	Support
Real	0.97	0.94	0.96	3414
Altred	0.94	0.97	0.96	3413
Accuracy	-	-	0.96	6822
Macro Avg	0.96	0.96	0.96	6822
Weighted Avg	0.96	0.96	0.96	6822

3 Conclusion

This study describes the development and evaluation of an automatic fingerprint biometric identification system for attendance systems using a Convolutional Neural Network. This method is designed to overcome the limitations of conventional template matching systems, such as sensitivity to noise, fingerprint position variations, scalability limitations, and the inability to detect fake fingerprints. Testing results show that the proposed system exhibits outstanding and reliable performance, with an overall accuracy rate of 95.62% and consistent Precision, Recall, and F1-score values of 0.96 across all class categories Real and Altered. The model's performance, which emphasizes automatic fingerprint feature extraction through deep learning, proves that this system has been effectively constructed for its main purpose. This CNN-based system has been successfully integrated with a web-based IoT monitoring platform, ESP32 module, and AS608 fingerprint sensor, resulting in a legitimate, precise, and highly promising solution for implementation in automatic attendance systems in campus and industrial environments. The system is capable of processing identification in 0.1675 seconds with a confidence level of 80%. To enhance further development, it is recommended that research focus on increasing dataset variety, exploring more novel model architectures, and expanding the system's capabilities to handle a wider range of scanning conditions to make the system more comprehensive and reliable in actual production settings or large-scale implementation in educational institutions and workplaces.

References

[1] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, “Security and accuracy of fingerprint-based biometrics: A review,” *Symmetry (Basel)*, vol. 11, no. 2, 2019, doi: 10.3390/sym11020141.

[2] A. K. Trivedi, “Fingerprint Orientation Estimation: Challenges and Opportunities,” Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.11563

[3] Y. Yu, Q. Niu, X. Li, J. Xue, W. Liu, and D. Lin, “A Review of Fingerprint Sensors: Mechanism, Characteristics, and Applications,” *Micromachines (Basel)*, vol. 14, no. 6, 2023, doi: 10.3390/mi14061253.

[4] P. Ahmad Raihan, M. Ramaddan Julianti Candidate, S. Ramdhan, and S. Fazilla Candidate, “International Journal of Communication Networks and Information Security IoT-Based Biometric Attendance System Using Arduino and ThingsBoard,” vol. 15, no. 4, pp. 103–117, 2023, [Online]. Available: https://ezproxy.niagara.edu/login?url=https://www.proquest.com/scholarly-journals/iot-based-biometric-attendance-system-using/docview/3049831727/se-2?accountid=28213

[5] A. Martulandi and D. Setiawan, “Sistem Kehadiran Biometrik Sidik Jari Menggunakan IoT yang Terintegrasi dengan Telegram,” *Engineering, MAthematics and Computer Science (EMACS) Journal*, vol. 3, no. 3, pp. 103–107, 2021, doi: 10.21512/emacsjournal.v3i3.7426.

[6] R. A. Rachman and E. Haryatmi, “RANCANG BANGUN ABSENSI MAHASISWA MENGGUNAKAN SIDIK JARI PADA RASPBERRY PI BERBASIS INTERNET OF THINGS (IOT) SECARA REAL TIME,” *Jurnal Ilmiah Informatika Komputer*, vol. 23, no. 3, pp. 154–165, 2018, doi: 10.35760/ik.2018.v23i3.2371.

[7] R. D. Nurfita, “Implementasi Deep Learning Berbasis Tensorflow,” *Jurnal Emitor*, vol. 18, no. 01, pp. 22–27, 2018.

[8] S. Minaee, E. Azimi, and A. Abdolrashidi, “FingerNet: Pushing The Limits of Fingerprint Recognition Using Convolutional Neural Network,” *Computer Vision and Pattern Recognition*, 2019, [Online]. Available: http://arxiv.org/abs/1907.12956

[9] E. Asem, L. M. Abouelmagd, A. E. Tolba, and S. Elmougy, “Biometric CNN Model for Verification Based on Blockchain and Hyperparameter Optimization,” *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00653-y.

[10] A. Hussian, F. Murshed, M. N. Alandoli, and G. Aljafari, “A Hybrid Deep Learning Approach for Secure Biometric Authentication Using Fingerprint Data,” *Computers*, vol. 14, no. 5, pp. 1–19, 2025, doi: 10.3390/computers14050178.

[11] A. Takahashi, Y. Koda, K. Ito, and T. Aoki, “Fingerprint feature extraction by combining texture, minutiae, and frequency spectrum using multi-task CNN,” *IJCB 2020 - IEEE/IAPR International Joint Conference on Biometrics*, 2020, doi: 10.1109/IJCB48548.2020.9304861.

[12] C. P. Dea Mirela Sirait, F. Parulian, and M. Iqbal, “Employee Presence System Using Nodemcu For Fingerprint And Face Id Internet Of Things (Iot),” *Jurnal Indonesia Sosial Teknologi*, vol. 4, no. 11, pp. 2014–2025, 2023, doi: 10.59141/jist.v4i11.795.

[13] Y. I. Shehu, A. Ruiz-Garcia, V. Palade, and A. James, “Sokoto Coventry Fingerprint Dataset,” vol. 103, pp. 5–7, 2018, [Online]. Available: http://arxiv.org/abs/1807.10609

[14] M. Hirsi Mohamed, “Fingerprint Classification Using Deep Convolutional Neural Network,” *Journal of Electrical and Electronic Engineering*, vol. 9, no. 5, p. 147, 2021, doi: 10.11648/j.jeee.20210905.11.

[15] R. Garg, G. Singh, A. Singh, and M. P. Singh, “Fingerprint recognition using convolution neural network with inversion and augmented techniques,” *Systems and Soft Computing*, vol. 6, no. May, p. 200106, 2024, doi: 10.1016/j.sasc.2024.200106.

standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting.

Springer Nature will add OpenAccess standard text here during typesetting.Springer Nature will add OpenAccess standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting. Springer Nature will add OpenAccess standard text here during typesetting.