Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2026, 2026:26

https://doi.org/10.28919/cmbn/9679

ISSN: 2052-2541

A COMPARATIVE ANALYSIS OF KERNEL FILTERING TECHNIQUES FOR CONVOLUTIONAL NEURAL NETWORKS IN FINGERPRINT RECOGNITION

JONATHAN LIM, DANI SUANDI*

Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, 11480, Indonesia

Copyright © 2026 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: Fingerprint recognition remains one of the most widely adopted biometric methods for individual verification due to its uniqueness and reliability. This study investigates the role of kernel-based filtering in the feature extraction stage and evaluates the performance of four Convolutional Neural Network (CNN) architectures: Siamese Network, LeNet, ResNet, and VGG-19. Experiments were conducted on the SOCOFing dataset, with preprocessing steps involving image resizing, normalization, and augmentation. To enhance the discriminative quality of fingerprint features, Sobel, Gabor, Laplacian, and Gaussian kernels were applied prior to model training. Model performance was assessed using accuracy, loss, and the area under the ROC curve (AUC), with AUC emphasized as the primary metric to address class imbalance. The results demonstrate that Gaussian and Sobel kernels consistently yield superior and stable performance across all models. Furthermore, statistical validation through Friedman and Nemenyi tests confirmed significant differences among kernel–model combinations, underscoring the benefit of multi-kernel preprocessing in biometric classification. Overall, the findings highlight the critical role of kernel selection in optimizing fingerprint recognition systems, with Gaussian filtering showing particular promise for enhancing adaptability, robustness, and classification accuracy.

Keywords: Fingerprint recognition, Feature extraction, Biometric classification, Statistical validation, Kernel-based filtering, Convolutional Neural Networks (CNNs).

2020 AMS Subject Classification: 68T07, 68T10, 62H30, 94A08.

*Corresponding author

E-mail address: dani.suandi@binus.ac.id

Received November 04, 2025

1. INTRODUCTION

In the digital era, the demand for secure, efficient, and reliable identity verification systems has grown substantially, particularly in critical sectors such as data protection, financial transactions, border security, and access control [1-3]. Biometric technologies have emerged as a dominant solution for ensuring trustworthy authentication, with fingerprint recognition being one of the most widely adopted modalities due to its uniqueness, permanence, universality, and ease of acquisition [4-5]. Despite its widespread use, fingerprint recognition still encounters notable challenges. Issues such as low-quality image capture, susceptibility to spoofing attacks, and intra-class similarities in ridge patterns can undermine recognition accuracy and limit system robustness [6-7]. Addressing these challenges requires methodological advances in both feature extraction and classification strategies to maintain high performance under diverse operational conditions.

A key component in enhancing fingerprint recognition performance lies in preprocessing techniques that improve image quality prior to classification. Previous studies have demonstrated that appropriate preprocessing can amplify discriminative ridge–valley features, reduce noise, and mitigate distortions, ultimately boosting recognition accuracy across various models [8-9]. However, identifying preprocessing methods that generalize effectively across different deep learning architectures remains an open research problem. The need for adaptive and robust preprocessing is particularly pressing given the variability of fingerprint datasets, which may include noise from acquisition devices, partial prints, and environmental interference [10-11].

Recent advancements in deep learning have provided powerful tools for addressing these challenges. Convolutional Neural Networks (CNNs), in particular, have shown remarkable capabilities in automatically learning hierarchical and discriminative representations from raw image data [12,13]. CNN variants such as LeNet, VGG-19, ResNet, and Siamese Networks have demonstrated strong performance in biometric recognition tasks, offering robust classification and generalization across varying datasets [14,15,16,17]. Nevertheless, the quality of input features remains a determining factor in CNN effectiveness. Kernel-based filtering, as a preprocessing strategy, has been widely employed to enhance fingerprint image features by highlighting edges, suppressing noise, and preserving local structural details [9, 18, 19]. Classical kernels such as Sobel, Gabor, Laplacian, and Gaussian filters remain highly relevant due to their interpretability and proven capacity to strengthen image representations for learning-based models [8, 20].

Building upon these insights, this study systematically investigates the effectiveness of kernel-based filtering in CNN-driven fingerprint recognition. Using the SOCOFing dataset [21], we

evaluate the performance of multiple CNN architectures—Siamese Network, LeNet, ResNet, and VGG-19—under different preprocessing configurations. Model performance is assessed using accuracy, loss, and the area under the ROC curve (AUC), with AUC prioritized to account for class imbalance [8, 22, 23]. Furthermore, statistical validation through Friedman and Nemenyi tests is employed to establish the significance of observed differences between kernel–model combinations [24, 25]. To the best of our knowledge, few studies have provided a systematic and statistically validated comparison of kernel filtering techniques across multiple CNN architectures in fingerprint recognition. This work fills that gap by offering evidence-based insights into the effectiveness of multikernel preprocessing and by identifying Gaussian filtering as a particularly promising strategy for enhancing adaptability, robustness, and classification accuracy.

2. METHODS

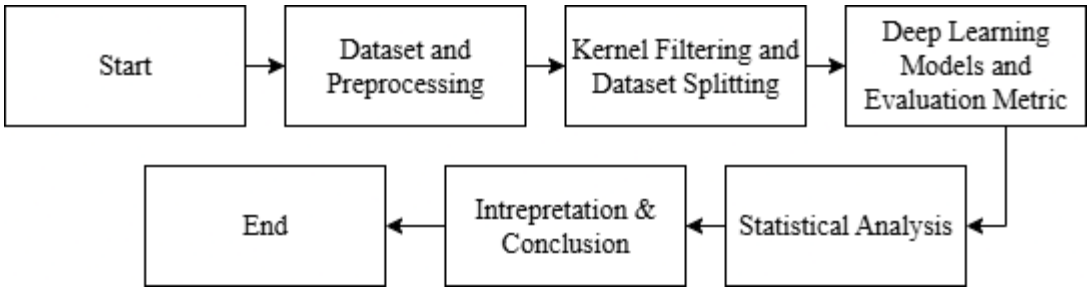

Figure 1. Research workflow

The methodology of this study was designed to systematically evaluate the effect of kernel-based preprocessing on the performance of several Convolutional Neural Network (CNN) architectures for fingerprint recognition. As illustrated in Figure 1, the research workflow comprises six main stages, beginning with dataset acquisition and preprocessing. The subsequent stage involves the application of kernel filtering, followed by dataset splitting to prepare training and testing subsets. Thereafter, multiple CNN models—namely Siamese Network, LeNet, ResNet, and VGG-19—are trained and evaluated using performance metrics including accuracy, loss, and the area under the ROC curve (AUC). To ensure the reliability of the findings, statistical analyses are then performed using the Friedman and Nemenyi tests, which enable objective comparisons among kernel–model configurations. Finally, the results are synthesized and interpreted to derive conclusions regarding the effectiveness of kernel-based preprocessing in improving fingerprint recognition performance.

2.1. Dataset and Preprocessing

The Sokoto Conventry Fingerprint (SOCOFing) dataset [21] was utilized in this study. This publicly available dataset can be accessed at https://www.kaggle.com/datasets/ruizgara/socofing. It contains 150,000 grayscale fingerprint images representing various individuals under different conditions, such as gender, fingerprint class, and level of alteration. Each image was resized to a uniform resolution of 96 × 96 pixels to ensure consistency and compatibility across all model architectures. To reduce the risk of overfitting and enhance the generalizability of the models, a series of preprocessing steps were applied. These included normalization to scale pixel intensity values between 0 and 1, and data augmentation techniques such as random rotations and zooming. These transformations aimed to simulate real-world variations and improve the robustness of the training process by increasing the diversity of input data.

Each fingerprint image was renamed for better indexing and saved as NumPy arrays for processing [26]. All photos were scaled to a defined resolution of 96 × 96 pixels to ensure consistent input dimensions across model topologies. To stabilize training and increase convergence, pixel intensity data were standardized to the range [0, 1] using Min-Max normalization [27].

Figure 2. 100__M_Left_index_finger.BMP

A sample fingerprint image from the SOCOFing dataset with the file name format 100__M_Left_index_finger.BMP is seen in Figure 2. In order to organize and classify the dataset, metadata such as subject ID, gender, hand, and finger type are encoded in the file name. File name and picture information attributes are described in Table 1. An example of a real (unaltered) image is provided, showing a male subject's left hand and index finger. This metadata makes up the entire dataset used for model training and evaluation, along with the pixel data from the grayscale image.

Table 1. Example Dataset

Attribute	Description
File Name	100__M_Left_index_finger.BMP
Gender	Male
Left / Right Hand	Left
Finger Name	Index
Type	Real Image

To improve model generalization and prevent overfitting, data augmentation techniques like random rotations, horizontal flipping, zooming, cropping, contrast modification, and translation were used. These changes intended to emulate real-world acquisition situations, boosting the model's capacity to recognize fingerprints in a variety of scenarios. To reduce overfitting and enhance generalization, multiple data augmentation techniques were applied, including random rotations, horizontal flipping, zooming, cropping, contrast adjustment, and translation. These transformations aimed to simulate real-world variations in fingerprint acquisition and to improve model robustness in diverse scenarios [8, 28].

2.2 Kernel Filtering and Dataset Splitting

Prior to model training, a multi-kernel filtering technique was applied to enhance feature representation. In addition to the original unfiltered images, four distinct kernels—Sobel, Gabor, Laplacian, and Gaussian—were utilized. The Sobel filter emphasized image edges and suppressed background noise, thereby facilitating more accurate detection of structural patterns. The Gabor filter improved the visibility of fine ridge details essential for fingerprint recognition. The Laplacian filter amplified high-frequency components, resulting in sharper ridge boundaries and enhanced contrast. Conversely, the Gaussian filter smoothed the images to mitigate noise while preserving critical features. By applying these filters, multiple feature-enhanced versions of each fingerprint image were generated, improving robustness to variations in image quality and increasing the diversity of inputs for model training.

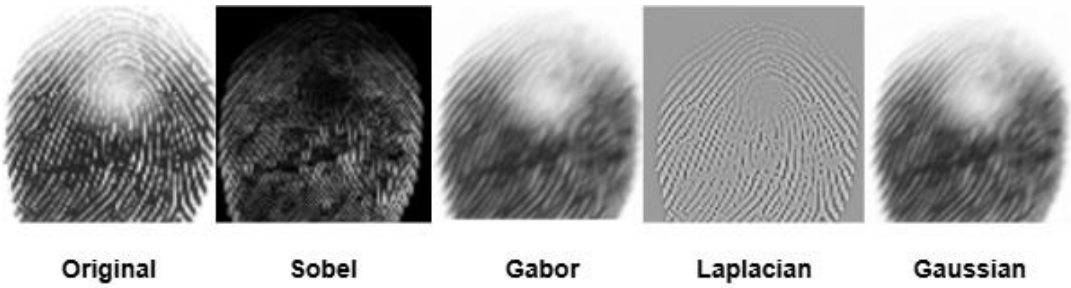

Figure 3. Fingerprint After Applying Kernel

Figure 3 presents a comparison between the original fingerprint image and its counterparts after applying each kernel filter. The Laplacian filter enhances contrast along ridge boundaries, the Gaussian filter suppresses noise while maintaining the overall ridge pattern, the Gabor filter accentuates fine ridge details, and the Sobel filter produces sharp edge delineations. Each kernel exerts a distinct influence on texture representation and ridge visibility, which, as illustrated by these visual results, can directly affect the overall classification performance.

2.3 Deep Learning Models and Evaluation Metrics

Convolutional Neural Network (CNN) architectures, including LeNet, VGG-19, ResNet, and the Siamese Network, were employed to classify biometric images. Model performance was primarily evaluated using the Area Under the Curve (AUC) metric, complemented by accuracy and loss measurements. Given the class imbalance in the dataset, AUC was prioritized as the main performance indicator [22, 23]. The Siamese Network directly compared pairs of fingerprint images by computing similarity scores, eliminating the need for explicit class labels. LeNet, a lightweight model, was selected for its computational efficiency and suitability for small grayscale images. ResNet-50 utilized residual connections to facilitate deeper network training while maintaining performance through the use of pre-trained weights via transfer learning. VGG-19, characterized by its consistent 3×3 convolutional filters, was also integrated with transfer learning to effectively capture spatial features across multiple layers. A detailed summary of each model architecture is presented in Tables 2.

Table 2. Summary Model

Model	Architecture		Parameter
	Layer	Output	
Siamese	Input 1	(96, 96, 1)	0
	Input 2	(96, 96, 1)	0
	Functional	(64)	8,012,736
	Substract	(64)	0
	Dense_2	(128)	8,320
	Dense_3	(64)	8,256
	Dense_4	(1)	65
LeNet	Input 1	(96, 96, 1)	0
	Input 2	(96, 96, 1)	0
	Functional	(64)	942,221
	Substract	(64)	0
	Dense_2	(128)	256
	Dense_3	(64)	8,256
	Dense_4	(1)	65
ResNet	Input 1	(96, 96, 3)	0
	Input 2	(96, 96, 3)	0
	Functional	(1)	25,955,457
	Substract	(1)	0
	Dense_2	(128)	256
	Dense_3	(64)	8,256
	Dense_4	(1)	65
VGG-19	Input 1	(96, 96, 3)	0
	Input 2	(96, 96, 3)	0
	Functional	(1)	25,955,457
	Substract	(1)	0
	Dense_2	(128)	256
	Dense_3	(64)	8,256
	Dense_4	(1)	65

The architecture of the Siamese Network, designed to process two 96×96×1 grayscale image inputs, is summarized in Table 2. The Functional layer contains the highest number of parameters (8,012,736), followed by multiple Dense layers that produce a single similarity score, with a Subtract operation used to compute feature differences. Similarly, the LeNet architecture—also operating on 96×96×1 grayscale images—demonstrates efficiency for small-scale datasets due to its compact design, comprising only 942,221 parameters within the Functional layer. The ResNet-50 model, which utilizes 96×96×3 RGB images as input, includes 25,955,457 parameters in the Functional layer. Through transfer learning, its residual connections facilitate deeper training while maintaining optimal performance. The VGG-19 architecture, also employing 96×96×3 RGB inputs and an equivalent parameter count to ResNet-50, is distinguished by its uniform 3×3 convolutional filters, enabling effective spatial feature extraction across layers.

Model training was conducted using the Adam optimizer, with each CNN architecture evaluated across five image types—original, Sobel, Gabor, Laplacian, and Gaussian. The primary evaluation metric was the Area Under the Receiver Operating Characteristic Curve (AUC), chosen due to the class imbalance in the dataset. Accuracy and loss metrics were also recorded to provide additional insights into model behavior. To assess the consistency of AUC values, bootstrap resampling was performed 100 times on the test dataset, using predefined indices (0, 24, 49, 74, and 99) for standardized comparison.

2.4 Statistical Analysis

To ensure a rigorous and unbiased evaluation, statistical analyses were performed using the Friedman and Nemenyi tests alongside key performance indicators such as AUC, accuracy, and loss. These non-parametric methods are particularly well-suited for comparing multiple models across diverse experimental conditions, as they do not assume normality in data distribution. The Friedman test was initially applied to identify overall performance differences among the kernel–model combinations. When significant disparities were detected, the Nemenyi post-hoc test was subsequently used to pinpoint specific model pairs exhibiting statistically significant differences [24]. This two-tiered analytical approach enhances the robustness and interpretability of the findings, ensuring that observed performance variations reflect true model behavior rather than random fluctuations.

3. RESULTS AND DISCUSSION

3.1. Model Accuracy Across Kernel Configurations

Each of the four CNN architectures was evaluated under five different image conditions: four kernel-filtered variations (Sobel, Gabor, Laplacian, and Gaussian) and one unfiltered (original) version. Following training for five epochs, the resulting accuracy values were aggregated, as illustrated in Figure 4 and summarized in detail in Table 3.

Table 3. Accuracy results for all CNN models across kernel filters

Kernel	Siamese	LeNet	ResNet	VGG-19
Non-kernel	0.889	0.83	0.501	0.5
Sobel	0.867	0.782	0.504	0.833
Gabor	0.9	0.833	0.515	0.499
Laplacian	0.899	0.779	0.505	0.49
Gaussian	0.908	0.833	0.501	0.822

Table 3 presents a comparative analysis of classification accuracy across four CNN architectures—Siamese Network, LeNet, ResNet, and VGG-19—each evaluated under five image conditions: original (non-kernel), Sobel, Gabor, Laplacian, and Gaussian filters. The Siamese Network achieved the highest overall performance, reaching an accuracy of 0.909 with Gaussian-filtered images, slightly outperforming the non-kernel (0.889) and Sobel (0.900) conditions. This indicates that Gaussian filtering effectively enhances feature clarity by reducing noise while preserving critical ridge information. The LeNet model exhibited stable performance, with accuracies ranging from 0.77 to 0.83, achieving its best results under Sobel and Gabor filtering (both 0.833). These results suggest that LeNet, due to its lightweight structure, benefits from filters emphasizing edge and texture information. In contrast, the ResNet model demonstrated relatively uniform and lower performance across all kernel variations (0.501–0.555), peaking under the Laplacian filter. The limited improvement observed in ResNet may stem from its high model complexity and insufficient dataset size to fully leverage its deep architecture. The VGG-19 model displayed notable improvement when using Sobel filtering (0.833) compared to the original image (0.500), highlighting the filter's ability to strengthen spatial feature representation in deep convolutional layers. Overall, Sobel and Gaussian filters consistently enhanced accuracy across most models, emphasizing that edge enhancement and noise reduction play a crucial role in improving fingerprint image classification performance.

3.2. Area Under Curve (AUC) as Primary Evaluation Metric

Due to the class imbalance within the dataset, the Area Under the Curve (AUC) was selected as the principal evaluation metric rather than accuracy. AUC offers a more reliable measure of

each model's capacity to discriminate between matched and unmatched fingerprint pairs. The comparative performance of all models across different kernel configurations is summarized in Table 4.

Table 4. AUC results for all CNN models across kernel filters

Kernel	Siamese	LeNet	ResNet	VGG-19
Non-kernel	0.938	0.912	0.5	0.5
Sobel	0.922	0.901	0.5	0.928
Gabor	0.94	0.906	0.5	0.5
Laplacian	0.932	0.916	0.5	0.5
Gaussian	0.94	0.919	0.5	0.909

The Siamese Network achieved consistently high AUC values across all kernel types, ranging from 0.922 to 0.940, with the *Gaussian* filter yielding the best performance. This demonstrates the model's robustness and its capacity to benefit from noise reduction and edge-preserving smoothing introduced by Gaussian filtering. The LeNet model also showed strong and stable performance, with AUC values between 0.901 and 0.919, slightly improving under *Gabor* and *Gaussian* filters. These findings suggest that LeNet effectively leverages feature enhancement from edge and texture-sensitive filters to improve fingerprint pair discrimination.

In contrast, ResNet exhibited uniformly low AUC scores (0.5 across all kernels), indicating random classification performance. This result may stem from overparameterization relative to the dataset size or inadequate fine-tuning of pre-trained layers, which prevented the model from effectively learning relevant fingerprint features. Meanwhile, the VGG-19 architecture demonstrated a marked improvement when using the *Sobel* and *Gaussian* filters, achieving AUC values of 0.928 and 0.909, respectively—significantly outperforming the non-kernel condition (0.5). This suggests that deep convolutional models like VGG-19 benefit substantially from pre-filtering methods that enhance edge information, which aligns with their hierarchical spatial feature extraction capabilities.

Overall, the results underscore that filter selection plays a crucial role in enhancing model discriminability, particularly for architectures that rely heavily on spatial and texture cues. The *Gaussian* and *Sobel* filters consistently improved AUC across models, confirming their effectiveness in producing cleaner and more structurally informative fingerprint representations for CNN-based classification.

3.3. Loss Metric Analysis

Table 5 presents the loss values obtained for all CNN architectures—Siamese Network, LeNet, ResNet, and VGG-19—across different kernel filtering methods (non-kernel, Sobel, Gabor, Laplacian, and Gaussian). The loss metric reflects the deviation between predicted and actual classifications, where lower values indicate better model convergence and higher predictive accuracy.

Table 5. Loss values for all CNN models across kernel filters

Kernel	Siamese	LeNet	ResNet	VGG-19
Non-kernel	0.304	0.433	0.693	0.693
Sobel	0.345	0.485	0.693	0.41
Gabor	0.282	0.423	0.693	0.693
Laplacian	0.287	0.533	0.693	0.693
Gaussian	0.266	0.413	0.693	0.449

The Siamese Network consistently achieved the lowest loss values among all models, ranging from 0.266 to 0.345, with the *Gaussian* filter producing the minimal loss (0.266). This finding reinforces the earlier AUC results, demonstrating that Gaussian filtering effectively enhances feature clarity and reduces noise, leading to improved model optimization. Similarly, LeNet exhibited moderate loss values between 0.413 and 0.533, with the lowest observed under the *Gaussian* filter. This indicates that even lightweight models benefit from noise reduction and edge-preserving preprocessing, which facilitates stable learning across convolutional layers.

In contrast, ResNet maintained a constant loss value of 0.693 across all kernel settings, signifying poor convergence and near-random performance. This outcome aligns with its AUC results and suggests that the model's complexity and depth were not well-suited to the dataset size or training configuration, preventing effective learning of discriminative features. Meanwhile, VGG-19 displayed substantial variation in loss values across kernels, achieving its best performance with the *Sobel* (0.410) and *Gaussian* (0.449) filters—significantly lower than the non-kernel baseline (0.693). This implies that VGG-19 benefits from preprocessing methods that enhance ridge and edge structures, which complement its deep hierarchical feature extraction process.

Overall, the results confirm that Gaussian and Sobel filters consistently yield lower loss values, indicating improved learning stability and generalization across models. Conversely, the uniform high loss of ResNet highlights the importance of model–dataset compatibility and the necessity of appropriate hyperparameter tuning for deep architectures in biometric image classification tasks.

3.4. Statistical Significance Testing

To further validate the performance variations across kernel filtering methods, Friedman and Nemenyi post-hoc tests were performed as in [29] using AUC values obtained from 100 bootstrap resamplings of the test data set. The Friedman test confirmed that the performance differences among kernels were statistically significant for both the LeNet and Siamese Network models ($p < 0.05$), indicating that kernel selection had a measurable effect on classification outcomes.

Figure 4 presents the Nemenyi heatmaps for LeNet (left) and Siamese (right), illustrating the pairwise statistical significance between kernel configurations. The color scale represents the magnitude of the *p*-values, where darker red tones indicate smaller differences (high similarity) and lighter blue tones denote larger, statistically significant differences between kernel performances. For the LeNet model, the Gaussian kernel demonstrated a clear advantage, showing strong dissimilarity (low *p*-values) compared to Laplacian and non-kernel configurations, as evidenced by the lighter blue cells in the heatmap. This suggests that Gaussian filtering substantially improved LeNet's ability to extract meaningful edge and texture features, leading to more stable classification results. In contrast, Sobel and Gabor kernels exhibited strong similarity (dark red regions), implying comparable performance in this architecture.

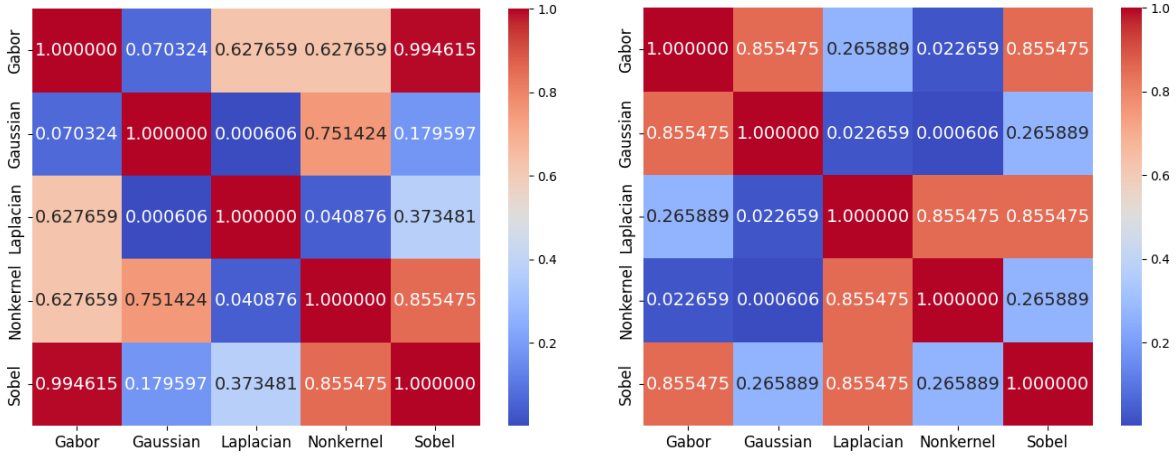

Figure 4. Nemenyi heatmap for Lenet (Left) and Siamese (Right)

In the Siamese Network, the Nemenyi test results also revealed that Gaussian and Gabor filters achieved notably better performance compared to Laplacian and non-kernel setups. The strong similarity between Sobel and Gabor, as shown by their high correlation (dark red), suggests that both filters capture complementary spatial gradients that enhance the pairwise matching capability of the Siamese architecture. Overall, these findings highlight that Gaussian and Gabor kernels

consistently provide statistically significant improvements in model performance across architectures, whereas Laplacian and non-kernel approaches yield weaker results. The Nemenyi heatmaps thus confirm that the Gaussian kernel remains the most effective filtering strategy for optimizing CNN-based fingerprint matching models.

3.5. Scoring System and Kernel Ranking

To obtain a comprehensive evaluation of kernel performance across all convolutional models, a composite scoring framework was developed by integrating three normalized performance metrics—accuracy, AUC, and loss—for each model–kernel configuration. The individual metric values were first derived from the evaluation results of each CNN architecture and subsequently normalized to ensure comparability across different scales. Normalization was conducted using the Min–Max scaling approach, defined as:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where $x_{\max}$ and $x_{\min}$ denote the maximum and minimum values of each respective metric across all model–kernel pairs. Since lower loss indicates superior performance, an inversion transformation was applied to the loss metric, expressed as:

$$Loss_{norm} = \frac{Loss_{max} - Loss}{Loss_{max} - Loss_{min}}$$

Finally, an aggregated performance score was computed according to the weighted formula.

$$Scoring = 0.5(AUC_{norm}) + 0.3(Accuracy_{norm}) + 0.2(Loss_{norm})$$

which assigns greater importance to AUC, followed by accuracy and loss. This scoring methodology ensures a balanced and unbiased assessment, allowing for fair comparison of kernel performance across architectures without favoring any single metric.

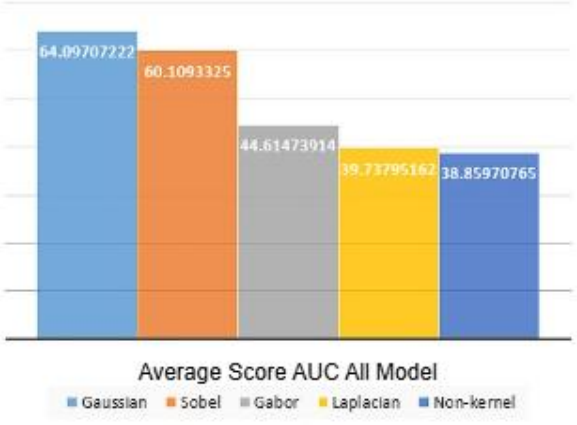

Figure 5. Average normalized score for each kernel

As illustrated in Figure 5, the Gaussian kernel achieved the highest overall average normalized score (64.10), demonstrating superior generalization and stability across all CNN models. The Sobel kernel followed with a slightly lower score (60.11), indicating strong yet less consistent performance. In contrast, the Gabor and Laplacian kernels yielded moderate average scores (44.61 and 39.74, respectively), while the non-kernel (baseline) configuration showed the weakest performance (38.86). These findings highlight the robustness and adaptability of Gaussian filtering, which effectively enhances feature representation by smoothing noise while retaining critical ridge and valley structures. The results reaffirm that the Gaussian kernel not only contributes to improved classification accuracy and AUC but also provides the most balanced and reliable performance across diverse CNN architectures.

3.6. Robustness of Gaussian Kernel

To provide a more focused evaluation of kernel robustness, Figure 6 illustrates the performance of the Gaussian kernel across four CNN architectures: Siamese, LeNet, ResNet, and VGG-19. The figure presents three key performance indicators—loss, AUC, and accuracy—to comprehensively assess the Gaussian kernel's effectiveness in fingerprint classification.

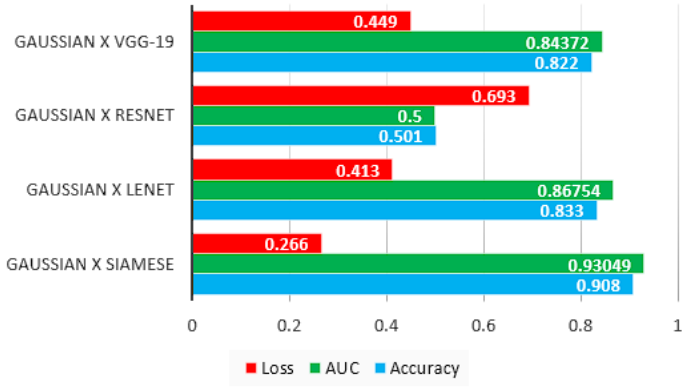

Figure 6. Gaussian kernel performance across all models.

As shown in the figure, the Siamese network achieved the best overall performance with the Gaussian filter, recording the lowest loss value (0.266) and the highest AUC (0.93049) and accuracy (0.908). These results indicate that the Gaussian kernel substantially enhances the model's discriminative capability in identifying fingerprint similarities. The LeNet model followed closely, with a loss of 0.413, AUC of 0.86754, and accuracy of 0.833, confirming that Gaussian filtering effectively strengthens mid-level feature extraction even in simpler convolutional architectures. In contrast, the ResNet model demonstrated a higher loss value (0.693) with moderate AUC (0.501) and accuracy (0.5), suggesting limited improvement when applying

Gaussian filtering to deeper architectures with inherent normalization and skip connections. The VGG-19 model performed relatively well with an AUC of 0.84372 and accuracy of 0.822, but its loss (0.449) remained higher than that of Siamese and LeNet, implying that the benefits of Gaussian filtering diminish in more complex networks with pre-trained feature hierarchies. These findings reaffirm that the Gaussian kernel provides a robust balance between noise suppression and feature preservation, leading to higher recognition accuracy and stability across most CNN architectures. Its ability to smooth irrelevant variations while retaining crucial ridge and valley structures makes it particularly effective for fingerprint representation learning, especially in architectures such as Siamese and LeNet, where fine-grained local features play a dominant role in model performance.

4. CONCLUSION

This study demonstrates that kernel-based preprocessing significantly influences CNN performance in fingerprint recognition tasks. Gaussian filtering consistently achieved superior results, particularly with Siamese and LeNet architectures, while Sobel provided stable performance across most models. Statistical validation confirmed that these differences are significant, underscoring the critical role of preprocessing in biometric systems. Additionally, architecture-specific limitations were observed, such as ResNet's reduced compatibility with grayscale images and the marginal benefits of deeper layers in VGG-19. Future research should explore adaptive kernel selection strategies, integration with attention mechanisms, and cross-dataset evaluations to enhance generalizability. Investigating hybrid approaches that combine multiple kernels or leverage learnable filters could further optimize feature extraction. These directions aim to develop more accurate, robust, and scalable biometric authentication frameworks.

DATA AVAILABILITY STATEMENT

The dataset and base code in this research are publicly available on Kaggle at https://www.kaggle.com/datasets/ruizgara/socofing.

CREDIT AUTHORSHIP CONTRIBUTION

Jonathan Lim: Writing – original draft, Resources, Data Curation, Visualization, Investigation, Formal analysis.

Dani Suandi: Writing – review & editing, Methodology, Conceptualization, Investigation, Supervision.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] H. Jung, S. Sim, H. Lee, Biometric Authentication Security Enhancement Under Quantum Dot Light-Emitting Diode Display via Fingerprint Imaging and Temperature Sensing, Sci. Rep. 13 (2023), 794. https://doi.org/10.1038/s41598-023-28162-6.

[2] M. Jasem Mahammad, Y. Al Mashhadany, A. Mohammad Eyada, Real Time Model of a Lock Gate System Upon Fingerprint with Microcontroller, Indones. J. Electr. Eng. Comput. Sci. 21 (2021), 37-43. https://doi.org/10.11591/ijeecs.v21.i1.pp37-43.

[3] D. Van De Ville, Y. Farouj, M.G. Preti, R. Liégeois, E. Amico, When Makes You Unique: Temporality of the Human Brain Fingerprint, Sci. Adv. 7 (2021), eabj0751. https://doi.org/10.1126/sciadv.abj0751.

[4] K. Li, D. Wu, L. Ai, Y. Luo, The Influence of Close Non‐Match Fingerprints Similar in Delta Regions of Whorls on Fingerprint Identification, J. Forensic Sci. 66 (2021), 1482-1494. https://doi.org/10.1111/1556-4029.14698.

[5] F. Saeed, M. Hussain, H.A. Aboalsamh, Automatic Fingerprint Classification Using Deep Learning Technology (DeepFKTNet), Mathematics 10 (2022), 1285. https://doi.org/10.3390/math10081285.

[6] J.B. Li, Z.Y. Chen, X. Li, L.J. Jing, Y.P. Zhang, et al., Feedback on a Shared Big Dataset for Intelligent TBM Part I: Feature Extraction and Machine Learning Methods, Undergr. Space 11 (2023), 1-25. https://doi.org/10.1016/j.undsp.2023.01.001.

[7] J.M. Shrein, Fingerprint Classification Using Convolutional Neural Networks and Ridge Orientation Images, in: 2017 IEEE Symposium Series on Computational Intelligence, IEEE, 2017, pp. 1-8. https://doi.org/10.1109/SSCI.2017.8285375.

[8] A. Genovese, V. Piuri, K.N. Plataniotis, F. Scotti, PalmNet: Gabor-Pca Convolutional Networks for Touchless Palmprint Recognition, IEEE Trans. Inf. Forensics Secur. 14 (2019), 3160-3174. https://doi.org/10.1109/TIFS.2019.2911165.

[9] N. Kumar, B. Singh, B.K. Panigrahi, C. Chakraborty, H.M. Suryawanshi, et al., Integration of Solar PV with Low-Voltage Weak Grid System: Using Normalized Laplacian Kernel Adaptive Kalman Filter and Learning Based InC Algorithm, IEEE Trans. Power Electron. 34 (2019), 10746-10758. https://doi.org/10.1109/TPEL.2019.2898319.

[10] S. Minaee, E. Azimi, A. Abdolrashidi, FingerNet: Pushing the Limits of Fingerprint Recognition Using Convolutional Neural Network, arXiv:1907.12956, 2019. https://doi.org/10.48550/arXiv.1907.12956.

[11] J. Sulam, R. Ben-Ari, P. Kisilev, Maximizing AUC With Deep Learning for Classification of Imbalanced

Mammogram Datasets, in: VCBM 2017 - Eurographics Workshop on Visual Computing for Biology and Medicine, 2017: pp. 131-135. https://doi.org/10.2312/vcbm.20171246.

[12] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, et al., CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope, Electronics 10 (2021), 2470. https://doi.org/10.3390/electronics10202470.

[13] N. Dey, Y.D. Zhang, V. Rajinikanth, R. Pugalenthi, N.S.M. Raja, Customized VGG19 Architecture for Pneumonia Detection in Chest X-Rays, Pattern Recognit. Lett. 143 (2021), 67-74. https://doi.org/10.1016/j.patrec.2020.12.010.

[14] U. Chaudhuri, B. Banerjee, A. Bhattacharya, Siamese Graph Convolutional Network for Content Based Remote Sensing Image Retrieval, Comput. Vis. Image Underst. 184 (2019), 22-30. https://doi.org/10.1016/j.cviu.2019.04.004.

[15] G. Wei, G. Li, J. Zhao, A. He, Development of a Lenet-5 Gas Identification CNN Structure for Electronic Noses, Sensors 19 (2019), 217. https://doi.org/10.3390/s19010217.

[16] R. Wightman, H. Touvron, H. Jégou, ResNet Strikes Back: An Improved Training Procedure in Timm, arXiv: 2110.00476, 2021. https://doi.org/10.48550/arXiv.2110.00476.

[17] Z. Wu, C. Shen, A. van den Hengel, Wider or Deeper: Revisiting the ResNet Model for Visual Recognition, Pattern Recognit. 90 (2019), 119-133. https://doi.org/10.1016/j.patcog.2019.01.006.

[18] H. Jiang, H. Wang, W. Hu, D. Kakde, A. Chaudhuri, Fast Incremental SVDD Learning Algorithm with the Gaussian Kernel, Proc. AAAI Conf. Artif. Intell. 33 (2019), 3991-3998. https://doi.org/10.1609/aaai.v33i01.33013991.

[19] R.G. Zhou, D.Q. Liu, Quantum Image Edge Extraction Based on Improved Sobel Operator, Int. J. Theor. Phys. 58 (2019), 2969-2985. https://doi.org/10.1007/s10773-019-04177-6.

[20] A. sousanabadi Farahani, M. modabbarifar, Investigating the Effect of Machining Parameters on the Cutting Force and Surface Quality of RZ5/TiB2 Magnesium Based Metal Matrix Composite by Sobel Sensitivity Analysis Method, J. Modares Mech. Eng. 23 (2023), 63-67. https://doi.org/10.22034/mme.23.10.63.

[21] Y.I. Shehu, A. Ruiz-Garcia, V. Palade, A. James, Sokoto Coventry Fingerprint Dataset, arXiv: 1807.10609, 2018. https://doi.org/10.48550/arXiv.1807.10609.

[22] A.C.J.W. Janssens, F.K. Martens, Reflection on Modern Methods: Revisiting the Area Under the ROC Curve, Int. J. Epidemiol. 49 (2020), 1397-1403. https://doi.org/10.1093/ije/dyz274.

[23] K. Namdar, M.A. Haider, F. Khalvati, A Modified AUC for Training Convolutional Neural Networks: Taking Confidence into Account, Front. Artif. Intell. 4 (2021), 582928. https://doi.org/10.3389/frai.2021.582928.

[24] J. Liu, Y. Xu, T-Friedman Test: A New Statistical Test for Multiple Comparison with an Adjustable

Conservativeness Measure, Int. J. Comput. Intell. Syst. 15 (2022), 29. https://doi.org/10.1007/s44196-022-00083-8.

[25] Q. Lin, T. Yang, Y. Yao, Large-Scale Optimization of Partial AUC in a Range of False Positive Rates, in: Advances in Neural Information Processing Systems 35, Neural Information Processing Systems Foundation, Inc. (NeurIPS), San Diego, California, USA, 2022, pp. 31239-31253. https://doi.org/10.52202/068431-2265.

[26] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, et al., Array Programming with NumPy, Nature 585 (2020), 357-362. https://doi.org/10.1038/s41586-020-2649-2.

[27] Nation Aviation University, 1 L Guzara, Kiev, Ukraine, P. Prystavka, O. Cholyshkina, Pyramid Image and Resize Based on Spline Model, Int. J. Image Graph. Signal Process. 14 (2022), 1-14. https://doi.org/10.5815/ijigsp.2022.01.01.

[28] S. Kumar, R.L. Velusamy, Kernel Approach for Similarity Measure in Latent Fingerprint Recognition, in: 2016 International Conference on Emerging Trends in Electrical Electronics and Sustainable Energy Systems (ICETEESES), IEEE, 2016, pp. 368-373. https://doi.org/10.1109/ICETEESES.2016.7581411.

[29] R.H. Bakhtiar, Y. Dani, D. Suandi, Machine Learning Based Classification of Rice Leaf Diseases: A Comparative Study, Commun. Math. Biol. Neurosci. 2025 (2025), 43. https://doi.org/10.28919/cmbn/9198.