Eur. J. Stat. 6 (2026) 12
doi: 10.28924/ada/stat.6.12

Performance Analysis of an Adapted ResNet-50 Model for Fingerprint Recognition on Synthetically Modified Images

Natasya Anjelita Sugianto, Felivia Kusnadi, Robyn Irawan*

Center for Mathematics and Society, Faculty of Science
Parahyangan Catholic University, Jl.Ciumbuleuit 94, Bandung 40141, West Java, Indonesia
natasyaanjelita@gmail.com, felivia@unpar.ac.id, robynirawan.tjia@unpar.ac.id
**Correspondence: robynirawan.tjia@unpar.ac.id*

Abstract. Fingerprint recognition systems are widely deployed in security and forensic applications, yet their ability to identify individuals whose fingerprints have been intentionally or naturally altered remains a practical concern. This paper evaluates the performance of a ResNet-50 model adapted for fingerprint identity classification using the Sokoto Coventry Fingerprint (SOCOFing) dataset, which provides 6,000 original images from 600 individuals alongside 49,270 synthetically modified images generated via the STRANGE toolbox. The model was trained exclusively on original fingerprint images, reflecting the realistic deployment scenario where operational databases contain only unaltered enrollment records. Six validation sets were constructed: three organized by modification difficulty level (easy, medium, hard) and three by modification type (obliteration, Z-cut, central rotation). Training used a transfer learning approach with full fine-tuning, the Adam optimizer, and Sparse Categorical Cross-Entropy loss over configurations of 20 and 50 training epochs. Results show that validation accuracy at 50 epochs decreased from 94.98% for easy modifications to 73.88% for hard modifications, confirming a consistent relationship between alteration severity and classification difficulty. Among modification types, obliteration and central rotation produced stable training dynamics and accuracy above 91% at 50 epochs, while Z-cut modifications caused repeated episodes of validation accuracy collapse when training extended beyond 20 epochs. At 20 epochs, Z-cut validation accuracy reached 75.13%, the highest among all modification types at that epoch count, indicating that the model generalizes quickly to Z-cut images but subsequently overfits. These results demonstrate that ResNet-50 trained only on unmodified fingerprints retains useful recognition ability across most alteration scenarios, while identifying Z-cut as a source of generalization instability that warrants targeted mitigation strategies such as early stopping and regularization.

1. Introduction

Fingerprint recognition has been actively studied for over five decades and remains one of the most reliable methods for automatic biometric identification [2–4]. The core operating

Received: 7 May 2026.

Key words and phrases. fingerprint recognition; residual neural networks; convolutional neural networks; transfer learning; biometric authentication; fingerprint modification; deep learning.

principle is to compare a query fingerprint image against records in a reference database to confirm or determine a person's identity [5, 12]. Applications span law enforcement for suspect identification and forensic evidence analysis, consumer electronics for device authentication and payment authorization, and border control and physical access management [3,4]. The continued use of fingerprints in all these settings reflects a key biological advantage: ridge patterns are unique to each individual, stable over a lifetime under normal conditions, and present from fetal development onward, making them a robust and persistent identifier [2, 3].

Despite these favorable properties, fingerprint recognition systems encounter a category of challenges that is often underrepresented in controlled laboratory studies. Ridge patterns can deteriorate naturally due to skin conditions such as eczema, psoriasis, or sustained physical labor that gradually erodes surface texture [1]. A more serious concern arises when individuals deliberately alter their fingerprints to evade identification, a practice documented in criminal investigations and studied systematically by researchers [1, 7, 37]. Yoon, Feng, and Jain [1] classified intentional alterations into three main types: *obliteration*, which destroys or obscures ridge patterns through burning, abrasion, or corrosive chemicals; *Z-cut*, a surgical distortion where the fingertip skin is cut horizontally and the two halves are reattached with an offset or inverted orientation; and *imitation*, which involves grafting skin from another body region onto the fingertip to replace the original topology entirely. All three types can substantially disrupt the ridge features that automated recognition systems depend on, reducing identification reliability even when matched against a clean reference image [1].

A fundamental difficulty that arises from this situation is the mismatch between how fingerprint databases are built and how fingerprints may appear during operation. Forensic and security databases are typically constructed from original enrollment images collected under cooperative conditions, with no modified records available [8]. It is not feasible to anticipate every modification a subject might later apply, so the recognition model must generalize from clean training data to altered test images without any exposure to such alterations during training. Understanding how model performance degrades as a function of modification type and severity is therefore essential for setting realistic operational expectations and for designing systems that remain reliable when confronted with deliberately altered fingerprints [12, 37].

To situate this problem within the broader research context, it helps to trace how fingerprint recognition methods have progressed over time. Early automated systems relied on rule-based approaches using fuzzy logic and genetic algorithms, which required hand-crafted feature specifications and proved difficult to adapt to unseen fingerprint conditions [16]. Neural network-based methods offered a first step away from manual feature engineering, as they can learn representations directly from data [14, 16, 17]. The broader rise of deep learning [13, 14] then transformed the field significantly: by stacking many layers of nonlinear operations, deep networks learn hierarchical representations that progress from local ridge orientations in early layers to complex identity-relevant combinations in deeper layers [15, 18].

Convolutional neural networks (CNNs) proved especially well-suited for image analysis because they exploit spatial structure through shared-weight convolutional filters, introducing translational invariance while keeping parameter counts manageable [14, 15]. Krizhevsky, Sutskever, and Hinton [19] demonstrated with AlexNet that a deep CNN could dramatically outperform classical image classifiers on the large-scale ImageNet benchmark, and the wave of architectural innovation that followed is surveyed comprehensively by Gu et al. [26]. A significant obstacle that emerged as networks grew deeper was the accuracy degradation problem: adding more layers counterintuitively increased training error because of vanishing or exploding gradients during backpropagation [20]. He, Zhang, Ren, and Sun [20] resolved this with deep residual learning, which inserts shortcut connections that add the block input directly to its output and allow gradients to bypass individual layers, making it feasible to train networks with 50, 100, or more layers without performance degradation. The ResNet family achieved leading results on ImageNet, and subsequent architectures including DenseNet [22], EfficientNet [23], Inception-ResNet [24], and MobileNetV2 [25] extended these ideas further, though ResNet-50 has remained one of the most widely adopted backbones for transfer learning across image domains.

Within the fingerprint domain specifically, deep CNNs have been applied to a range of related problems with encouraging results. Nogueira, Lotufo, and Machado [9] applied CNNs to fingerprint liveness detection and achieved large accuracy gains over hand-crafted texture feature methods, showing that deep feature learning generalizes well to the visual characteristics of fingerprint images. Cao and Jain [10] tackled automated latent fingerprint recognition using deep representations, and Chugh, Cao, and Jain [11] developed a minutiae-centered deep network for presentation attack detection, together demonstrating that end-to-end deep learning can handle fine-grained fingerprint analysis tasks that previously required expert-designed pipelines. Sundararajan and Woodard [12] surveyed deep learning across multiple biometric modalities and found that CNN methods consistently outperform hand-crafted feature approaches, while Nanni, Ghidoni, and Brahnam [27] reinforced this with direct empirical comparisons on classification benchmarks. Priesnitz et al. [5] highlighted generalization to varying acquisition conditions as a persistent challenge in touchless fingerprint recognition, pointing to the same category of distribution shift problem that modified fingerprints present.

The specific problem of modified fingerprint recognition on a controlled benchmark was opened by Shehu et al. [7], who introduced the SOCOFing dataset and trained a CNN to distinguish modification types rather than recognize individual identities, a useful diagnostic task but one that left the harder recognition problem unaddressed. Mahmoud et al. [8] moved toward identity recognition on SOCOFing using a ResNet model, but their setup subsampled only 100 of the 600 available individuals and included modified images in training, making the problem considerably simpler and less representative of operational conditions where modified images would not be available during enrollment. Alsharman et al. [3] proposed a bifurcation minutiae-based mathematical model combined with neural network feature selection for fingerprint security, an interpretable approach that works well for intact fingerprints but is more likely to struggle when modifications disrupt the ridge continuity that minutiae extraction depends on. Together, these studies leave open the question of how well a purely original-image-trained deep model can generalize to all three modification types and across multiple difficulty levels using the full set of 600 individuals.

The present study addresses this gap with a design that is closer to real deployment conditions. First, the model is trained exclusively on the 6,000 original fingerprint images from all 600 individuals in SOCOFing, without any modified images in training, reflecting the typical state of an existing operational database where only enrollment-time images are stored. Second, including all 600 individuals rather than a subset makes this a 600-way classification problem, substantially more challenging and more representative of the scale at which real biometric systems operate. Third, model performance is evaluated separately across modification difficulty levels (easy, medium, hard) and modification types (obliteration, Z-cut, central rotation), providing a systematic quantitative characterization of which dimensions of alteration most affect recognition reliability and how training duration interacts with modification properties.

2. Data and Methods

2.1. Dataset. The Sokoto Coventry Fingerprint (SOCOFing) dataset [6] was used for all experiments.[1] It contains 6,000 original fingerprint images from 600 African subjects, collected using Hamster Plus (HSDU03P^{TM}) or SecuGen SDU03P^{TM} optical sensors. Each subject contributed images of all ten fingers, yielding ten images per individual. All original images have a resolution of $96 \times 103 \times 3$ pixels in BMP format. Metadata encoded in the filename includes the subject index (1 to 600), gender, hand (left or right), and finger name. The subject index is the class label for recognition.

The dataset also includes 49,270 synthetically modified images produced by the STRANGE toolbox [7]. STRANGE generates three modification types: obliteration (simulating abrasion or burning), Z-cut (simulating surgical incision and reattachment), and central rotation (simulating surgical repositioning of the central region). Each is applied at three difficulty levels named easy, medium, and hard, corresponding to increasing proportions of the fingerprint area altered. Modifications are centered on the most discriminative region of the fingerprint [1]. Figure 1 illustrates examples of each modification type at all three difficulty levels.

For this study, the 6,000 original images formed the training set. Six validation sets were constructed from modified images (Table 1): three organized by difficulty level (sets 1, 2, 3) and three by modification type (sets 4, 5, 6).

[1]The dataset is publicly available at `https://www.kaggle.com/datasets/ruizgara/socofing`.

TABLE 1. Composition of the six validation sets used in this study.

Set	Modification Type	Difficulty Level	Images
1	All types	Easy	17,931
2	All types	Medium	17,067
3	All types	Hard	14,272
4	Obliteration	All	17,854
5	Z-cut	All	14,977
6	Central rotation	All	16,439

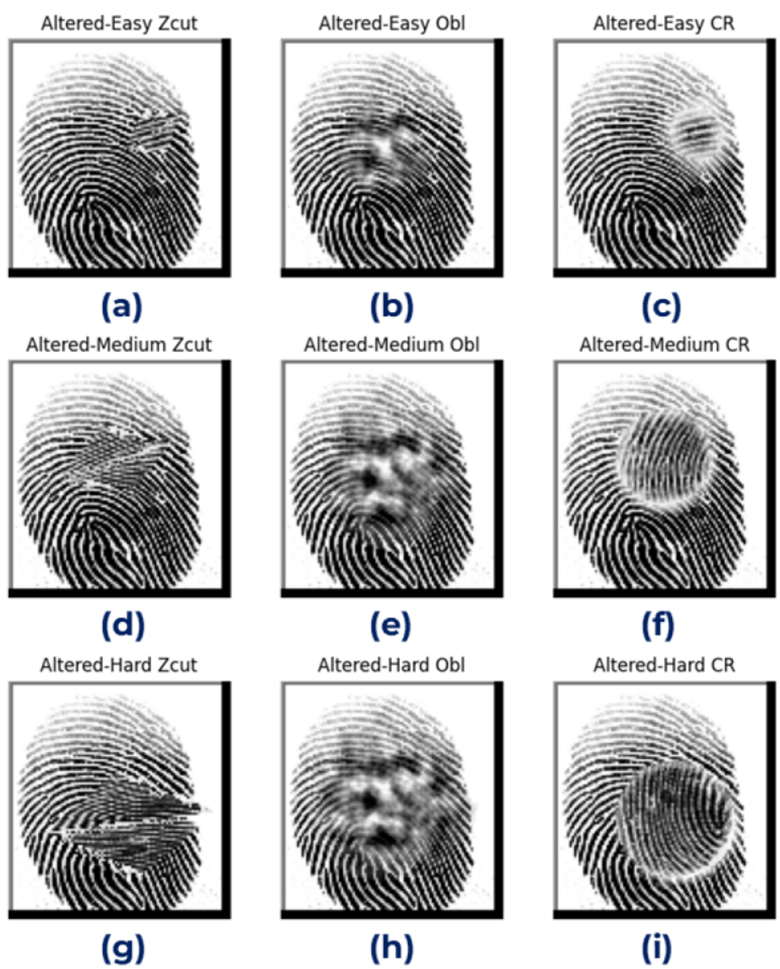

FIGURE 1. Examples of the three modification types applied to the same original fingerprint at three difficulty levels (SOCOFing dataset). Columns: Z-cut (left), obliteration (center), central rotation (right). Rows: easy (top), medium (middle), hard (bottom).

2.2. **Data Preprocessing.** All images were normalized by dividing pixel intensities by 255, rescaling values from [0, 255] to [0, 1]. The original dimensions ($96 \times 103 \times 3$) are smaller than the $224 \times 224 \times 3$ input required by ResNet-50. Zero-padding was applied to both spatial dimensions to reach the required size without resampling, so all spatial information in the original images is preserved. Shorten and Khoshgoftaar [35] provide a broad survey of preprocessing and augmentation strategies for deep learning; augmentation such as random rotation or elastic deformation was deliberately excluded from this study to test the baseline generalization capacity of ResNet-50 on clean training data, leaving augmentation as an avenue for future improvement.

2.3. **Model Architecture.** The recognition model is based on ResNet-50 [20], pretrained on the ImageNet dataset with 1.2 million images across 1,000 object categories [19]. ResNet-50 consists of 50 layers organized into four stages of residual blocks, as illustrated in Figure 2. Two block types are used: identity blocks, where the shortcut connection passes the input unchanged, and convolutional blocks, where a 1×1 convolution in the shortcut adjusts channel dimensions at stage boundaries [20]. The internal structure of both block types is shown in Figure 3.

The defining property of residual learning is the shortcut connection:

$$H(\mathbf{x}_l, w_l) \;:=\; F(\mathbf{x}_l, w_l) + \mathbf{x}_l, \tag{1}$$

where $\mathbf{x}_l$ is the block input, w_l denotes the block parameters, and $F(\mathbf{x}_l, w_l)$ is the output of the convolutional sub-path [20]. The network learns the residual F rather than the full mapping H, which is easier to optimize when H is close to an identity. This formulation also stabilizes gradient flow: following He et al. [21], the gradient of the loss $\mathcal{L}$ with respect to block-l input is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_l} \;=\; \frac{\partial \mathcal{L}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1} F(\mathbf{x}_i, w_i)\right), \tag{2}$$

where L denotes a later block. The first term on the right is transmitted directly through the shortcut, ensuring gradient magnitude does not vanish even in a very deep network [21].

Batch normalization is applied after each convolutional operation to stabilize training and reduce sensitivity to weight initialization [15]. The intermediate activation function is ReLU:

$$f(x) = \max(0, x). \tag{3}$$

For this study, the original 1,000-neuron output layer was replaced with a 600-neuron dense layer corresponding to the 600 subjects in SOCOFing. The output activation is Softmax:

$$\sigma(z_j) = \frac{e^{z_j}}{\sum_{m=1}^{600} e^{z_m}}, \qquad j = 1, \ldots, 600, \tag{4}$$

where z_j is the pre-activation value of output neuron j. The full adapted model contains 83,746,392 trainable parameters (approximately 319.47 MB).

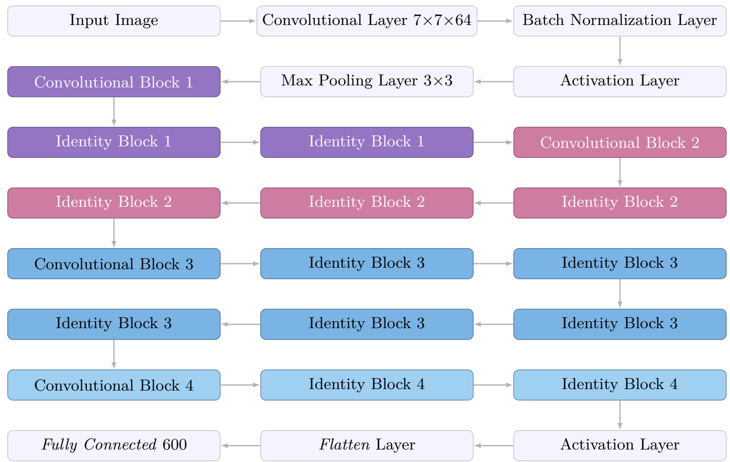

FIGURE 2. Overall layer sequence of the adapted ResNet-50 for 600-class fingerprint recognition. The original 1,000-neuron output layer is replaced with a 600-neuron Softmax output layer to match the 600-subject classification task. Colour coding: purple = Stage 1, rose = Stage 2, sky-blue = Stage 3, light-blue = Stage 4.

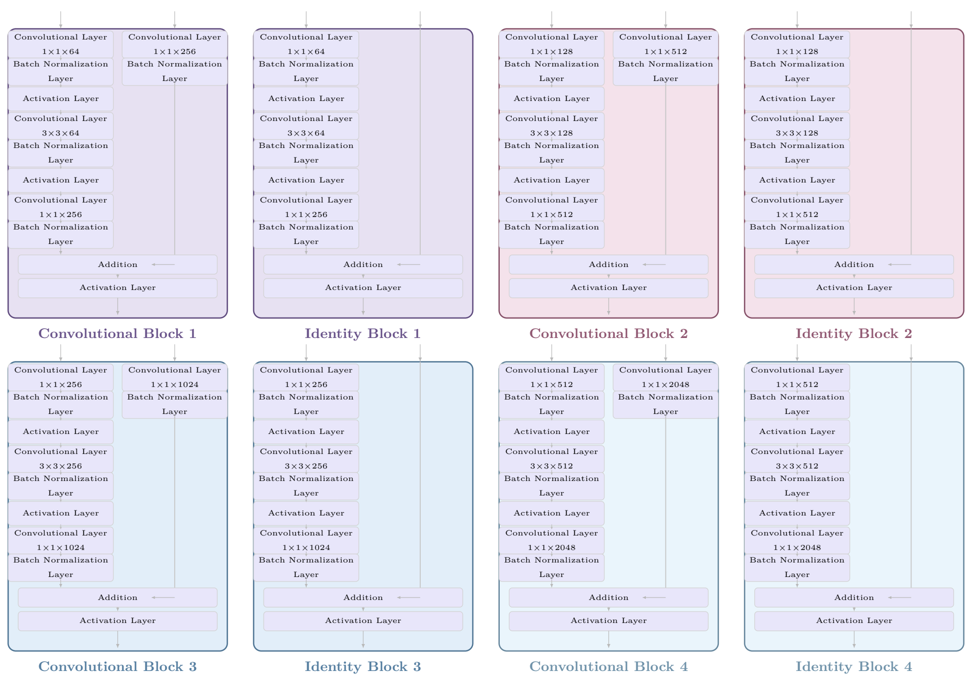

FIGURE 3. Internal structure of each residual block type across all four stages. Left column of each block: main convolutional path (three Convolutional + Batch Normalization + Activation sub-blocks). Right column: shortcut path, which contains a Convolutional + Batch Normalization layer for convolutional blocks and a direct pass-through line for identity blocks. Both paths are summed at the Addition node, followed by a final Activation Layer.

2.4. **Training Procedure.** Training follows the fine-tuning variant of transfer learning, in which all parameters of the pretrained network are updated during training [28, 29, 38]. Fine-tuning was preferred over feature extraction because fingerprint images differ substantially from ImageNet's natural image domain; freezing pretrained layers is known to be suboptimal when the domain gap is large, and Kornblith, Shlens, and Le [30] showed that better ImageNet accuracy generally implies better transfer to target tasks under fine-tuning conditions.

The training objective is the Sparse Categorical Cross-Entropy loss [14]:

$$J(w, b) = -\frac{1}{N} \sum_{i=1}^{N} \log \sigma_{r_i}, \tag{5}$$

where N is the number of training samples, r_i is the true class label for sample i, and σ_{r_i} is the predicted probability for that class. This is equivalent to Categorical Cross-Entropy but accepts integer labels directly rather than one-hot vectors, reducing memory requirements for large numbers of classes [14].

Parameter updates used the Adam optimizer [31], which belongs to the broader family of adaptive gradient methods [32]. At iteration t with gradient g_t, Adam computes biased first- and second-moment estimates

$$m_t = \beta_1 \, m_{t-1} + (1 - \beta_1) \, g_t, \qquad v_t = \beta_2 \, v_{t-1} + (1 - \beta_2) \, g_t^2, \tag{6}$$

applies bias corrections $\hat{m}_t = m_t/(1 - \beta_1^t)$ and $\hat{v}_t = v_t/(1 - \beta_2^t)$, and updates parameters as

$$\theta_t \;\leftarrow\; \theta_{t-1} - \alpha \, \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{7}$$

with learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ [31]. Adam adapts an effective learning rate for each parameter based on gradient history, making it well-suited for fine-tuning the large and heterogeneous parameter set of ResNet-50 [32].

Two training configurations were evaluated: 50 epochs and 20 epochs. The shorter run was added because signs of overfitting appeared in validation curves after epoch 20 in some configurations [34]. A mini-batch size of 32 was used throughout. Model accuracy is defined as the proportion of correctly classified images over total images evaluated. Experiments were conducted on an NVIDIA P100 GPU on the Kaggle platform; training one 50-epoch configuration required approximately 50 to 60 minutes [39].

3. Results

3.1. **Performance by Modification Difficulty Level: 50 Epochs.** Figure 4 shows training and validation accuracy and loss for validation sets 1 (easy), 2 (medium), and 3 (hard) at 50 training epochs.

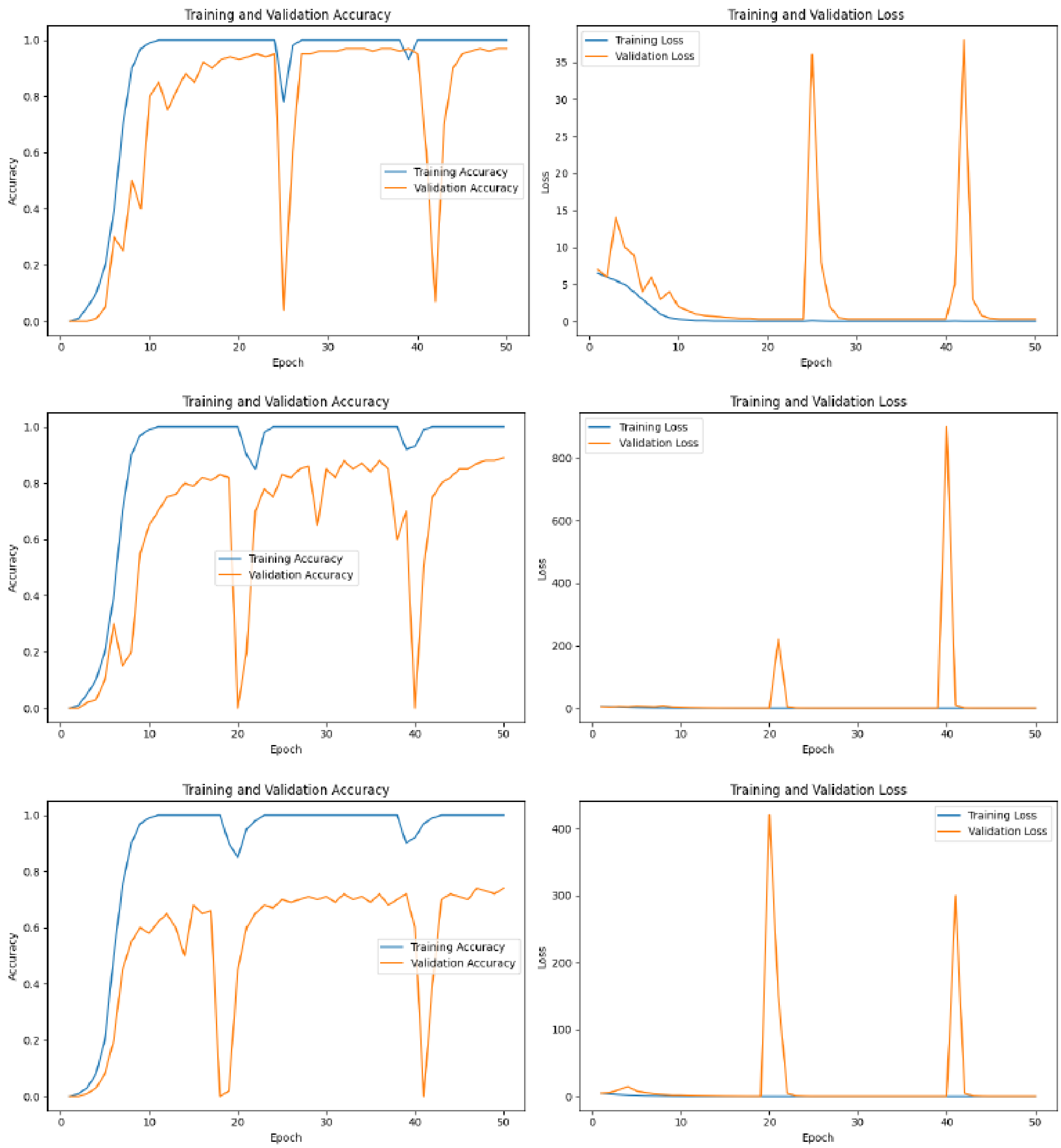

FIGURE 4. Training (blue) and validation (orange) accuracy (left column) and loss (right column) over 50 epochs for the three difficulty-level validation sets. Top row: easy (set 1). Middle row: medium (set 2). Bottom row: hard (set 3).

For set 1 (easy), training accuracy approached 100% by epoch 50. Validation accuracy increased in parallel and reached 94.98% at the final epoch, with a validation loss of 0.2359. The validation curve showed some fluctuation, expected because validation images differ structurally from the clean training images.

For set 2 (medium), the training trajectory was similar, but the gap between training and validation accuracy was larger. Final validation accuracy was 86.93% with a loss of 0.5831.

For set 3 (hard), the gap widened further. Final validation accuracy was 73.88% with a loss of 1.5202.

Across all three difficulty-level sets, abrupt drops in validation accuracy accompanied by sharp spikes in validation loss appeared at several epochs after epoch 20. These instabilities are consistent with overfitting: the model memorizes training-specific patterns that do not transfer to modified images [14, 34]. Importantly, all three of these validation sets contain Z-cut images alongside the other modification types.

3.2. **Performance by Modification Difficulty Level: 20 Epochs.** To assess whether the instabilities after epoch 20 reflect overfitting, training was repeated with a maximum of 20 epochs. Figure 5 shows the resulting curves.

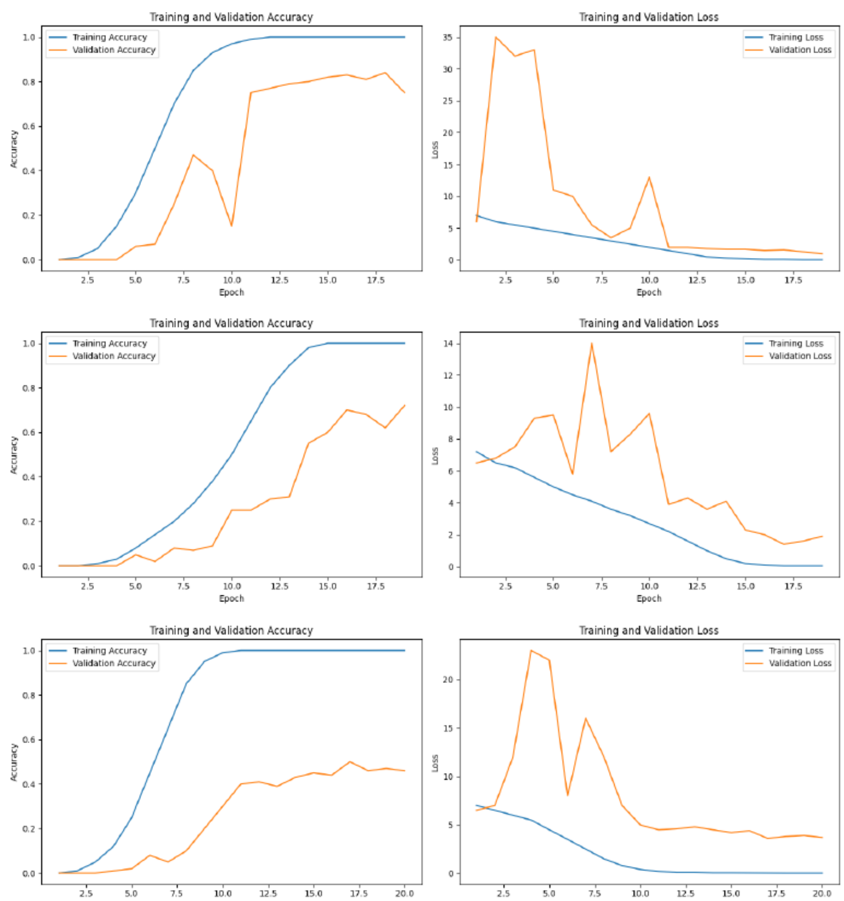

FIGURE 5. Training (blue) and validation (orange) accuracy (left) and loss (right) over 20 epochs for difficulty-level validation sets 1 (top), 2 (middle), and 3 (bottom). Compared with Figure 4, curves are substantially more stable with no abrupt accuracy collapses.

At 20 epochs, validation curves were substantially more stable with no abrupt drops. For set 1, validation accuracy reached 84.50% with a loss of 0.9627. For set 2, accuracy was 71.26% with a loss of 1.3904. For set 3, accuracy dropped to 46.14% with a loss of 3.6090.

The very low accuracy for hard-level modifications at 20 epochs indicates that the model needs more training to handle severely altered images. For easy and medium modifications, 20 epochs produce stable and reasonable accuracy, while 50 epochs are recommended for hard-level modifications.

3.3. **Performance by Modification Type: 50 Epochs.** Figure 6 presents training and validation curves for sets 4 (obliteration), 5 (Z-cut), and 6 (central rotation) at 50 epochs.

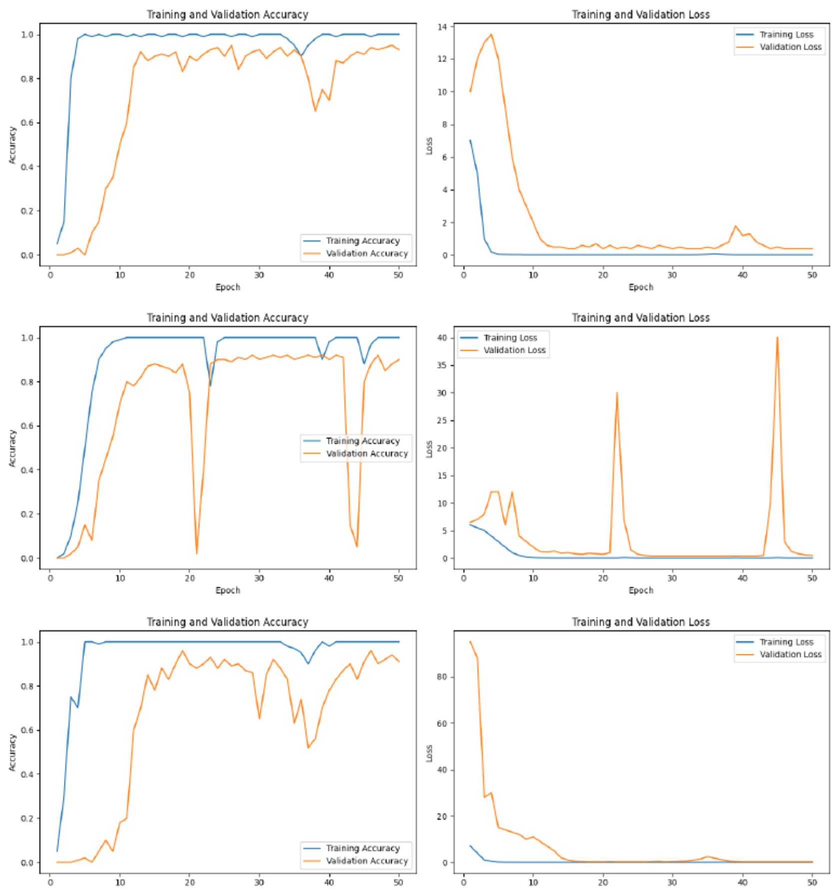

FIGURE 6. Training (blue) and validation (orange) accuracy (left) and loss (right) over 50 epochs for the three modification-type validation sets. Top row: obliteration (set 4). Middle row: Z-cut (set 5). Bottom row: central rotation (set 6).

For obliteration (set 4), final validation accuracy was 93.33% with a loss of 0.3882. Training and validation curves showed no extreme instabilities, with validation loss decreasing gradually throughout training.

For central rotation (set 6), validation accuracy was 91.30% with a loss of 0.3341. Curves were stable with no abrupt changes, and validation loss remained low and smooth across all epochs.

For Z-cut (set 5), final validation accuracy was 87.68% with a loss of 0.6337. However, the validation curves showed repeated episodes of abrupt accuracy drops followed by recovery at multiple epochs. This pattern matches the instabilities seen in difficulty-level sets 1, 2, and 3, which all contain Z-cut images. The specificity of this behavior to Z-cut-containing sets provides strong evidence that Z-cut modifications create a unique and persistent generalization challenge.

3.4. **Performance by Modification Type: 20 Epochs.** At 20 epochs, the behavior of Z-cut relative to the other types reversed. Z-cut (set 5) achieved the highest validation accuracy among the three types: 75.13% with a loss of 1.4558, and training curves showed no instability. Obliteration (set 4) reached 64.56% with a loss of 2.5190, and central rotation (set 6) reached 58.96% with a loss of 2.1954.

This reversal shows that the model reaches a good generalizing state for Z-cut images within 20 epochs but then overfits as training continues. For obliteration and central rotation, longer training consistently improves validation accuracy without introducing instability.

3.5. **Summary of Results.** Tables 2 and 3 summarize all performance results at the final training epoch. Bold values indicate the best accuracy or lowest loss within each epoch configuration; italic values indicate the worst. Training time per configuration is also reported.

Table 2. Model performance on difficulty-level validation sets (sets 1, 2, 3) at 20 and 50 training epochs. Bold: best accuracy/lowest loss per epoch group; italic: worst.

Epochs	Set	Level	Accuracy	Loss	Train time
20	1	Easy	**0.8450**	**0.9627**	00:20:43
	2	Medium	0.7126	1.3904	00:20:33
	3	Hard	*0.4614*	*3.6090*	00:19:18
50	1	Easy	**0.9498**	**0.2359**	00:59:54
	2	Medium	0.8693	0.5831	00:58:22
	3	Hard	*0.7388*	*1.5202*	00:54:00

Table 3. Model performance on modification-type validation sets (sets 4, 5, 6) at 20 and 50 training epochs. Bold: best accuracy/lowest loss per epoch group; italic: worst.

Epochs	Set	Type	Accuracy	Loss	Train time
20	4	Obliteration	0.6456	*2.5190*	00:21:32
	5	Z-cut	**0.7513**	1.4558	00:18:39
	6	Central rotation	*0.5896*	2.1954	00:20:09
50	4	Obliteration	**0.9333**	0.3882	00:49:30
	5	Z-cut	0.8768	0.6337	00:46:16
	6	Central rotation	0.9130	**0.3341**	00:46:51

4. Discussion

4.1. **Effect of Modification Difficulty on Recognition Performance.** The results show a clear and consistent relationship between modification severity and classification accuracy that holds across both 20-epoch and 50-epoch training. At 50 epochs, accuracy fell from 94.98% for easy to 73.88% for hard modifications; at 20 epochs, from 84.50% to 46.14%. This pattern is expected because harder modifications alter a larger proportion of the central fingerprint region, which contains the most discriminative ridge features [1, 6]. When more of this region is disrupted, fewer intact ridge structures are available for the model to use when making a prediction.

The fact that the model retained accuracy above 73% at hard difficulty despite never seeing modified images in training is a meaningful result. Studies that trained on mixed datasets including modified images report higher accuracy on altered test images [8], but this advantage comes from direct exposure to the test distribution during training, which is unrealistic in most operational settings. The present model must rely solely on features learned from clean images and the pretrained ResNet-50 backbone. Transfer learning from ImageNet initializes the network with filters capable of detecting edges, textures, and local patterns [20, 30], and many of these low-level features remain informative even in heavily modified fingerprint images. Kornblith, Shlens, and Le [30] showed that models with better ImageNet accuracy consistently achieve better transfer to downstream image tasks, providing theoretical grounding for why a strong pretrained backbone helps in this context. Deeper layers of ResNet-50 combine low-level features into more abstract identity-relevant representations [15, 18], and these representations appear to retain enough fingerprint identity information to yield non-trivial accuracy even at hard difficulty.

The gap between 20-epoch and 50-epoch accuracy is largest for hard modifications (46.14% vs. 73.88%), indicating that severely altered images require more training iterations to adapt network weights sufficiently from the ImageNet initialization. This is consistent with the transfer learning literature: the number of fine-tuning steps needed increases with the domain gap between source and target distributions [28, 29, 38]. For easy and medium modifications, the domain gap is smaller because larger portions of the fingerprint remain intact, so 20 epochs suffice to reach a stable and reasonably accurate model.

From an applied standpoint, these results suggest that a system intended to handle only easy or medium modifications could be trained with fewer epochs and shorter compute time, while a system designed for harder modification scenarios would require longer training or a more diverse training set.

4.2. **Effect of Modification Type on Recognition Performance.** Among the three modification types, obliteration and central rotation yielded high and consistent validation accuracy at 50 epochs (93.33% and 91.30% respectively), with no abrupt instabilities in training curves. The similar performance of these two types is notable given their different mechanisms: obliteration disrupts ridge texture locally through burning or abrasion, while central rotation surgically repositions ridge segments. Both modifications, however, target the central fingerprint region while leaving peripheral areas intact, and the peripheral ridge features likely provide sufficient information for the model to identify the subject [1, 3]. The result for obliteration also aligns with the finding by Nogueira et al. [9] that CNNs are robust to local texture disruption, as deep convolutional features are not entirely dependent on any single region of the input image [15, 26].

Z-cut presents a qualitatively different challenge. The Z-cut procedure cuts the fingerprint along a horizontal axis and reattaches the halves with an offset, creating a ridge-break pattern across the central axis that does not appear anywhere in the original training images [1, 7]. This creates a structural distribution shift that the model finds difficult to handle consistently. The repeated accuracy collapses during 50-epoch training of Z-cut-containing validation sets likely reflect oscillating gradient updates: the model temporarily moves toward representations that handle Z-cut images well, then overshoots into parameter regions that misclassify them. Biggio and Roli [37] discuss analogous instabilities in models trained on distributions that diverge from the test distribution under adversarial perturbations, noting that gradient-based optimization can oscillate between locally good and poor solutions. The fingerprint Z-cut scenario shares this property because Z-cut images represent a distribution that the training set does not sample from at all [14, 34].

The reversal of Z-cut performance between 20 and 50 epochs is an important diagnostic. At 20 epochs, Z-cut achieved the highest accuracy among all modification types (75.13%), and training was stable. At 50 epochs, Z-cut showed the worst stability despite not necessarily having the lowest final accuracy among types. This pattern indicates that the model reaches a good generalizing parameter region for Z-cut images within the first 20 epochs, but continued training drives the model toward overfitting the original training distribution in ways that specifically hurt Z-cut generalization. For Z-cut-sensitive applications, early stopping based on validation loss [33, 34], combined with dropout regularization [14, 15], is recommended.

4.3. **Transfer Learning and Fine-Tuning.** The choice to fine-tune all 83.7 million parameters rather than only the output layer was motivated by the large domain gap between natural images and fingerprint images [28,29,38]. When domains differ substantially, freezing pretrained weights often performs poorly because those features do not align with target domain patterns [29]. Fine-tuning allows the network to shift its representations toward fingerprint-specific patterns. The domain gap here is particularly large because ImageNet images are full-color natural scenes, whereas fingerprint images are near-grayscale ridge-valley patterns at a scale irrelevant to natural object recognition.

A consequence of full fine-tuning on a small training set (6,000 images, 10 per class) is susceptibility to overfitting [22, 34]. The instabilities seen after epoch 20 are most plausibly explained by this mechanism. Data augmentation strategies such as random rotations, flips, or elastic distortions were not applied in this study [35, 38] but would increase effective training set size and encourage the model to learn more robust representations. Shorten and Khoshgoftaar [35] provide a comprehensive review of augmentation methods for deep learning; geometric transforms and random erasing would be particularly relevant here to simulate the partial ridge loss caused by fingerprint modifications. Dropout in the fully connected layer or L2 weight decay could also help regularize the model [14, 15].

Another dimension worth investigating is the choice of backbone architecture. DenseNet [22] uses a connectivity pattern where each layer receives feature maps from all preceding layers, which promotes feature reuse and provides implicit regularization. EfficientNet [23] scales width, depth, and resolution jointly using a compound coefficient and achieves competitive accuracy with fewer parameters. MobileNetV2 [25] and Inception-ResNet variants [24] offer further accuracy-versus-cost trade-offs. The broad survey by Alzubaidi et al. [17] covers these and many other architectures, providing a useful reference for selecting alternatives. Comparing these against ResNet-50 on SOCOFing under the same experimental conditions would clarify whether the observed performance patterns are specific to residual architectures.

4.4. **Comparison with Prior Work.** The study most directly comparable to this one is Mahmoud et al. [8], who applied a ResNet model to fingerprint identity recognition on SOCOFing. Their experimental setup used only 100 of 600 subjects and included modified images in training. Both differences make the problem substantially easier and less representative of operational conditions. Despite the more restrictive setup in the present study, accuracy above 87% was achievable for obliteration and central rotation, and above 73% for hard-level modifications.

Shehu et al. [7] addressed a complementary problem: classifying fingerprint images into four modification-type categories rather than recognizing individual identity. In a complete operational system, a modification-type classifier could be used upstream to route images to recognition models optimized for each modification type, likely improving overall system accuracy.

Nanni, Ghidoni, and Brahnam [27] demonstrated that deep learned features outperform hand-crafted descriptors in computer vision classification, and this advantage is expected to be particularly pronounced for modified fingerprints where hand-crafted minutiae extractors depend on intact ridge structures that modifications specifically disrupt. Alsharman et al. [3] used a minutiae-based neural network approach that works well for intact fingerprints but may struggle when ridge continuity is broken by Z-cut or hard-level obliteration. The broader biometric deep learning survey by Sundararajan and Woodard [12] supports the conclusion that end-to-end CNN methods are more robust to distribution shift than pipeline methods relying on hand-crafted features, which aligns with the results reported here.

Compared to fingerprint liveness detection work [9,11], the identity recognition task addressed here is considerably harder. Liveness detection is a binary classification problem, while identity recognition among 600 individuals requires the model to discriminate fine-grained differences between individuals in the presence of modification-induced structural changes.

4.5. **Limitations and Future Directions.** Several limitations should be noted. First, training used only 6,000 images from one dataset collected from one geographic population using specific optical sensors. Generalizability to other population groups, sensor technologies, or real (non-synthetic) modifications is not established by this study [5]. Evaluating on real modified fingerprints from forensic case databases would be an important next step.

Second, only ResNet-50 was evaluated. Comparisons with DenseNet [22], EfficientNet [23], or MobileNetV2 [25] under identical conditions are needed to determine whether the performance patterns are specific to residual architectures or hold more broadly [17,26].

Third, data augmentation was not applied, limiting effective training set size given only 10 images per individual [35]. Augmentation tailored to fingerprint images, such as elastic ridge deformation or random partial masking, could substantially improve robustness to unseen modifications [22,38].

Fourth, performance was evaluated using aggregate accuracy and loss. A per-class or per-modification-position analysis would reveal which individuals or fingerprint regions cause most errors. Emmert-Streib et al. [18] discuss interpretability techniques for deep learning that could be adapted to identify which regions of modified fingerprints most affect classification decisions.

Future work should explore: few-shot learning approaches [36] to address the small per-class training set size, where the model learns general fingerprint identity features from fewer examples; metric learning or contrastive learning [14] that cluster images of the same individual regardless of modification; cross-modification transfer, where the model is trained on images from one modification type and tested on another; and combinations of the recognition model with Shehu et al.'s [7] modification-type classifier to enable type-aware recognition.

5. Conclusion

This paper evaluated the performance of an adapted ResNet-50 model for fingerprint identity recognition when trained on clean original images and tested on six sets of synthetically modified images from the SOCOFing dataset. The study addressed the practical scenario where an operational database contains only unaltered enrollment records, making generalization to modified fingerprints the primary challenge.

Three main findings emerged. First, modification difficulty has a consistent and meaningful effect on recognition accuracy, with performance ranging from 94.98% for easy modifications to 73.88% for hard modifications at 50 training epochs. This confirms that greater alteration severity progressively reduces the model's ability to correctly identify individuals.

Second, modification type affects not only average accuracy but also training dynamics. Obliteration and central rotation produced stable training curves and high accuracy (above 91% at 50 epochs), while Z-cut modifications caused repeated validation accuracy collapses during extended training. Z-cut requires careful training management, specifically shorter training runs or targeted regularization, to avoid the overfitting instabilities that emerge after epoch 20.

Third, training duration interacts with modification type and difficulty. For Z-cut images, 20 epochs produced the best accuracy among all types and stable dynamics, while 50 epochs introduced instability. For obliteration and central rotation, 50 epochs consistently improved accuracy over 20 epochs without instability.

Overall, the results support the feasibility of deploying ResNet-50-based fingerprint recognition in settings where training data consist only of original enrollment images. The model retains useful generalization ability across most modification scenarios. Addressing Z-cut instability through regularization and early stopping, and extending robustness to hard-level modifications through data augmentation and larger or more diverse training data, are the most actionable directions for improving system reliability in forensic and security applications.

Acknowledgments: The authors thank the Center for Mathematics and Society, Faculty of Science, Universitas Katolik Parahyangan, for the support and facilities provided during this research.

Competing interests: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] S. Yoon, J. Feng, A.K. Jain, Altered Fingerprints: Analysis and Detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012), 451–464. https://doi.org/10.1109/TPAMI.2011.161.

[2] A.K. Jain, K. Nandakumar, A. Ross, 50 Years of Biometric Research: Accomplishments, Challenges, and Opportunities, Pattern Recognit. Lett. 79 (2016), 80–105. https://doi.org/10.1016/j.patrec.2015.12.013.

[3] N. Alsharman, A. Saaidah, O. Almomani, I. Jawarneh, L. Al-Qaisi, Pattern Mathematical Model for Fingerprint Security Using Bifurcation Minutiae Extraction and Neural Network Feature Selection, Secur. Commun. Netw. 2022 (2022), 4375232. https://doi.org/10.1155/2022/4375232.

[4] I. Getzi, S. Navachudar, J. Karannevar, Liveness Detection Using Fingerprint Pores, Int. J. Eng. Res. Technol. 4 (2018), 1–3.

[5] J. Priesnitz, C. Rathgeb, N. Buchmann, C. Busch, M. Margraf, An Overview of Touchless 2D Fingerprint Recognition, EURASIP J. Image Video Process. 2021 (2021), 8. https://doi.org/10.1186/s13640-021-00548-4.

[6] 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.

[7] Y.I. Shehu, A. Ruiz-Garcia, V. Palade, A. James, Detection of Fingerprint Alterations Using Deep Convolutional Neural Networks, in: V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis, (eds) Artificial Neural Networks and Machine Learning – ICANN 2018, Lecture Notes in Computer Science, Vol 11139, Springer, Cham, (2018). https://doi.org/10.1007/978-3-030-01418-6_6.

[8] A. Tarek Mahmoud, W.A. Awad, G. Behery, M. Abouhawwash, M. Masud, et al., An Automatic Deep Neural Network Model for Fingerprint Classification, Intell. Autom. Soft Comput. 36 (2023), 2007–2023. https://doi.org/10.32604/iasc.2023.031692.

[9] R.F. Nogueira, R.A. Lotufo, R.C. Machado, Fingerprint Liveness Detection Using Convolutional Neural Networks, IEEE Trans. Inf. Forensics Secur. 11 (2016), 1206–1213. https://doi.org/10.1109/TIFS.2016.2520880.

[10] K. Cao, A.K. Jain, Automated Latent Fingerprint Recognition, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2019), 788–800. https://doi.org/10.1109/TPAMI.2018.2818162.

[11] T. Chugh, K. Cao, A.K. Jain, Fingerprint Spoof Buster: Use of Minutiae-Centered Patches, IEEE Trans. Inf. Forensics Secur. 13 (2018), 2190–2202. https://doi.org/10.1109/TIFS.2018.2812193.

[12] K. Sundararajan, D.L. Woodard, Deep Learning for Biometrics, ACM Comput. Surv. 51 (2018), 1–34. https://doi.org/10.1145/3190618.

[13] Y. LeCun, Y. Bengio, G. Hinton, Deep Learning, Nature 521 (2015), 436–444. https://doi.org/10.1038/nature14539.

[14] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[15] C.C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018. https://doi.org/10.1007/978-3-319-94463-0.

[16] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, et al., State-of-the-Art in Artificial Neural Network Applications: A Survey, Heliyon 4 (2018), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938.

[17] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, et al., Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions, J. Big Data 8 (2021), 53. https://doi.org/10.1186/s40537-021-00444-8.

[18] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, M. Dehmer, An Introductory Review of Deep Learning for Prediction Models with Big Data, Front. Artif. Intell. 3 (2020), 4. https://doi.org/10.3389/frai.2020.00004.

[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Commun. ACM 60 (2017), 84–90. https://doi.org/10.1145/3065386.

[20] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[21] K. He, X. Zhang, S. Ren, J. Sun, Identity Mappings in Deep Residual Networks, in: Lecture Notes in Computer Science, Springer, Cham, 2016: pp. 630–645. https://doi.org/10.1007/978-3-319-46493-0_38.

[22] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

[23] M. Tan, Q.V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, arXiv:1905.11946, 2019. https://doi.org/10.48550/arXiv.1905.11946.

[24] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI), 2017. https://doi.org/10.1609/aaai.v31i1.11231.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.

[26] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, et al., Recent Advances in Convolutional Neural Networks, Pattern Recognit. 77 (2018), 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.

[27] L. Nanni, S. Ghidoni, S. Brahnam, Handcrafted vs. Non-handcrafted Features for Computer Vision Classification, Pattern Recognit. 71 (2017), 158–172. https://doi.org/10.1016/j.patcog.2017.05.025.

[28] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, et al., A Comprehensive Survey on Transfer Learning, in: Proceedings of the IEEE, Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 43–76. https://doi.org/10.1109/JPROC.2020.3004555.

[29] K. Weiss, T.M. Khoshgoftaar, D. Wang, A Survey of Transfer Learning, J. Big Data 3 (2016), 9. https://doi.org/10.1186/s40537-016-0043-6.

[30] S. Kornblith, J. Shlens, Q.V. Le, Do Better ImageNet Models Transfer Better?, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 2656–2666. https://doi.org/10.1109/CVPR.2019.00277.

[31] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980, 2014. https://doi.org/10.48550/arXiv.1412.6980.

[32] S. Ruder, An Overview of Gradient Descent Optimization Algorithms, arXiv:1609.04747, 2016. https://doi.org/10.48550/arXiv.1609.04747.

[33] S. Sharma, S. Sharma, A. Athaiya, Activation Functions in Neural Networks, Int. J. Eng. Appl. Sci. Technol. 04 (2020), 310–316. https://doi.org/10.33564/IJEAST.2020.v04i12.054.

[34] S. Afaq, S. Rao, Significance of Epochs on Training a Neural Network, Int. J. Sci. Technol. Res. 9 (2020), 485–488.

[35] C. Shorten, T.M. Khoshgoftaar, A Survey on Image Data Augmentation for Deep Learning, J. Big Data 6 (2019), 60. https://doi.org/10.1186/s40537-019-0197-0.

[36] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a Few Examples, ACM Comput. Surv. 53 (2020), 1–34. https://doi.org/10.1145/3386252.

[37] B. Biggio, F. Roli, Wild Patterns: Ten Years after the Rise of Adversarial Machine Learning, Pattern Recognit. 84 (2018), 317–331. https://doi.org/10.1016/j.patcog.2018.07.023.

[38] M. Ekman, Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers Using TensorFlow, Pearson, 2021.

[39] D. Gyawali, Comparative Analysis of CPU and GPU Profiling for Deep Learning Models, arXiv:2309.02521, 2023. https://doi.org/10.48550/arXiv.2309.02521.