Multimedia Tools and Applications (2026) 85:168
https://doi.org/10.1007/s11042-026-21409-0

A multibiometric system based on finger photo and palm photo

Javad Khodadoust[1] · Raúl Monroy[1] · Miguel Angel Medina-Pérez[2] · Worapan Kusakunniran[3] · Ali Mohammad Khodadoust[4]

Received: 27 September 2023 / Revised: 23 January 2026 / Accepted: 27 January 2026

Abstract

Recognition of finger photos and palm photos is an emerging area in biometrics, primarily leveraging textural attributes for authentication. Limited efforts have explored the application of deep neural networks (DNNs) for finger photo minutiae extraction, while palm photo minutiae extraction remains challenging due to issues such as blurriness and creases, which can lead to false minutiae. In this paper, we introduce a generative adversarial network (GAN) that utilizes frequency-domain patches to deblur and enhance image quality, effectively addressing both deblurring and crease removal. Furthermore, we present a deep convolutional neural network (DCNN) for minutiae extraction from enhanced patches, alongside a model for singular point (SP) detection. Additionally, we propose a score-based multibiometric system that seamlessly integrates palm and finger photos without the need for score normalization. To validate the effectiveness of our approach, we conducted extensive experiments on a database comprising 30,000 hand photos from 2,500 volunteers, as well as a smaller dataset of 2,400 hand photos from 200 volunteers for cross-database evaluation. Our results demonstrate the enhanced accuracy of our models and establish the superiority of our multibiometric system over state-of-the-art (SOTA) methods.

Keywords Multibiometrics · Palm photo · Finger photo · Deep neural network · Minutiae · Singular point

1 Introduction

In the past, identification of individuals relied on documents such as passports, identity (ID) cards, driving licenses, or personal identification numbers (PINs). These conventional methods, however, had their vulnerabilities, as they could be lost, forgotten, guessed, or even cloned [1]. As the importance of security has grown in parallel with advancements in technology, including the proliferation of electrical devices, computers, and the Internet, a contemporary and highly secure alternative known as biometric authentication has emerged. Biometric authentication harnesses an individual's unique physical or behavioral

Extended author information available on the last page of the article

Published online: 13 February 2026

traits to establish their identity. This approach offers a more robust and dependable means of identification [2].

Biometric systems are typically designed and implemented to operate with a single biometric modality. However, in pursuit of improving the accuracy of individual authentication, the concept of biometric information fusion has emerged. This fusion can be broadly categorized into two primary domains [1]: unimodal and multimodal approaches. Unimodal biometric systems are primarily focused on person recognition using a solitary source of biometric data, which undergoes various processing techniques. Within the realm of unimodal systems, there exist methods such as multi-sample techniques, which combine information from the same input data, multi-instance and multi-sensor approaches that conduct multiple data acquisitions using the same or different sensors, and multi-algorithm methods that employ diverse strategies for analysis. Conversely, multimodal biometric systems involve the acquisition and utilization of multiple biometric traits for person authentication. These systems can either employ multiple sensors or a single sensor capable of capturing multiple traits.

Unimodal systems face several inherent challenges, including issues related to intraclass variation, susceptibility to spoofing attacks, and a notable failure-to-enroll rate [3]. In contrast, multimodal systems offer several advantages over their unimodal counterparts. They are adept at overcoming the limitations of unimodal biometric systems, particularly when some biometric traits lack universality. Multimodal systems frequently achieve higher recognition accuracy, exhibit reduced sensitivity to varying environmental conditions, and prove more resistant to forgery attempts [1].

To fuse biometric data in systems, there are four primary approaches [4]: sensor-level fusion, feature-level fusion, score-level fusion, and decision-level fusion. Among these methods, score-level fusion is the most commonly used, but it presents a challenging issue. When using different biometric traits, such as face and fingerprint, the resulting scores may have different ranges. For instance, face recognition might produce scores between 0 and 100, while fingerprint recognition may yield scores within the range of 0 to 1. Furthermore, even if both sets of scores fall within the same range, the distribution of scores may differ significantly. Consequently, the normalization of scores becomes essential, representing a formidable task in multibiometric systems [5].

The choice of biometric traits depends on our goals. For instance, when the objective is to recognize an individual without their awareness of the biometric capture, employing treats such as ear, and gait are preferable to using fingerprint and finger vein. However, when the goal is to unlock a smartphone, iris, fingerprint, and palmprint recognition are suitable options. It is also worth noting that some biometric traits, such as the face, can be utilized for both purposes.

In recent times, with the widespread adoption of smartphones, they have become an integral part of individuals' daily lives. Many people use their smartphones for various purposes, such as making purchases, conducting financial transactions, accessing their social media profiles, recording videos, and taking photos. The continuous advancements in smartphone camera technology are noticeable, with many companies placing significant emphasis on camera capabilities in their product advertisements.

The simultaneous growth in both smartphone cameras and algorithms for biometric recognition has attracted the interest of researchers towards using smartphones for biometric authentication. As previously mentioned, the choice of biometric traits heavily depends on our specific goals. Two biometric traits that have garnered significant attention from researchers are fingerprints and palmprints [6]. The practice of using smartphone cameras to capture fingertip images, commonly referred to as ‘finger photos,’ has seen a significant increase [7]. Both palmprints and fingerprints share several common features, which has led to many methods and systems developed for fingerprint recognition being adapted for palmprint recognition. The most commonly shared features include minutiae, the orientation field (OF), and SPs. However, there are notable differences between palmprints and fingerprints. Palmprints tend to have a higher number of minutiae and SPs compared to fingerprints. Additionally, palmprints feature more creases, and these creases are wider than those found in fingerprints. While creases in palmprints are considered a valuable feature in non-minutiae-based palmprint recognition methods [8–10], they can also lead to the extraction of false minutiae in minutiae-based palmprint recognition methods. Estimating the correct OF in palmprints, particularly in regions with creases, presents a more challenging task. These differences, as well as the shared features, combined with the capabilities of smartphone cameras and the advancements in DCNNs, have motivated us to explore the development of a multibiometric system based on finger photos and palm photos, which are contactless fingerprints and palmprints captured by smartphone cameras.

By capturing a hand photo, we can obtain both finger photos and palm photos, both of which can be recognized based on minutiae. Consequently, we have developed a multibiometric system that does not require score normalization. However, it is essential to note that such a system necessitates an accurate method for extracting genuine minutiae from finger photos and palm photos. Additionally, detecting SPs, which are crucial for fingerprint classification and alignment during the matching process, is essential. The achievements of this study can be outlined as follows:

- We introduce a GAN model that operates on patches of finger photos and palm photos in the frequency domain. This model enhances individual patches, thereby improving the accuracy of minutiae extraction and SP detection.
- We develop a DCNN model for minutiae extraction that operates on spatial-domain patches and can accurately extract minutiae.
- We propose a DCNN model for SP detection that utilizes frequency-domain patches and ensures accurate detection of SPs.
- We introduce a score-level fusion multibiometric system that works with minutiae and does not require score normalization.
- We create a new hand photo database consisting of 30,000 hand photos contributed by 2,500 volunteers, with three photos from dry hands and three photos from wet hands for each individual, totaling twelve photos per individual. Additionally, a supplementary dataset of 2,400 hand photos was collected from 200 volunteers using a different smartphone. This supplementary dataset was specifically designed for cross-database evaluation in our experiments.

The remainder of the paper is structured as follows. In Section 2, prior studies on patch-based fingerprint enhancement in the frequency domain using deep learning techniques, as well as SP detection methods for fingerprints using deep learning techniques and minutiae extraction from contactless fingerprints, are discussed. Section 3 provides an in-depth explanation of our proposed models. Section 4 showcases the results of our experiments. Finally, in Sections 5 and 6, we respectively present our discussion and concluding remarks.

2 Previous work

This section examines prior research on patch-based fingerprint enhancement in the frequency domain utilizing deep learning techniques, alongside SP detection methods and minutiae extraction approaches for contactless fingerprints, all employing deep learning methods.

2.1 Patch-based fingerprint enhancement

Horapong et al. [11] proposed a patch-based enhancement technique that significantly improved latent fingerprint clarity, particularly in challenging cases where traditional methods fail. Their approach leverages DNNs to refine and enhance patches of the fingerprint selectively, thereby preserving intricate details while reducing noise. The DNNs are trained specifically to identify and amplify features that are often lost during the conventional enhancement processes. By focusing on small, localized areas of the fingerprint, their method effectively handles variations in quality across the entire print, leading to superior overall enhancement. This patch-based method also allows for adaptive processing, where each patch is enhanced based on its unique characteristics, further improving the accuracy and reliability of fingerprint identification systems.

Kriangkhajorn et al. [12] introduced a spectral filter predictor aimed at improving latent fingerprint restoration through frequency-domain processing. Their method utilizes the spectral representation of fingerprints, which effectively preserves crucial details such as minutiae and SPs. Unlike traditional spatial-domain techniques, which often face challenges like aliasing and the creation of artifacts, their frequency-domain approach maintains these essential features throughout the restoration process. Moreover, the integration of deep learning techniques within this frequency-domain framework enhances the quality of restoration, making it particularly suitable for latent fingerprints with intricate backgrounds and low visibility. This method surpasses conventional approaches, providing a robust solution for increasing the clarity and usefulness of latent fingerprints in forensic contexts.

2.2 SP detection in fingerprints

Qin et al. [13] introduced a technique for detecting SPs in fingerprints by utilizing fully convolutional networks (FCNs) in conjunction with a probabilistic model. This approach departs from conventional methods that primarily depend on OFs, which can be unreliable, especially in low-quality fingerprints. Instead, their method segments fingerprints into overlapping blocks, recasting the detection task as a classification problem through the use of convolutional neural networks (CNNs). By converting these CNNs into FCNs, the approach facilitates efficient and dense prediction of SPs across multiple scales.

Liu et al. [14] introduced an approach for detecting SPs in fingerprints by leveraging Faster-RCNN, a DCNN architecture commonly used in object detection. Their method incorporates an orientation constraint to improve the accuracy of identifying SPs, which are crucial for fingerprint recognition and alignment. Unlike traditional techniques that depend on OF and are often prone to noise, their approach directly processes raw fingerprints, bypassing the need for complex preprocessing.

Chen et al. [15] introduced a method for detecting SPs in fingerprints by framing the task as a semantic segmentation problem. They developed a specialized CNN called SinNet, which is specifically designed to accurately and efficiently segment the small regions containing SPs. Unlike traditional approaches that rely heavily on precise OF and extensive preprocessing, their method bypasses these requirements. SinNet's architecture features a symmetrical encoder-decoder network, enhanced with inception modules, skip connections, and batch normalization, which contributes to its effectiveness.

Pang et al. [16] introduced a technique for fingerprint SP detection that merges classical digital image processing methods with deep learning strategies. Their approach specifically targets the difficulties associated with identifying SPs in low-quality fingerprints by employing a hybrid model that combines the angle matching index (AMI) with an enhanced convergence index filter. This technique is designed to enhance the precision and efficiency of SP detection while minimizing reliance on the fingerprint's OF.

2.3 Minutiae extraction from contactless fingerprints

Tan and Kumar [17] introduced a DNN-driven method for extracting minutiae from contactless fingerprints and compensating for variations in pose, which distinguishes it from traditional techniques that rely heavily on image enhancement and are prone to errors from false minutiae. Their approach includes a three-step pose compensation process: estimating the view angle based on the core point's position, using an ellipsoid model to simulate and adjust for different finger poses, and aligning intersection areas across various view angles. In a separate development, these authors later proposed a minutiae attention network [18], which utilizes a dual-branch architecture to improve the precision of matching contactless and contact-based fingerprints. This network includes a global-net branch for overall feature extraction and a minutiae attention branch that focuses on local minutiae regions through an attention mechanism, effectively handling distortions common in contactless fingerprints. Additionally, a reciprocal distance loss function is employed to penalize mismatches, increasing the reliability of minutiae extraction across different fingerprint modalities.

Zhang et al. [19] introduced a minutiae extraction technique that combines ridge orientation analysis with frequency estimation to effectively identify minutiae, even in challenging, low-quality fingerprints. Their method employs adaptive filtering to strengthen ridge patterns, followed by a sophisticated detection process for bifurcation and termination points to reduce false minutiae. This approach greatly enhances the dependability of minutiae extraction, particularly in situations where conventional methods falter due to noise and image distortion.

Cotrim and Pedrini [20] proposed a method for minutiae extraction using a multiscale approach within DCNNs. Their approach emphasized the importance of capturing fine-grained details at various scales, which is crucial for accurate minutiae detection in contactless fingerprint images. By leveraging a hierarchical feature extraction process, their model was able to discern intricate patterns within fingerprint ridge structures, improving the reliability of minutiae localization. In a subsequent work [21], these authors introduced an advanced minutiae extraction method that utilizes a residual squeeze-and-excitation

U-shaped network (RSU-Net) to further enhance extraction accuracy from contactless fingerprints. This approach integrates the squeeze-and-excitation mechanism with a U-Net architecture, enabling the network to effectively capture intricate fingerprint features while preserving the spatial hierarchies essential for minutiae detection.

Feng and Kumar [22] introduced a fingerprint recognition system that focuses on accurately extracting minutiae, which are essential for reliable fingerprint matching. Their method utilizes a pixelwise local dilated neural network to capture detailed, fine-scale features, while a patch-wise global neural network ensures the inclusion of the fingerprint's broader structural context. By integrating these local and global features, the system generates a detailed minutiae location map, which is further fine-tuned through a recursive connected components algorithm for exact minutiae localization. Additionally, the system features innovative loss functions aimed at improving minutiae orientation detection and enhancing the overall discriminative ability of the extracted features.

3 Proposed method

Given the success of machine learning and deep learning in image processing and related tasks, we have developed our method based on deep learning techniques. In this section, we outline our GAN model, which uses frequency-domain patches to enhance image patches. This method deblurs patches, removes creases, and enhances them for processing finger and palm photos. Additionally, we describe our DCNN model for SP detection, which is used to identify SPs from enhanced frequency-domain patches. We also discuss a separate DCNN model designed to extract minutiae from the enhanced spatial-domain patches. Figure 1 illustrates the overall workflow for processing hand photos in our multibiometric system.

We begin by outlining the necessary preprocessing steps, followed by a detailed explanation of the architecture of our models. Initially, we outline the necessary preprocessing steps. Following that, we detail the architecture of our models.

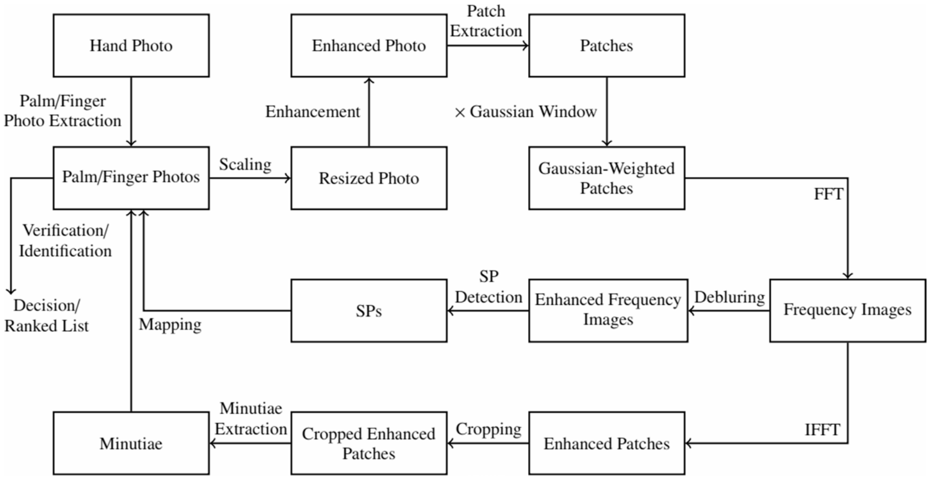


Figure 1 Workflow diagram for the processing of hand photos in our multibiometric system. The diagram illustrates the sequence from initial hand photo acquisition to minutiae extraction and SP detection, followed by verification or identification

3.1 Extraction of palm and fingertips from hand photos

The initial step involves extracting images of the palm and fingertips from a hand photo. For this purpose, we employ a DCNN model as outlined by Genovese et al. [23]. To isolate the fingertips and remove their background, we also apply the method proposed by Marasco and Vurity [24]. Figure 2 illustrates a palm photo and a fingertip image obtained using this extraction method after the background has been removed from the fingertip image.

3.2 Scaling

Contact-based fingerprints and palmprints maintain a constant distance from the sensor during capture, whereas contactless fingerprints, contactless palmprints, finger photos, and palm photos are subject to variable distances. As a result, scaling is an essential step, especially during the matching process, and must be automated in an automatic system. In this paper, we adopt the rearranged Fourier subbands (RFS) frequency estimation method proposed by Kunsuk and Areekul [25] to address this challenge. The method begins by dividing the enhanced fingertip image and enhanced palm photo into non-overlapping blocks. For each block, a window that overlaps with neighboring blocks is considered. This window is then processed through a short-time Fourier transform (STFT) to assess friction ridge frequencies in both finger and palm photos captured from varying distances. A Gaussian window is applied to suppress the direct current (DC) component and minimize edge effects, ensuring a smoother frequency analysis. The Fourier coefficients obtained from the STFT are subsequently reorganized into subbands that highlight the friction ridge frequencies. By converting these coefficients from rectangular to polar coordinates, the method identifies the subband with the highest spectral magnitude, enabling accurate estimation of friction ridge frequencies and scaling ratios across fingertip images and palm photos taken from different distances from the smartphone camera. To avoid confusion, we use the term ‘finger photo’ instead of ‘fingertip image’ from this point onward. Figure 3 presents the subband with the maximum spectral magnitude for each window, along with the scaling ratio for two finger photos. The same methodology is applied to palm photos. To scale finger and palm photos,

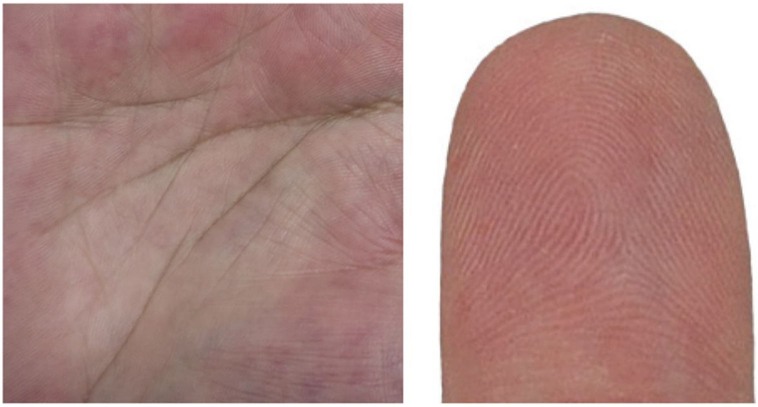


Figure 2 The image on the left shows a palm photo extracted using the method proposed by Genovese et al. [23], while the image on the right is a finger photo extracted using the method proposed by Marasco and Vurity [24] from a finger image obtained through Genovese et al.’s method

one can either adjust the resolution or resize the images. Since fingerprint and palmprint recognition require a constant resolution, such as 500 pixels per inch (ppi), we opt to resize the palm and finger photos. In this figure, the frequency representation for the upper finger photo is 8.719, while for the lower finger photo it is 7.570. The calculated scale ratio of 1.151 indicates that the top finger photo should be enlarged by this factor to achieve proper scaling.

3.3 Initial enhancement of palm and finger photos

Although the GAN model is designed to enhance each patch, an initial enhancement step is essential. The enhancement method used in Subsection 3.2 is not applied here due to its tendency to produce noisy results in some areas; instead, the scale ratio is utilized solely for resizing finger and palm photos. Initially, finger and palm photos are resized according to the scale ratio, followed by the application of contrast-limited adaptive histogram equalization (CLAHE). Figure 4 illustrates two palm photos, their grayscale counterparts, and the enhancement resulting from CLAHE. Our database images are categorized into four types: dry palm, dry fingertip, wet palm, and wet fingertip, which are detailed in Subsection 4.1.

3.4 Quality assessment and OF estimation

To train our models, we require two types of data: labels and inputs. For our patch-based GAN model, high-quality patches are needed as labels. We generate these labels by creating a reliability map that assesses the quality of the patches. For input data preparation, we introduce noise, apply a blur filter, and include patches with creases.

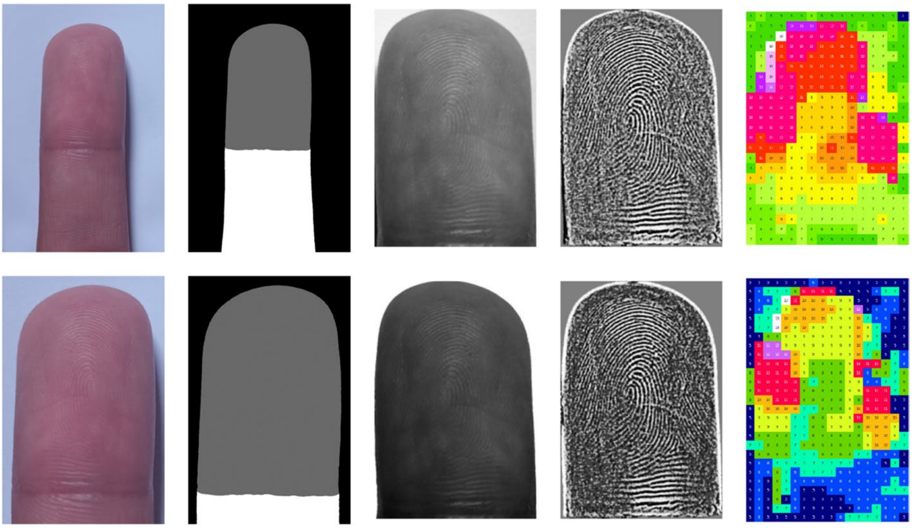


Figure 3 Illustration of the subband number containing the maximum spectral magnitude within each window. The top left panel features a finger image, with the image directly below showing the same finger image captured at a closer distance to the smartphone camera. The rightmost column depicts the mask used for fingertip extraction, where gray denotes the fingertip, white denotes the remaining finger, and black denotes the background. Adjacent columns display the finger photos, the enhanced finger photos, and the subband number with the maximum spectral magnitude for each window

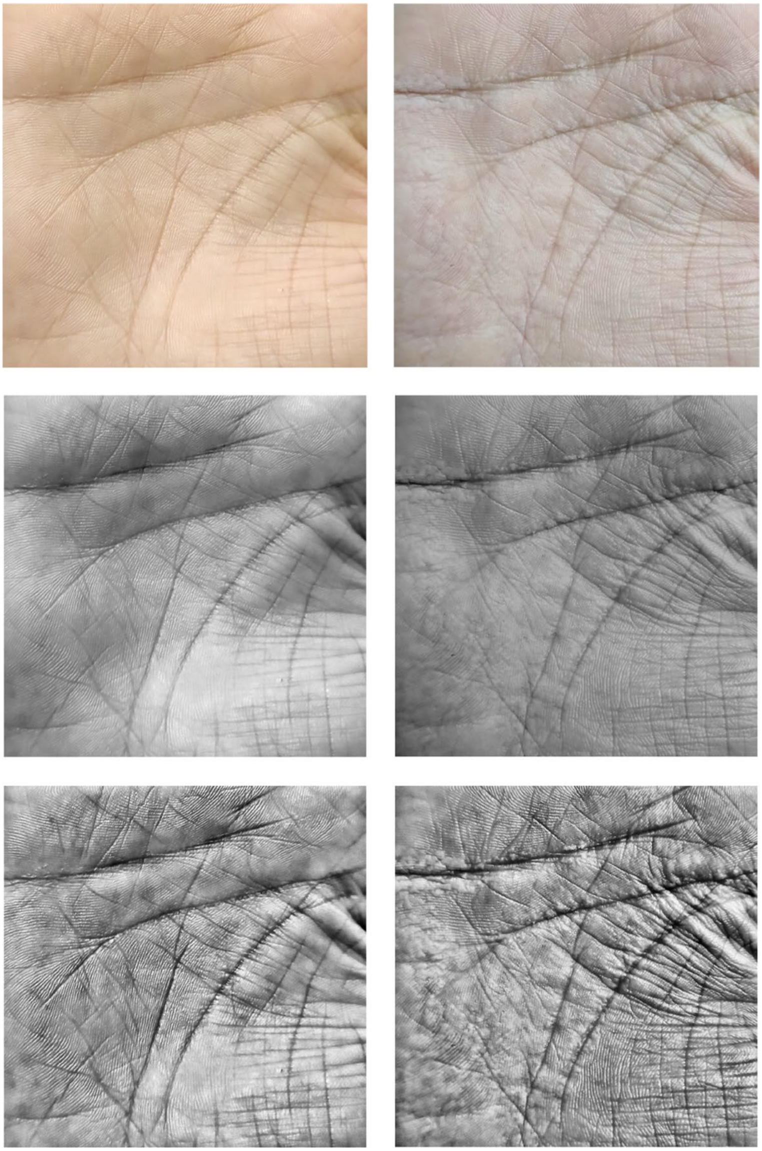


Figure 4 Comparison of dry and wet palm photos of the same hand under different lighting conditions. The top-left image shows a dry palm photo with yellow light, while the top-right image displays a wet palm photo after being submerged in water for 15 minutes, illuminated by white light. The middle row features gray-scale versions of these palm photos, and the bottom row showcases the images after applying CLAHE

To prepare labels for training our DCNN models for minutiae extraction and SP detection, OF data are essential. We utilize the OF estimation method proposed by Khan et al. [26]. Figure 5 illustrates the OF and reliability map generated for a palm photo. Notably, this OF estimation method demonstrates robustness, even in the presence of creases.

3.5 Our GAN model for deblurring and enhancing patches

In this study, we segment each finger photo and palm photo into non-overlapping blocks of 16×16 pixels and generate corresponding windows centered on each block with a size of 64×64 pixels. These windows inevitably overlap with neighboring blocks, which we refer to as ‘patches’–a term used to describe both the windows and the subimages derived from them. Compared to smaller blocks, these larger patches offer enhanced noise resilience and improved resolution in the frequency domain. However, the application of the fast Fourier transform (FFT) to a block or patch may introduce boundary artifacts, resulting in periodic discontinuities in the Fourier transform. To reduce these artifacts, we apply a Gaussian window with a standard deviation of $\sigma = 16$, which was determined experimentally to provide a good balance between artifact suppression and preservation of ridge details. Figure 6 illustrates a patch and its corresponding frequency image, as well as the result of multiplying this patch by a Gaussian window and the subsequent frequency image.

In the training stage of our GAN model, we utilize a reliability map to generate labels and inputs by extracting patches with an average reliability score above 0.9, a threshold experimentally established to provide an optimal balance between preserving ridge detail and suppressing noise. These patches are enhanced using the method proposed by Khodadoust

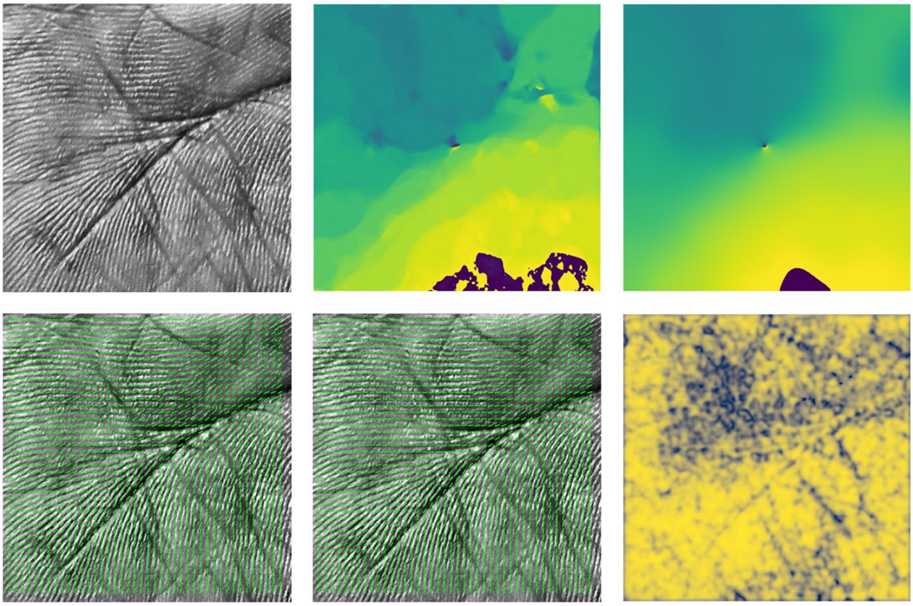


Figure 5 OF and reliability assessment for a palm photo: The top-left image represents the palm photo after applying CLAHE. The top-middle and top-right images respectively depict the OF before and after applying Khan et al.’s method [26]. The bottom-left and bottom-middle images display the OF on the photo, while the bottom-right image presents the reliability map

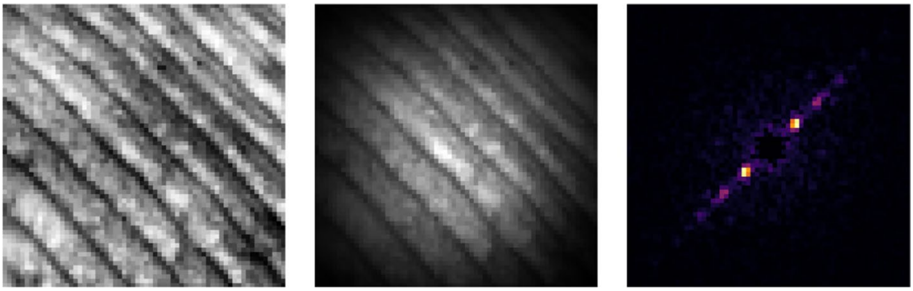


Figure 6 Illustration of a patch processed with a Gaussian window. The left panel shows the original patch. The middle panel depicts the patch after multiplication with a Gaussian window with $\sigma = 16$. The right panel presents the frequency-domain representation of the processed patch. For clarity, the DC component and certain neighboring frequencies have been set to zero

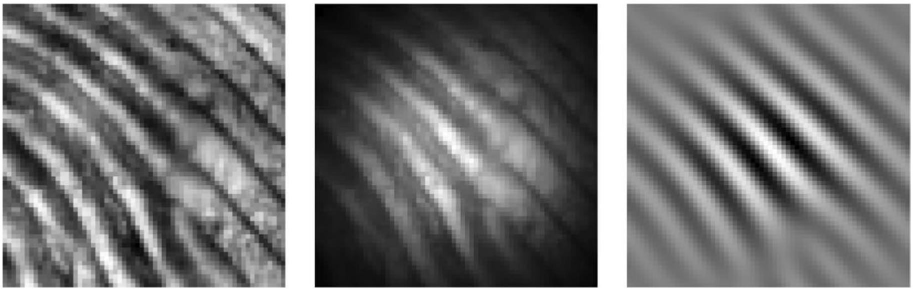


Figure 7 Patch enhancement. The left panel displays the original patch, the middle panel shows the patch after applying a Gaussian window, and the right panel presents the enhanced patch

et al. [10], followed by a manual review to discard any incorrectly enhanced patches. This method is chosen for its effectiveness in reducing noise and connecting ridges in the presence of creases. Figure 7 shows the result of enhancing a patch using this method.

To prepare the inputs, we introduce random noise and apply blur filters to the patches independently. Uniform noise is applied within the range of 0 to 7, and Gaussian noise is introduced within the range of 0 to 5. Blurring is performed using Gaussian blur with a radius between 0 and 4, and motion blur with angles 0, 45, and 135 degrees and distances between 0 and 20. Figure 8 shows different blur filters on a patch.

Our GAN model use frequency images 64×64 and its prediction is a frequency images 64×64. We selected the frequency domain for its substantial benefits over the spatial domain. This domain enables us to focus on ridge spectra that are confined within a narrow bandwidth, which simplifies the process of removing background noise and mitigating aliasing issues. This approach is particularly effective for deep learning models, as it minimizes the need for extensive weight coefficients and reduces training time. The Fourier transform plays a crucial role by decomposing images into their frequency components, allowing for targeted enhancement. By amplifying high-frequency components, we address blurriness and restore sharpness, while adjusting low-frequency components helps correct large-scale distortions such as wrinkles. Furthermore, frequency-domain processing supports the accurate preservation of genuine minutiae and SPs, leading to superior detail restoration and

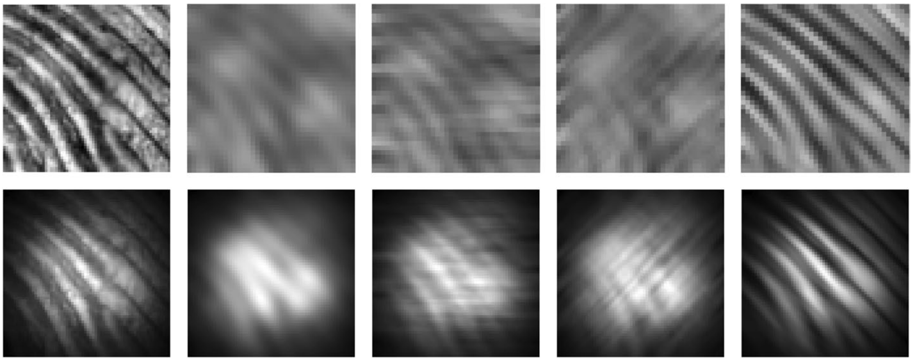


Figure 8 Application of blur filters on a patch. The top left image shows the original patch. Progressing from left to right, the subsequent images display the patch with Gaussian blur (radius 3 pixels) and motion blur applied at angles of 0, 45, and 135 degrees, each with a distance of 15 pixels. The bottom row presents the results of each corresponding top image after multiplication with a Gaussian window

image quality. Additionally, reducing spectra associated with creases aids in connecting ridges, enhancing the overall consistency and clarity of the images [12]. Figure 9 illustrates the architecture of our model, with the generator based on the encoder-decoder structure. Tables 1 and 2 provide further details of the model.

To assess the similarity between two frequency images–one representing the input patch, denoted as $P(u, v)$, and the other representing the label's frequency image, denoted as $L(u, v)$–we utilize Pearson's correlation coefficient. This metric quantifies the degree of similarity between the two frequency images. In the context of our model training process, we define the following loss function:

$$r = \frac{\sum_{i=1}^{n}\left(P_i(u,v) - \bar{P}(u,v)\right)\left(L_i(u,v) - \bar{L}(u,v)\right)}{\sqrt{\sum_{i=1}^{n}\left(P_i(u,v) - \bar{P}(u,v)\right)^2 \sum_{i=1}^{n}\left(L_i(u,v) - \bar{L}(u,v)\right)^2}}, \tag{1}$$

$$\text{loss}(P, L) = 1 - r, \tag{2}$$

where $P_i(u,v)$ and $L_i(u,v)$ represent the pixel values at position i in the images $P(u, v)$ and $L(u, v)$, respectively. $\bar{P}(u,v)$ and $\bar{L}(u,v)$ are the mean pixel values of the respective images, and n is the total number of pixels.

We also apply the inverse FFT (IFFT) to the model's output and convert it to a binary image to evaluate the similarity between this binary patch and the binary patch from the label. For this purpose, we employ the Jaccard Index, also known as the intersection over union (IoU) metric. The IoU quantifies the overlap between two binary images relative to their combined area, with higher IoU values indicating a stronger resemblance between the images. Mathematically, the IoU is defined as:

$$\text{IoU} = \frac{\text{Intersection of L(x, y) and P(x, y)}}{\text{Union of L(x, y) and P(x, y)}}, \tag{3}$$

$$\text{loss(IoU)} = 1 - \text{IoU}, \tag{4}$$

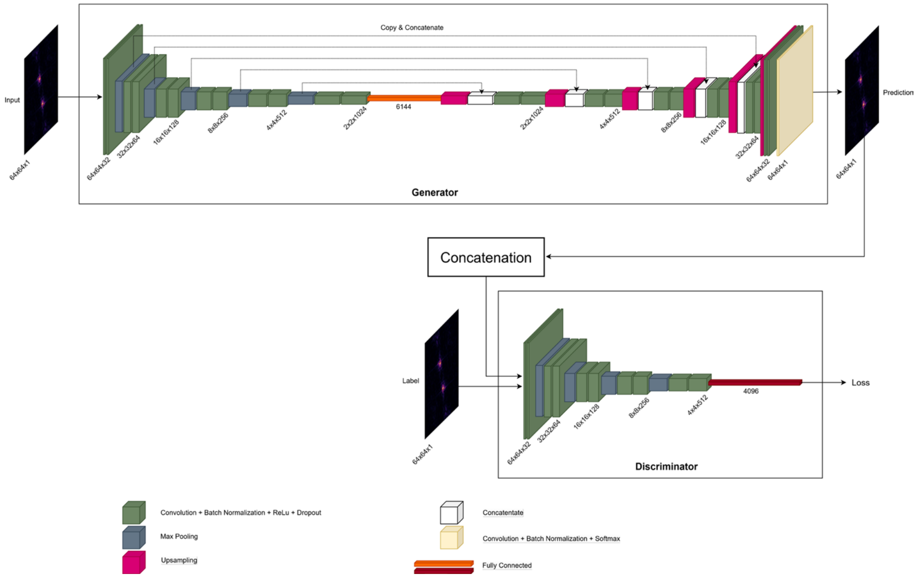


Figure 9 Architecture of our GAN. The diagram illustrates the structure of the GAN, including both the generator and discriminator networks, as well as their connections and interactions during the training process

where $L(x, y)$ represents the binary label image in the spatial domain, and $P(x, y)$ corresponds to the binary image obtained from the predicted frequency image after transforming it into the spatial domain and converting it to binary. This metric is crucial for accurately evaluating the alignment between two binary images, especially when addressing the nuances involved in converting pixels from grayscale to binary.

Our final loss function is a linear combination of these two separate loss functions, as illustrated below:

$$\text{Loss} = \lambda_1 \cdot \text{loss}(P, L) + \lambda_2 \cdot \text{loss}(\text{IoU}), \tag{5}$$

where λ_1 and λ_2 are experimentally determined weights, set to 0.53 and 0.47, respectively.

3.6 Our DCNN model for extracting minutiae

In this paper, we propose an encoder-decoder-based DCNN model for extracting minutiae from patches generated by our GAN model. The GAN outputs frequency-domain images of size 64×64. To process these images, we apply the IFFT to convert them back into the spatial domain, which is then used as input for our DCNN minutiae extraction model. A Fourier coefficient is a complex number that is decomposed into its magnitude and phase components. The patches representing these magnitudes and phases are termed the spectral patch and the phase patch, respectively [12]. In our GAN approach, we utilize the spectral patches for ridge restoration, while keeping the phase patches unchanged. This ensures that the genuine minutiae locations, which are associated with the phases of the Fourier spectra, are preserved and remain unaffected by our GAN model. Therefore, we employ the spa-

Table 1 Detailed architecture of our GAN, including standard convolutions, max pooling, and transpose convolutions. The figure highlights the structure of the generator network

Operation	Kernel size	Stride	Number of kernels	Out-put shape
Conv.	3×3	1	32	$64 \times 64 \times 32$
Conv.	3×3	1	32	$64 \times 64 \times 32$
Max Pooling	2×2	2	-	$32 \times 32 \times 32$
Conv.	3×3	1	64	$32 \times 32 \times 64$
Conv.	3×3	1	64	$32 \times 32 \times 64$
Max Pooling	2×2	2	-	$16 \times 16 \times 64$
Conv.	3×3	1	128	$16 \times 16 \times 128$
Conv.	3×3	1	128	$16 \times 16 \times 128$
Max Pooling	2×2	2	-	$8 \times 8 \times 128$
Conv.	3×3	1	256	$8 \times 8 \times 256$
Conv.	3×3	1	256	$8 \times 8 \times 256$
Max Pooling	2×2	2	-	$4 \times 4 \times 256$
Conv.	3×3	1	512	$4 \times 4 \times 512$
Conv.	3×3	1	512	$4 \times 4 \times 512$
Max Pooling	2×2	2	-	$2 \times 2 \times 512$
Conv.	3×3	1	1,024	$2 \times 2 \times 1,024$
Conv.	3×3	1	1,024	$2 \times 2 \times 1,024$
TranspConv.	2×2	2	1,024	$2 \times 2 \times 1,024$
Conv.	3×3	1	1,024	$2 \times 2 \times 1,024$
Conv.	3×3	1	1,024	$2 \times 2 \times 1,024$
TranspConv.	2×2	2	512	$4 \times 4 \times 512$
Conv.	3×3	1	512	$4 \times 4 \times 512$
Conv.	3×3	1	512	$4 \times 4 \times 512$
TranspConv.	2×2	2	256	$8 \times 8 \times 256$
Conv.	3×3	1	256	$8 \times 8 \times 256$
Conv.	3×3	1	256	$8 \times 8 \times 256$
TranspConv.	2×2	2	128	$16 \times 16 \times 128$
Conv.	3×3	1	128	$16 \times 16 \times 128$
Conv.	3×3	1	128	$16 \times 16 \times 128$
TranspConv.	2×2	2	64	$32 \times 32 \times 64$
Conv.	3×3	1	64	$32 \times 64 \times 64$
Conv.	3×3	1	64	$32 \times 64 \times 64$
TranspConv.	2×2	2	32	$64 \times 64 \times 32$
Conv.	3×3	1	32	$64 \times 64 \times 32$
Conv.	3×3	1	32	$64 \times 64 \times 32$
Conv.	3×3	1	1	$64 \times 64 \times 1$

Conv. convolution, *TranspConv.* transposed convolution

tial domain for the DCNN minutiae extraction model, as minutiae extraction from spectra patches is not practical. Figure 10 illustrates the architecture of our DCNN minutiae extraction model. Table 3 provides further details of the model.

In contrast to some existing patch-based minutiae extraction models, such as ContactlessNet [17], which merely verify the presence of a minutia within a patch and assign a single minutia to the entire patch, our approach is capable of precisely locating the exact position of minutiae within the patches.

Table 2 Detailed architecture of our GAN, including standard convolutions and max pooling. The figure highlights the structure of the descriminator network

Operation	Kernel size	Stride	Number of kernels	Output shape
Conv.	3 × 3	1	32	64 × 64 × 32
Conv.	3 × 3	1	32	64 × 64 × 32
Max Pooling	2 × 2	2	-	32 × 32 × 32
Conv.	3 × 3	1	64	32 × 32 × 64
Conv.	3 × 3	1	64	32 × 32 × 64
Max Pooling	2 × 2	2	-	16 × 16 × 64
Conv.	3 × 3	1	128	16 × 16 × 128
Conv.	3 × 3	1	128	16 × 16 × 128
Max Pooling	2 × 2	2	-	8 × 8 × 128
Conv.	3 × 3	1	256	8 × 8 × 256
Conv.	3 × 3	1	256	8 × 8 × 256
Max Pooling	2 × 2	2	-	4 × 4 × 256
Conv.	3 × 3	1	512	4 × 4 × 512
Conv.	3 × 3	1	512	4 × 4 × 512

Conv. convolution

To train our model, we require labeled minutiae data for each patch. To create these labels, we begin by initializing a binary image of size 16 × 16 set to zero, corresponding to the size of non-overlapping blocks. The patches produced by our GAN model are then binarized, skeletonized, and their minutiae are identified. We focus on the central 16 × 16 region of these patches. If a minutia is detected within this central region, its exact coordinates are marked as one in the corresponding binary label image.

The input images to our DCNN model are 18 × 18 centrally cropped patches derived from the GAN output. Figure 11 shows an example of a cropped image from a GAN-generated patch. The reason for using a slightly larger size than the 16 × 16 blocks is to ensure that any minutiae located near the block borders are not discarded during extraction. The output of our model is a binary image of size 16 × 16, matching the size of the labels and blocks. Given the low number of minutiae in both the labels and model outputs, we utilize the focal loss function [27] in this model to effectively handle the severe class imbalance by down-weighting the loss contribution of well-classified background pixels and focusing the training on the relatively rare minutiae points.

After extracting the minutiae from each block, they are mapped to the corresponding finger or palm photo. Figure 12 compares the minutiae extracted by our model with those obtained using the ContactlessNet model. Our model shows fewer false minutiae and preserves more true minutiae.

3.7 Our DCNN model for detecting SPs

In this study, we propose an encoder-decoder-based DCNN model for detecting SPs. Figure 13 illustrates the architecture of our model. This model processes frequency images generated by our GAN model. Although both spatial-domain and frequency-domain patches can be used for SP detection, we opt for frequency-domain patches due to their advantages. The architecture of this model closely follows the generator component of our GAN, as summarized in Table 1.

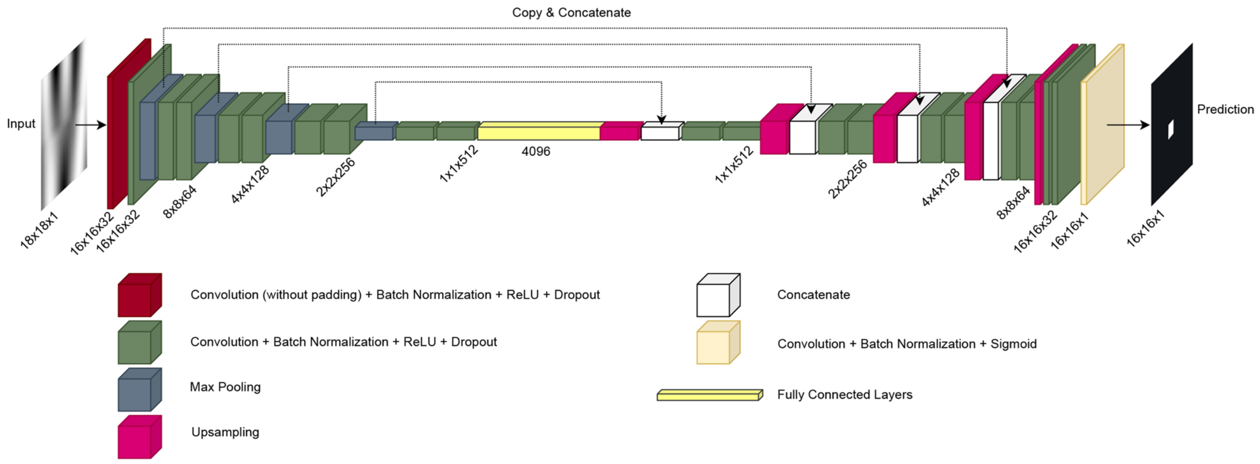


Figure 10 Architecture of our DCNN for minutiae extraction

To generate SP labels for training, we utilize the method proposed by Zhu et al. [28], which is based on OF estimation. By employing Khan et al.'s method [26], known for its accurate OF estimation, Zhu et al.'s method produces precise results. We first apply Khan et al.'s method to obtain the OF, followed by Zhu et al.'s method to detect SPs. Only blocks containing SPs are selected, and the corresponding patches are considered. We manually review these SPs, discarding any that are incorrectly detected. For correctly identified SPs, we create a 16×16 binary image, similar to the one used in our DCNN minutiae extraction model. In this binary image, the background is set to zero, while the exact location of the SP is set to one, forming the label for our DCNN SP detection model.

The input for our model is a frequency image of 64×64 patches from the output of our GAN model. Figure 14 displays three patches containing SPs that are input to our GAN model. The right column shows the results produced by the GAN, which are used by our DCNN SP detection model. The output of our DCNN SP detection model is a 16×16 binary image. With the limited number of SPs in both the labels and model outputs, we employ the focal loss function [27] to address class imbalance, for the same reason as in minutiae detection. Figure 15 illustrates the SPs detected from a finger photo by our DCNN SP detection model.

3.8 Score-level fusion

In our multimodal approach, we encompass both the finger photos and palm photos, extracting minutiae and SPs from both sources. The primary advantage of our method lies in its ability to consistently extract the same minutiae and SPs from both the finger photos and palm photos, which are subsequently used for matching purposes. By successfully extracting minutiae and SPs from both finger photos and palm photos, we streamline the process by employing a unified feature extractor, eliminating the need for result normalization when fusing information from these two image types. Moreover, our score-level fusion algorithm is designed to work with various biometric traits, allowing for their seamless integration using a weighting method, all without the necessity of score normalization.

After extracting minutiae and SPs from finger photos and palm photos, we compute matching scores for each set of images. Although fusion at the feature level–by aggregating all extracted minutiae into a single template–is an option, we choose score-level fusion. This approach involves creating separate templates for each palm photo and finger photo to enhance accuracy. For the matching process, we employ the method proposed by Shi et al. [29], which derives deep features, referred to as minutiae features, from the minutiae and performs matching using a graph neural network (GNN).

In some cases, minutiae from a particular finger photo or palm photo may not be extractable. To address this, we generate individual templates for each photo and assign corresponding weights to their matching scores. If the system fails to extract minutiae from a photo, the matching score is adjusted accordingly, leading to a reduction in the overall matching score.

In our approach, we evaluate the top 10 candidates with the highest matching scores, ranging from 0 to 1, for each individual finger and the palm, including the thumb, index finger, middle finger, ring finger, little finger, and palm. For each finger, we select the highest score and verify whether its ID appears among the candidate IDs for other fingers and the palm. If the ID is absent, we assign a weight of one-sixth to the score. If it is present in

Table 3 Detailed architecture of our GAN, including standard convolutions, max pooling, and transpose convolutions. The figure highlights the structure of the generator network

Operation	Kernel size	Stride	Number of kernels	Output shape
Conv.	3×3	1	32	$16 \times 16 \times 32$
Conv.	3×3	1	32	$16 \times 16 \times 32$
Max Pooling	2×2	2	-	$8 \times 8 \times 32$
Conv.	3×3	1	64	$8 \times 8 \times 64$
Conv.	3×3	1	64	$8 \times 8 \times 64$
Max Pooling	2×2	2	-	$4 \times 4 \times 64$
Conv.	3×3	1	128	$4 \times 4 \times 128$
Conv.	3×3	1	128	$4 \times 4 \times 128$
Max Pooling	2×2	2	-	$2 \times 2 \times 128$
Conv.	3×3	1	256	$2 \times 2 \times 256$
Conv.	3×3	1	256	$2 \times 2 \times 256$
Max Pooling	2×2	2	-	$1 \times 1 \times 256$
Conv.	3×3	1	512	$1 \times 1 \times 512$
Conv.	3×3	1	512	$1 \times 1 \times 512$
TranspConv.	2×2	2	512	$1 \times 1 \times 512$
Conv.	3×3	1	512	$1 \times 1 \times 512$
Conv.	3×3	1	512	$1 \times 1 \times 512$
TranspConv.	2×2	2	256	$2 \times 2 \times 256$
Conv.	3×3	1	256	$2 \times 2 \times 256$
Conv.	3×3	1	256	$2 \times 2 \times 256$
TranspConv.	2×2	2	128	$4 \times 4 \times 128$
Conv.	3×3	1	128	$4 \times 4 \times 128$
Conv.	3×3	1	128	$4 \times 4 \times 128$
TranspConv.	2×2	2	64	$8 \times 8 \times 64$
Conv.	3×3	1	64	$8 \times 8 \times 64$
Conv.	3×3	1	64	$8 \times 8 \times 64$
TranspConv.	2×2	2	32	$16 \times 16 \times 32$
Conv.	3×3	1	32	$16 \times 16 \times 32$
Conv.	3×3	1	32	$16 \times 16 \times 32$
Conv.	3×3	1	1	$16 \times 16 \times 1$

Conv. convolution, *TranspConv.* transposed convolution

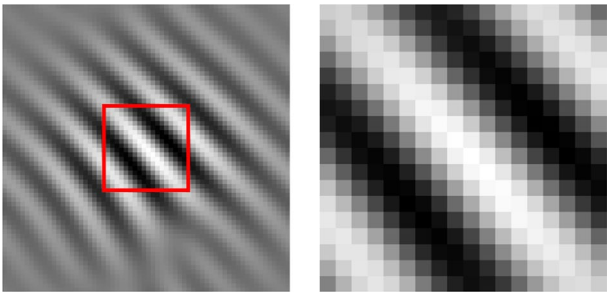


Figure 11 Providing input images for our DCNN minutiae extraction model. The image on the left represents the patch generated by our GAN model, while the image on the right is the corresponding centrally cropped section used as input for the DCNN minutiae extraction model

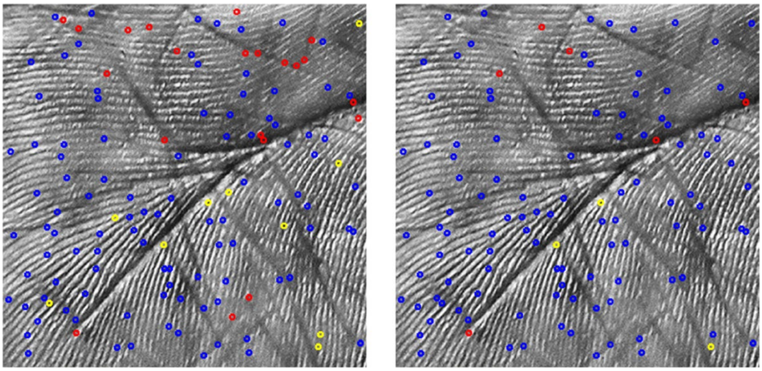


Figure 12 Minutiae comparison with contactlessNet: the left image shows minutiae extracted by ContactlessNet, while the right image presents minutiae extracted using our model. Genuine minutiae are highlighted in blue, spurious detections are marked in red, and undetected minutiae are encircled in yellow

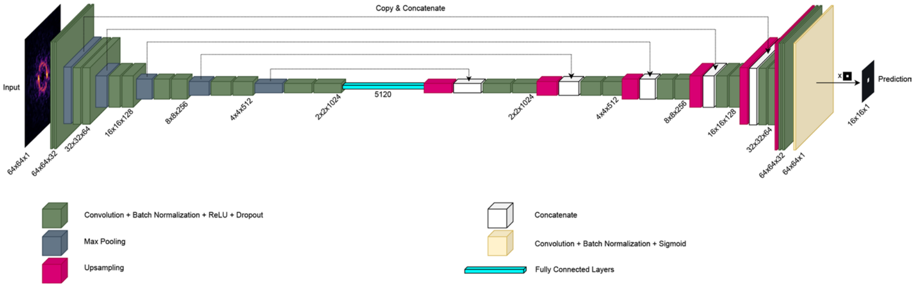


Figure 13 Architecture of our DCNN for SP detection

one other finger, we assign a weight of two-sixths. If it appears in all other fingers and the palm, we assign a weight of six-sixths. This process is repeated for each finger and the palm, resulting in updated scores.

After updating the scores, the next step is to calculate the final matching score across all fingers and the palm for each ID in the candidate list. This is done by applying (6) to combine the matching scores.

$$S_{id} = w_t s_{t,id} + w_i s_{i,id} + w_m s_{m,id} + w_r s_{r,id} + w_l s_{l,id} + w_p s_{p,id}. \tag{6}$$

where w_t, w_i, w_m, w_r, and w_l denote the weights for the thumb, index finger, middle finger, ring finger, and little finger, respectively, and w_p denotes the weight for the palm. Specifically, $w_t = w_i = w_m = w_r = w_l = 0.16$ and $w_p = 0.2$, ensuring that $w_t + w_i + w_m + w_r + w_l + w_p = 1$. The index *id* indicates the ID of each candidate. The variable *s* represents the matching score of each finger or palm, with $s_{i,1}$, for example, indicating the matching score of the index finger with ID 1.

For each ID, we multiply the score of each finger by a weight of 0.16 and the score of the palm by a weight of 0.2, as determined through experimentation. Finally, the IDs are ranked according to their calculated scores. Table 4 presents a numerical example of the

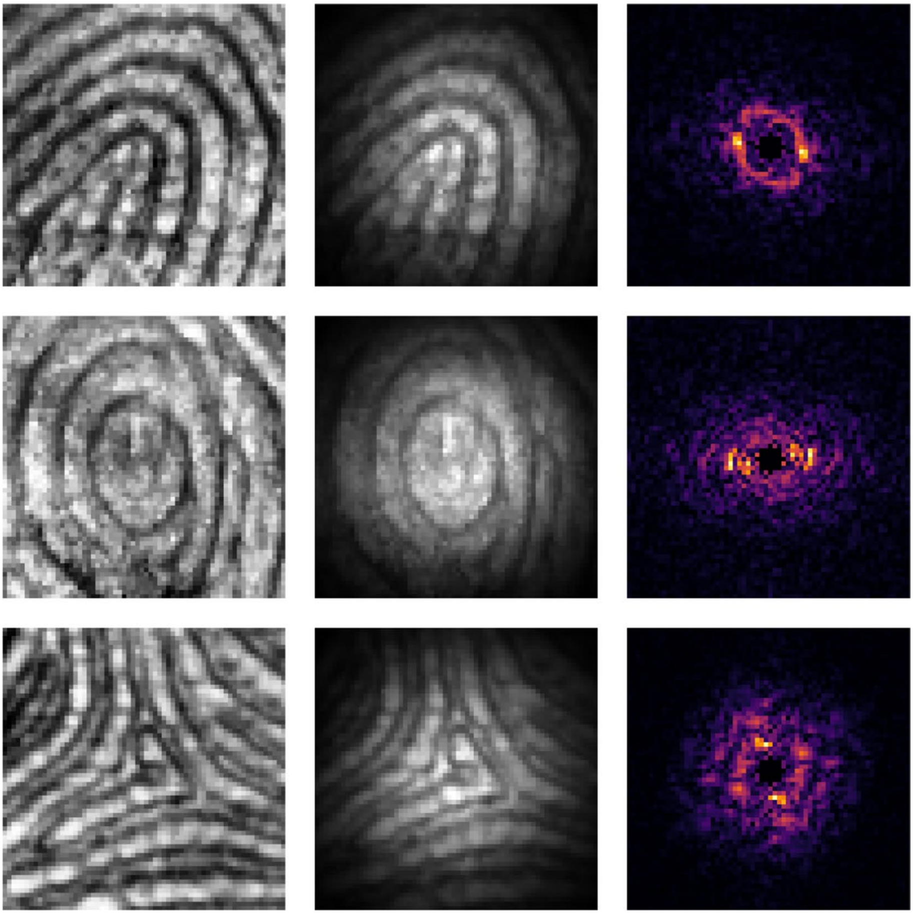


Figure 14 Patches containing SPs are shown. The left column presents the original patches with SPs. The middle column displays these patches after multiplication by a Gaussian window. The right column illustrates the output of our GAN, which serves as input to our DCNN SP detection model. For improved clarity, the central regions of the frequency images have been set to zero

process, illustrating the calculation of the final matching score for each ID. In the first row of the top table, the highest score is for the thumb, which is 0.823. Our algorithm searches the other columns to determine if the same ID exists in any of them. It is present in five other columns, so its score is multiplied by five-sixths, reducing it to 0.686. The next highest score in this row is related to the palm. We apply the same process as with the thumb, reducing its score to 0.669. This process is repeated for each row. Finally, each column is sorted based on the new scores. The last table demonstrates how (6) is applied to obtain the final scores for each ID.

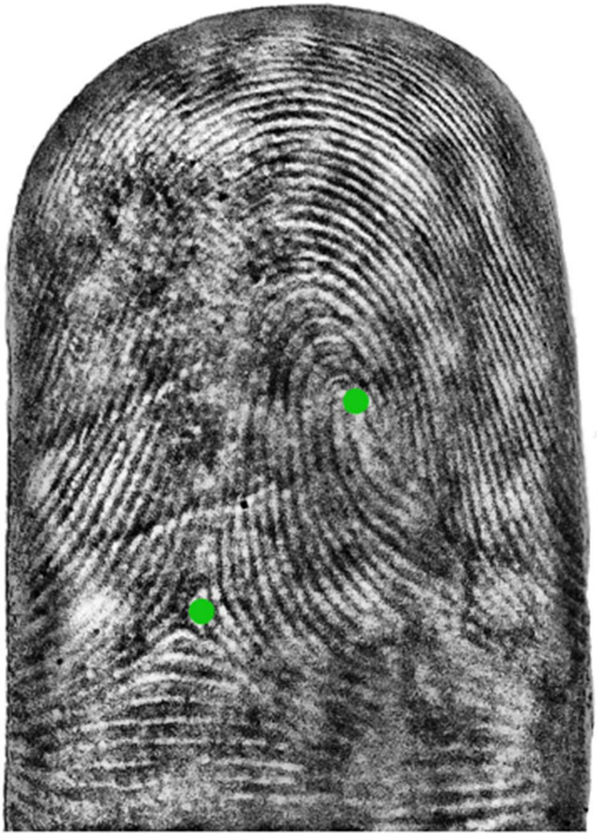


Figure 15 SP detection by our DCNN model

4 Experimental results

This section provides a comprehensive analysis of the experiments conducted to evaluate the reliability and effectiveness of our proposed multibiometric system, which utilizes finger photos and palm photos. We first introduce the databases employed and outline the experimental procedures. Subsequently, we present the experimental results obtained and compare them with the current SOTA in the field.

4.1 Database

To the best of our knowledge, no existing database contains high-resolution hand photos from which minutiae can be extracted from both finger photos and palm photos. Since our system relies on the analysis of such detailed images, it was essential for us to develop a custom database.

To collect hand photos, we used a mini photo studio equipped with two color lights, white and yellow. Figure 16 illustrates the photo studio setup and how a hand was positioned within it. Photos were taken from above. For photography, three different smartphones–Motorola G60, Samsung S22 Ultra, and iPhone 14 Pro Max–were used, with the highest resolution from each chosen. This approach ensured the acquisition of varied photos and

Table 4 Numerical example of score fusion in our multibiometric system. The top table shows the top ten candidates for each finger and palm, the middle table presents the updated scores according to our algorithm, and the bottom table illustrates the fusion of finger and palm scores to produce the final hand score

Thumb		Index		Middle		Ring		Little		Palm	
ID	Score	ID	Score	ID	Score	ID	Score	ID	Score	ID	Score
1	0.823	2	0.642	5	0.745	3	0.682	2	0.731	1	0.803
3	0.642	1	0.635	3	0.702	1	0.673	1	0.725	2	0.795
2	0.639	13	0.614	2	0.648	2	0.635	5	0.716	4	0.782
4	0.599	5	0.601	8	0.647	5	0.609	3	0.653	5	0.761
5	0.580	4	0.593	4	0.622	9	0.584	9	0.640	3	0.746
8	0.525	8	0.584	7	0.619	8	0.562	4	0.631	7	0.727
6	0.518	3	0.569	6	0.601	6	0.541	6	0.625	6	0.719
7	0.503	9	0.561	9	0.583	4	0.533	7	0.602	8	0.701
10	0.501	7	0.550	10	0.565	16	0.521	21	0.593	10	0.685
15	0.489	6	0.521	11	0.559	7	0.511	19	0.581	9	0.671
Thumb		Index		Middle		Ring		Little		Palm	
ID	Score	ID	Score	ID	Score	ID	Score	ID	Score	ID	Score
1	0.686	2	0.642	5	0.745	3	0.682	2	0.731	2	0.795
3	0.642	5	0.601	3	0.702	2	0.635	5	0.716	4	0.782
2	0.639	4	0.593	2	0.648	5	0.609	3	0.653	5	0.761
4	0.599	3	0.569	4	0.622	1	0.561	4	0.631	3	0.746
5	0.580	7	0.550	7	0.619	6	0.541	6	0.625	7	0.727
6	0.518	1	0.529	6	0.601	4	0.533	1	0.604	6	0.719
7	0.503	6	0.521	8	0.539	7	0.511	7	0.602	1	0.669
8	0.437	8	0.487	9	0.486	9	0.487	9	0.533	8	0.584
10	0.250	9	0.467	10	0.282	8	0.468	21	0.099	9	0.373
15	0.081	13	0.102	11	0.093	16	0.087	19	0.097	10	0.343

ID	Score
3	$0.642 \times 0.16 + 0.569 \times 0.16 + 0.702 \times 0.16 + 0.682 \times 0.16 + 0.653 \times 0.16 + 0.746 \times 0.2 = 0.5216$
1	$0.686 \times 0.16 + 0.529 \times 0.16 + 0 \times 0.16 + 0.561 \times 0.16 + 0.604 \times 0.16 + 0.669 \times 0.2 = 0.5146$
2	$0.639 \times 0.16 + 0.642 \times 0.16 + 0.648 \times 0.16 + 0.635 \times 0.16 + 0.731 \times 0.16 + 0.795 \times 0.2 = 0.5127$
5	$0.580 \times 0.16 + 0.601 \times 0.16 + 0.7450.16 \times +0.609 \times 0.16 + 0.716 \times 0.16 + 0.761 \times 0.2 = 0.50936$
4	$0.599 \times 0.16 + 0.593 \times 0.16 + 0.622 \times 0.16 + 0.561 \times 0.16 + 0.631 \times 0.16 + 0.782 \times 0.2 = 0.48976$
6	$0.518 \times 0.16 + 0.521 \times 0.16 + 0.601 \times 0.16 + 0.541 \times 0.16 + 0.625 \times 0.16 + 0.719 \times 0.2 = 0.45368$
7	$0.503 \times 0.16 + 0.550 \times 0.16 + 0.619 \times 0.16 + 0.511 \times 0.16 + 0.602 \times 0.16 + 0.727 \times 0.2 = 0.4534$
8	$0.437 \times 0.16 + 0.487 \times 0.16 + 0.539 \times 0.16 + 0.468 \times 0.16 + 0.533 \times 0.16 + 0.584 \times 0 : 2 = 0.39868$
9	$0.467 \times 0.16 + 0.486 \times 0.16 + 0.486 \times 0.16 + 0.487 \times 0.16 + 0.533 \times 0.16 + 0.373 \times 0 : 2 = 0.39648$
10	$0.250 \times 0.16 + 0.282 \times 0.16 + 0 \times 0.16 + 0.468 \times 0.16 + 0 \times 0.16 + 0.343 \times 0.2 = 0.26256$
11	$0 \times 0.16 + 0 \times 0.16 + 0.093 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0.343 \times 0.2 = 0.08996$
21	$0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0.099 \times 0.16 + 0 \times 0.2 = 0.01584$
19	$0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0.097 \times 0.16 + 0 \times 0.2 = 0.01552$
13	$0 \times 0.16 + 0.102 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.2 = 0.01632$
16	$0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0.087 \times 0.16 + 0 \times 0.16 + 0 \times 0.2 = 0.01392$
15	$0.081 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.16 + 0 \times 0.2 = 0.01296$

prevented dependency on a specific smartphone camera, while also expediting the photo collection process.

Hand photos were taken at a swimming pool. From each volunteer, 12 hand photos were captured under two conditions: three photos of each hand were taken before entering the

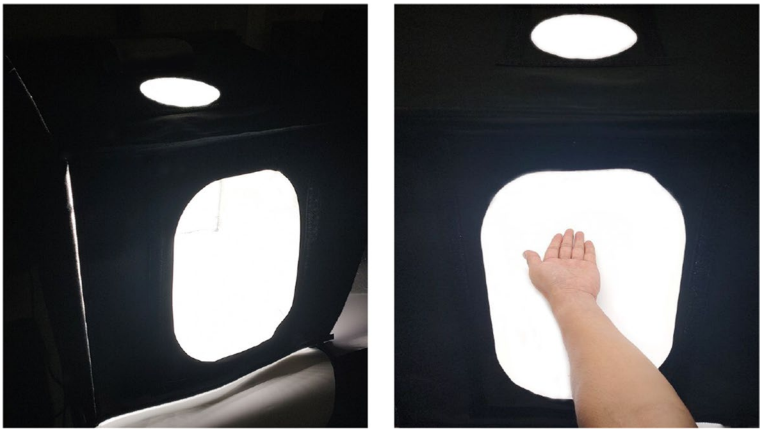


Figure 16 Mini photo studio equipment for hand photo capture

pool, and three more after 15 minutes in the water. For each set of three photos, two were taken from a constant distance of 25 cm from the camera, and one from 40 cm, or vice versa. This variation in distance was implemented because, in real-world scenarios, maintaining a consistent distance between a fingertip or palm and a smartphone camera can be challenging for everyday users.

A total of 2,500 volunteers participated, resulting in 30,000 hand photos. Photos from 1,500 volunteers were taken using the Motorola G60, 500 with the Samsung S22 Ultra, and 500 with the iPhone 14 Pro Max. Additionally, a supplementary dataset was collected from 200 volunteers, comprising 2,400 hand photos captured with the Huawei Pura 70 Ultra. This supplementary dataset was specifically created for cross-database evaluation, with the distance between the hand and the camera varying randomly.

Genovese et al.'s method [23] was utilized to process each hand photo, extracting the finger and palm photos. Following this, the segmentation technique developed by Marasco and Vurity [24] was applied to the finger photos to isolate the fingertips and generate binary masks. The images were then converted to grayscale and saved in BMP format. The final palm photos were $1,800 \times 1,800$ pixels in size, with a resolution of 500 ppi, while the finger photos were 600×600 pixels, also at 500 ppi. Additionally, palm photos of the left hand were horizontally flipped.

4.2 Experiment setup

Training was conducted on an NVIDIA Tesla A100 GPU. Our models were optimized using the adaptive moment estimation (ADAM) optimizer [30], with an initial learning rate of 0.0001 and momentum parameters set to $\beta_1 = 0.5$ and $\beta_2 = 0.999$. A batch size of 64 was employed throughout the training process to optimize model performance efficiently.

The training dataset consisted of 710,295 patches for palm photos and 694,748 patches for finger photos. The data was partitioned into 80% for training and 20% for testing, with the split performed at the volunteer (subject) level so that all patches from any given volunteer appeared exclusively in either the training or testing set.

To mitigate overfitting in our models, we implemented early stopping at 45, 35, and 40 epochs for the GAN model, the DCNN minutiae extraction model, and the DCNN SP detection model, respectively. The training durations were 105 hours for the GAN model, 60 hours for the DCNN minutiae extraction model, and 76 hours for the DCNN SP detection model. Training was halted at 140 epochs, 120 epochs, and 130 epochs for the respective models, with the optimal performance achieved at epochs 110, 98, and 106.

To effectively perform the matching process, we utilized a minutiae matching method capable of processing both finger and palm photos. We adopted the approach proposed by Shi et al. [29], which incorporates a DCNN model for minutiae feature extraction and a GNN for subsequent matching. In our experiments, minutiae are extracted initially using our models or standard techniques, after which Shi et al.'s method is applied for the matching phase.

To evaluate the performance of our GAN model, we utilized several metrics: mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and feature similarity index (FSIM) [31]. We evaluated our DCNN minutiae extraction model using precision, recall, and F1-score, while performance for our DCNN SP detection was assessed through detection rate (DR) and false alarm rate (FAR) [16]. For a comprehensive evaluation of both the minutiae extraction model and score-level fusion, we analyzed verification performance by examining the equal error rate (EER), receiver operating characteristic (ROC) curves, and the area under the ROC curve (AUC). Additionally, identification performance was assessed by measuring rank-1 accuracy and employing the cumulative match characteristic (CMC).

4.3 Hyperparameter tuning and overfitting prevention

To select optimal hyperparameters, we performed grid and manual searches over learning rate (1e-4–1e-5), batch size (32–128), and epoch limits. The ADAM optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.999$) with a learning rate of 0.0001 provided the best trade-off between convergence speed and stability. Early stopping was applied (45, 35, and 40 epochs for the GAN, minutiae DCNN, and SP DCNN respectively) to prevent overfitting. To further improve generalization, data augmentation (blur, noise injection, and variable capture distances) and subject-level train/test splits ensured no volunteer's patches appeared in both sets. These practices are consistent with established approaches, where systematic hyperparameter optimization and class-imbalance handling are emphasized to avoid overfitting and improve robustness.

4.4 Results

In this section, we assess the performance metrics of our models and compare our method with SOTA approaches.

4.4.1 Evaluation of our GAN model

We first conducted an ablation study on the use of a Gaussian window with five different σ values: 8, 12, 16, 20, and 24. The results in Table 5 show that $\sigma = 16$ achieves the best performance.

Table 5 Comparison of precision, recall, and F1-score for dry- and wet-type hand photos from our database using different Gaussian window σ values. The best value in each column is highlighted in bold

σ	Wet			Dry		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
8	78.93	86.02	82.32	81.77	88.31	84.91
12	82.11	89.44	85.62	85.11	91.11	88.01
16	**83.54**	**90.18**	**87.79**	**86.24**	**91.93**	**89.57**
20	81.67	88.92	85.14	84.41	90.46	87.33
24	76.47	83.73	79.94	80.33	87.21	83.63

Table 6 Comparison of our GAN model with other models using dry/wet-type photos from our database. The best value in each column is highlighted in bold

GAN model	Wet			Dry		
	PSNR (dB)	SSIM (%)	FSIM (%)	PSNR (dB)	SSIM (%)	FSIM (%)
DeblurGAN [32]	21.162	80.528	81.900	25.245	89.338	90.204
DeblurGAN-v2 [33]	25.598	84.263	86.227	29.671	91.824	92.672
FDeblur-GAN [34]	28.218	88.335	89.080	32.128	92.769	93.137
Ours	**31.147**	**91.669**	**93.101**	**35.025**	**93.947**	**95.465**

Given that our GAN model's objective is to deblur and enhance image patches, we evaluate its performance using spatial-domain metrics, including PSNR, SSIM, and FSIM. We present a comparative analysis of our GAN model against DeblurGAN [32], DeblurGAN-v2 [33], and FDeblur-GAN [34] in Table 6, using finger photos and palm photos from our database. PSNR values between 20-25 dB indicate low quality with noticeable distortion, 25-35 dB suggest moderate quality with minor artifacts, and values above 35 dB reflect high-quality images with minimal distortion. SSIM values below 0.5 indicate significant structural differences and poor quality, 0.5-0.8 suggest moderate quality with some visible differences, and values above 0.8 indicate high-quality images with well-preserved details. Similarly, FSIM values below 0.5 denote low feature similarity and poor quality, 0.5-0.8 indicate moderate quality with some differences, and values above 0.8 represent high similarity and excellent image quality.

4.4.2 Evaluation of our DCNN minutiae extraction model

Table 7 provides a summary and comparison of minutiae extraction accuracy for our DCNN model, evaluating performance through precision, recall, and F1-score on our database. A recovered minutia is deemed a true match if it falls within 8 pixels of the labeled minuti. This tolerance was selected after preliminary testing showed that smaller values unfairly rejected correctly matched minutiae displaced by minor misalignments or elastic deformations, while larger values risked accepting spurious matches. The one-to-one matching strategy ensures that each labeled minutia is paired with only a single predicted minutia. The one-to-one matching strategy ensures that each labeled minutia is paired with only a single predicted minutia. Figure 17 illustrates the confusion matrices of our DCNN minutiae extraction model for wet (left) and dry (right) samples.

Table 7 Comparison of precision, recall, and F1-score for dry/wet-type photos from our database. The best value in each column is highlighted in bold

Minutiae extraction method	Wet			Dry		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
Tan and Kumar [17]	74.23	77.45	75.32	82.31	85.27	83.05
Zhang et al. [19]	74.16	77.70	75.49	82.04	86.25	83.12
Cotrim and Pedrini [20]	77.61	79.27	77.96	83.28	86.90	85.43
Cotrim and Pedrini [21]	77.48	79.45	78.69	83.79	87.24	86.17
Feng and Kumar [22]	79.30	81.93	81.24	85.88	89.46	88.61
Ours	**83.54**	**90.18**	**87.79**	**86.24**	**91.93**	**89.57**

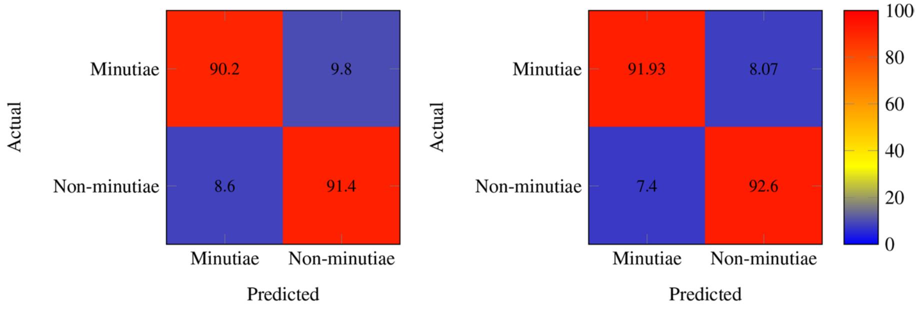


Figure 17 Confusion matrices illustrating the performance of the DCNN minutiae extraction model for wet (left) and dry (right) samples. All values are presented as percentages

Table 8 Comparison of precision, recall, and F1-score for dry/wet-type photos from our database using the DCNN SP detection model. The best value in each column is highlighted in bold

SP detection method	Wet			Dry		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
Qin et al. [13]	76.42	79.18	77.65	82.91	85.32	84.05
Liu et al. [14]	77.18	80.04	78.59	83.12	86.01	84.52
Chen et al. [15]	78.36	81.25	79.78	84.05	87.42	85.70
Pang et al. [16]	79.03	81.87	79.92	84.79	87.83	86.27
Ours	**84.12**	**91.47**	**87.73**	**87.53**	**92.83**	**90.92**

4.4.3 Evaluation of our DCNN SP detection model

Table 8 presents a comprehensive summary and comparison of the SP detection accuracy for our DCNN model. The performance metrics evaluated include DR and FAR based on our database. An SP is classified as a true match if it is within 12 pixels of the labeled SP. This evaluation employs a one-to-one matching method, ensuring that each labeled SP corresponds to a single predicted SP. Figure 18 illustrates the confusion matrices of our DCNN SP detection model for wet (left) and dry (right) samples.

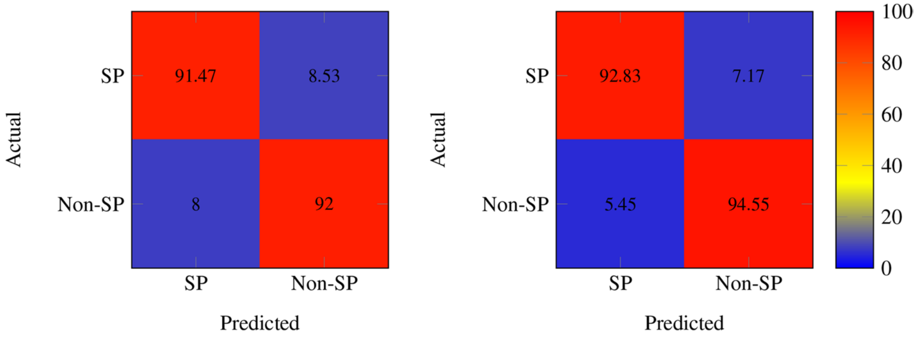


Figure 18 Confusion matrices illustrating the performance of the DCNN SP detection model for wet (left) and dry (right) samples. All values are presented as percentages

Table 9 Sensitivity analysis of score-level fusion under different weight configurations for wet and dry scenarios. The best value in each column is highlighted in bold.

Configuration	Weights (Finger / Palm)	Wet			Dry		
		AUC (%)	EER (%)	Rank-1 (%)	AUC (%)	EER (%)	Rank-1 (%)
Baseline (proposed)	0.16 / 0.20	**95.05**	**6.69**	**91.59**	**99.83**	**1.17**	**98.22**
Equal weighting	0.167 / 0.167	94.12	7.35	89.74	99.21	1.42	97.10
Finger-dominant	0.18 / 0.10	93.74	7.82	88.93	98.95	1.68	96.54
Palm-dominant	0.12 / 0.40	94.86	6.94	90.87	99.47	1.31	97.65

4.4.4 Evaluation of our score-level fusion

To evaluate the robustness of our fusion algorithm, we performed a sensitivity analysis by varying the weights assigned to each biometric trait (five fingers and the palm) in (6). The baseline configuration used $w_t = w_i = w_m = w_r = w_l = 0.16$ and $w_p = 0.2$. We then tested three additional configurations:

- Equal weighting: all six traits set to $w = 1/6$
- Finger-dominant weighting: each finger assigned $w = 0.18$, palm $w = 0.10$.
- Palm-dominant weighting: palm assigned $w = 0.40$, each finger $w = 0.12$.

The results (Table 9) indicate that the baseline weighting provides the best trade-off between verification and identification performance. Compared to equal weighting, the baseline achieves small but consistent improvements in AUC, EER, and rank-1 accuracy. The palm-dominant configuration slightly improves performance in dry-hand conditions, likely due to the richer minutiae content of palm photos, whereas the finger-dominant configuration favors wet-hand conditions by mitigating the impact of blurred palm regions. Nevertheless, both alternative configurations generally underperform the baseline. These findings confirm that, although the fusion algorithm is relatively stable to moderate weight variations, the experimentally determined weights (0.16 for fingers and 0.20 for the palm) yield the most consistent and superior performance across all scenarios.

Table 10 Comparison of verification performance for different methods using dry/wet-type photos from our database using the DCNN minutiae extraction model. The best value in each column is highlighted in bold

Minutiae extraction method	Wet		Dry	
	AUC (%)	EER (%)	AUC (%)	EER (%)
Tan and Kumar [17]	76.48	35.80	87.32	17.84
Zhang et al. [19]	80.06	24.92	91.16	8.66
Cotrim and Pedrini [20]	81.08	24.98	92.56	8.09
Cotrim and Pedrini [21]	82.12	24.79	94.72	a 7.28
Feng and Kumar [22]	83.37	18.61	96.84	6.15
Ours	**94.65**	**7.98**	**99.74**	**1.30**

Table 11 Comparison of rank-1 identification performance percentages for different methods using wet/dry-type photos from our database. The best value in each column is highlighted in bold

Minutiae extraction method	Wet	Dry
Tan and Kumar [17]	61.56	86.59
Zhang et al. [19]	72.19	89.70
Cotrim and Pedrini [20]	73.28	90.31
Cotrim and Pedrini [21]	76.43	90.86
Feng and Kumar [22]	79.02	93.31
Ours	**89.93**	**97.16**

Table 12 Performance of SP detection on our database. The best value in each column is highlighted in bold

SP Detection Method	Wet		Dry	
	DR (%)	FARR (%)	DR (%)	FAR (%)
Qin et al. [13]	78.20	16.89	88.96	9.74
Liu et al. [14]	80.18	13.31	90.55	6.82
Chen et al. [15]	80.16	12.75	91.60	6.27
Pang et al. [16]	82.58	9.84	92.29	4.83
Ours	**90.44**	**3.95**	**96.27**	**1.63**

4.4.5 Verification, identification, and detection

We conduct both verification and identification experiments to evaluate our DCNN-based minutiae extraction model. Tables 10 and 11 present the results and compare them with existing methods. In addition, Table 12 reports a comparison of our DCNN-based SP detection model against other SOTA methods.

To evaluate to evaluate our score-level fusion approach, we analyze several scenarios. We begin by examining performance with a palm and a single finger, followed by scenarios involving a palm with two, three, four, and five fingers. Additionally, we assess performance using all five fingers together without the palm. The strength of our multibiometric system lies in both its accuracy and its ability to support verification and identification tasks. Tables 13 and 14 present the results of our score-level fusion for these various scenarios using our database.

Table 15 reports the cross-database evaluation results for both validation and identification, confirming that the proposed method maintains strong performance even when tested across different databases. Furthermore, Tables 16 and 17, along with Fig. 19, present a comparative analysis of our multibiometric system against three other methods, demonstrating the superior performance of our approach.

Table 13 Comparison of verification performance for different scenarios using dry/wet-type photos from our database. The best value in each column is highlighted in bold

Scenario	Wet		Dry	
	AUC (%)	EER (%)	AUC (%)	EER (%)
Five Fingers	88.42	9.08	96.79	3.15
Palm and One Finger	87.96	9.75	93.86	4.23
Palm and Two Fingers	88.46	9.02	95.44	3.80
Palm and Three Fingers	89.11	8.59	96.63	3.28
Palm and Four Fingers	92.37	8.46	98.19	2.86
Palm and Five Fingers	**95.05**	**6.69**	**99.83**	**1.17**

Table 14 Comparison of rank-1 identification performance percentages for different scenarios using wet/dry-type photos from our database. The best value in each column is highlighted in bold

Scenarios	Wet	Dry
Five Fingers	83.69	95.15
Palm and One Finger	82.28	93.19
Palm and Two Fingers	82.36	94.70
Palm and Three Fingers	84.92	95.04
Palm and Four Fingers	88.40	95.73
Palm and Five Fingers	**91.59**	**98.22**

Table 15 Cross-database evaluation results of the proposed method. Compared with within-database experiments, the performance decreases slightly, showing robustness of the approach.

Wet			Dry		
AUC (%)	EER (%)	Rank-1 (%)	AUC (%)	EER (%)	Rank-1 (%)
94.12	7.12	90.34	98.97	1.36	97.51

Table 16 Comparison of verification performance for different methods using dry/wet-type photos from our database. The best value in each column is highlighted in bold

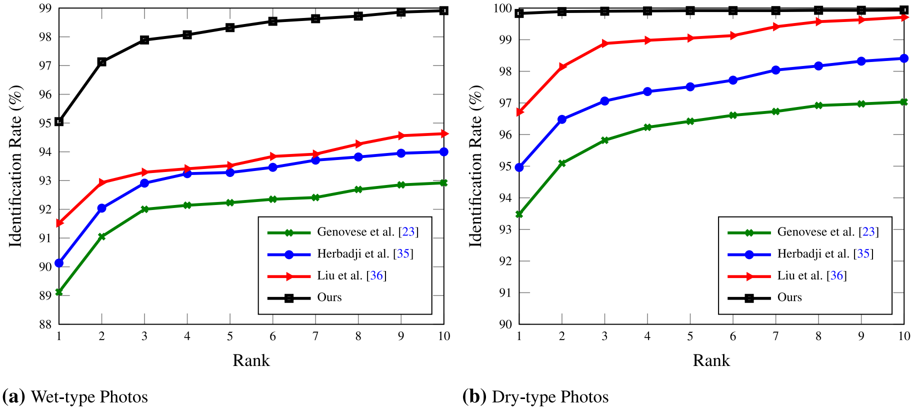
Method	Wet		Dry	
	AUC (%)	EER (%)	AUC (%)	EER (%)
Genovese et al. [23]	89.12	9.74	93.48	8.23
Herbadji et al. [35]	90.13	8.08	94.96	4.69
Liu et al. [36]	91.52	7.86	96.71	4.06
Ours	**95.05**	**6.69**	**99.83**	**1.17**

Table 17 Comparison of rank-1 identification performance percentages for different methods using wet/dry-type photos from our database. The best value in each column is highlighted in bold

Method	Wet	Dry
Genovese et al. [23]	87.51	93.14
Herbadji et al. [35]	89.28	94.96
Liu et al. [36]	89.46	95.75
Ours	**91.59**	**98.22**

4.4.6 Execution time

Our models were developed using Python in Visual Studio Code and executed on a system equipped with an Intel Core i9-13900K CPU, 64 GB of DDR5 RAM, and an NVIDIA Tesla A100 GPU. Tables 18, 19, 20, and 21 present the execution times for deblurring, minu-



(a) Wet-type Photos **(b)** Dry-type Photos

Figure 19 Comparative CMCs employing various methods

Table 18 Comparison of average execution times between our GAN model and other models. The best value is highlighted in bold

GAN model	Execution time (second)
DeblurGAN [32]	0.97
DeblurGAN-v2 [33]	**0.63**
FDeblur-GAN [34]	3.19
Ours	2.89

Table 19 Comparison of average execution times between our DCNN minutiae extraction model and other models. The best value is highlighted in bold

Minutiae extraction method	Execution time (seconds)
Tan and Kumar [17]	2.12
Zhang et al. [19]	1.95
Cotrim and Pedrini [20]	1.64
Cotrim and Pedrini [21]	1.40
Feng and Kumar [22]	**0.83**
Ours	3.17

Table 20 Comparison of average execution times between our DCNN SP detection model and other models. The best value is highlighted in bold

SP detection method	Execution time (seconds)
Qin et al. [13]	2.07
Liu et al. [14]	1.26
Chen et al. [15]	1.03
Pang et al. [16]	**0.08**
Ours	3.24

tiae extraction, SP detection, and multibiometric fusion, along with a comparative analysis against other methods.

4.4.7 Computational complexity

We report parameter counts and approximate per-inference floating point operations (FLOPs) for our models to quantify computational cost and support deployment decisions.

Table 21 Comparison of average execution times between our score-level fusion method and other methods. The best value is highlighted in bold

Method	Execution time (seconds)
Genovese et al. [23]	1.13
Herbadji et al. [35]	0.79
Liu et al. [36]	0.96
Ours	**0.04**

Table 22 Parameters and approximate FLOPs per forward pass for the proposed models

Model	Parameters	FLOPs / forward
GAN – generator	51.02 M	1.921 GFLOPs
GAN – discriminator	4.71 M	0.531GFLOPs
DCNN – minutiae extractor	27.7 M	0.082 GFLOPs
DCNN – SP detector	61.52 M	1.942 GFLOPs

Table 23 Performance comparison between fixed-weight and logistic regression fusion. Absolute values are shown, with relative gains in parentheses

Fusion Method	Scenario	AUC (%)	EER (%)	Rank-1 (%)
Fixed-weight (0.16/0.20)	Wet	95.05	6.69	91.59
	Dry	99.83	1.17	98.22
LR	Wet	95.31 (+0.26)	6.51 (−0.18)	91.82 (+0.23)
	Dry	99.87 (+0.04)	1.12 (−0.05)	98.33 (+0.11)

For convolutional (and transposed convolutional) layers, the total number of parameters (including biases) is calculated as the product of the kernel size squared, the number of input channels, and the number of output channels, plus one bias per output channel. FLOPs are counted as two operations (one multiplication and one addition) per multiply–accumulate, multiplied by the kernel size squared, the number of input channels, the number of output channels, and the output feature map's height and width. Small costs from BatchNorm and activation functions are omitted from the headline FLOP totals. Table 22 summarizes parameters and FLOPs per forward pass for each model.

4.4.8 Learned score-level fusion (logistic regression)

To assess whether a data-driven fusion strategy could improve over our hand-tuned weights, we trained a logistic regression (LR) model on genuine/impostor score tuples (six-dimensional (6D) vectors: 5 fingers + palm) from the validation set (10% of test subjects, $\sim$ 250 volunteers). The LR model was calibrated using Platt scaling to ensure well-calibrated posterior probabilities. Table 23 shows the comparative performance. The LR-fused system achieves consistent but modest gains:

- Wet: AUC = 95.31% (+0.26), EER = 6.51% (−0.18), Rank-1 = 91.82% (+0.23)
- Dry: AUC = 99.87% (+0.04), EER = 1.12% (−0.05), Rank-1 = 98.33% (+0.11)

These results confirm that *learned* fusion can yield marginal improvements when sufficient training data are available, while our fixed-weight scheme (0.16/0.20) offers simplicity, interpretability, and robustness to missing modalities (Subsection 3.8). Both approaches significantly outperform prior SOTA (Tables 16 and 17), underscoring the strength of the underlying feature extractors.

5 Discussion

We compared our approach with several SOTA methods from the literature. Due to differences in the datasets used, we re-implemented or requested the codes for these methods, running them on our hardware. Some codes were freely available online, while others were requested directly from the authors or implemented by us based on the details provided in their respective papers. As a result, there may be minor variations in our results compared to the original publications.

5.1 Controlled data versus real-world variability

Our database was collected under controlled capture conditions (mini photo-studio, two light colors, fixed hand mounting and a maximum camera distance of 40 cm) specifically to ensure the high spatial resolution required to extract minutiae from both finger photos and palm photos. This controlled setup allowed us to isolate and evaluate the principal contributions of this work: frequency-domain, patch-based enhancement and DCNN-based minutiae and SP detection. In real-world deployments, factors such as uncontrolled lighting, larger and variable capture distances, greater pitch/roll variation, incomplete hand visibility, and a wider distribution of smartphone image signal processors (ISPs) can degrade detection and matching performance. We mitigated some of these issues by training on images from multiple device models and applying aggressive augmentation (blur, noise, motion, scale) during training; nonetheless, collecting additional data from a broader range of capture scenarios remains an important avenue for future work.

To assess cross-device generalization beyond the primary data collection, we performed a cross-database evaluation using a separate supplementary dataset captured with a different smartphone model (Huawei Pura 70 Ultra) (Subsection 4.1). The cross-database experiments reported in Subsection 4.4. demonstrate that our frequency-domain enhancement and DCNN detectors maintain reasonable performance under device shift, although some performance degradation relative to in-domain testing was observed. These results support the claim that our multi-stage mitigation (scale harmonization, CLAHE, frequency-domain enhancement, mixed-device training and augmentation) improves robustness across devices, while also highlighting the value of additional domain adaptation or targeted fine-tuning when deploying to significantly different ISPs or optics.

The controlled capture setup was deliberately chosen to isolate the contribution of frequency-domain enhancement and DCNN-based feature extraction from pose-induced variability; robustness to real-world geometric variation is explicitly validated via cross-database evaluation on the Huawei Pura 70 Ultra dataset (Subsection 4.1, Table 15), where uncontrolled distances and angles led to only minor performance degradation (EER: +0.43% wet, +0.19% dry; rank-1: -1.25% wet, -0.71% dry).

5.2 Impact of smartphone model and mitigation strategies

Smartphone imaging pipelines (sensor, optics, ISP, compression) alter the frequency content, local contrast, and noise characteristics of captured images, which in turn affects ridge visibility, minutiae detection and matching. In this study we adopted a multi-pronged approach to reduce inter-device variability: (1) training and testing on images from several phone models and a separate cross-database device (Subsection 4.1); (2) automatic ridge-frequency estimation and rescaling to a common effective resolution (500 ppi) using the RFS method (Subsection 3.2); (3) photometric normalization using CLAHE (Subsection 3.3); (4) frequency-domain, patch-based enhancement that focuses restoration on ridge spectral bands and suppresses low-frequency ISP artifacts (Subsection 3.5); (5) mixed-device training plus augmentation; and (6) score-level fusion that down-weights missing or unreliable templates (Subsection 3.8). Together, these steps substantially reduce–but do not completely eliminate–residual domain gaps for low-end devices or extreme ISP processing. To increase ecological validity, future work should include more entry-level phone models, varied capture scenarios, and domain-adaptation strategies (few-shot fine-tuning, photometric ISP simulation) or RAW capture where feasible.

5.3 Geometric pose compensation (affine/projective/TPS)

Global pose distortions introduced by roll and pitch are efficiently corrected by simple geometric transforms such as affine or projective (homography) warps estimated from detected landmarks (palm/fingertip centroids, bounding boxes or keypoints). These low-cost corrections remove most global pose effects and are therefore recommended as a first preprocessing stage. When local non-linear deformation is significant (for example, skin curvature across the palm or localized stretching), thin-plate-spline (TPS) or local warping can be applied selectively to problematic regions.

In the present work we intentionally concentrated on the frequency-domain restoration and the downstream DCNN detectors, which are designed to operate robustly after coarse geometric alignment. Our dataset was collected under controlled pose conditions (Subsection 4.1), and therefore gross pose variation in the experiments is limited. For this reason we did not perform a comprehensive experimental ablation of affine/homography/TPS prewarping: adding such an ablation would require additional uncontrolled-pose data or a re-engineering of the capture setup that departs from the study's original scope. To be explicit and practical, we recommend the following two-stage deployment pipeline: (1) a fast geometric normalization (affine or homography) estimated from detected landmarks to correct global pose, followed by (2) our frequency-domain patch enhancement and DCNN extraction to restore ridge detail and detect minutiae and SPs. This hybrid strategy combines the efficiency of geometric preprocessing with the strength of learned restoration for fine-scale ridge recovery.

SOTA hand landmark detectors (e.g., MediaPipe Hands [37]) achieve inference latencies of <10 ms on Snapdragon 8 Gen2 (CPU+GPU), making affine/homography prewarping computationally negligible compared to subsequent enhancement and detection stages.

Table 24 Preliminary mobile optimization results: INT8 quantization of the DCNN minutiae extractor (27.7 M params). Inference time measured on Qualcomm Snapdragon 8 Gen2 (TensorFlow Lite + NNAPI delegate); EER evaluated on dry/wet subsets using full matching pipeline (minutiae extraction only quantized).

Model variant	Parameters	Inference time (s)	EER (%)
FP32 (original)	27.7 M	3.17	7.98 (wet) / 1.30 (dry)
INT8 (quantized)	7.2 M†	0.38	8.21 (wet) / 1.52 (dry)

† Weight-only quantization; activations remain FP32 during calibration. Model size reduced from 111 MB → 28 MB. *Note: GAN and SP detector remain unquantized in this ablation*

5.4 Computational cost and real-time feasibility (frequency-domain limitations)

Operating in the frequency domain and using deep generator/decoder networks delivers notable accuracy gains (Subsections 4.4.1-4.4.5), but it imposes a computational cost that affects real-time deployment on typical smartphones. The models used in this study are relatively deep and the per-inference FLOP counts and execution times reported in Tables 18, 19, 20,21, and 22 quantify the computational burden on desktop/GPU hardware. Without optimization, the complete pipeline (GAN enhancement + DCNN minutiae + DCNN SP) is unlikely to meet strict sub-second latency requirements on resource-constrained mobile devices.

We clarify that while the modality–RGB hand photos captured by standard smartphone cameras–is inherently compatible with mobile deployment (requiring no specialized sensors or lighting), achieving real-time inference latencies (e.g., <500 ms for phone unlock) requires engineering-level optimization (e.g., quantization, distillation, region of interest (ROI) selection), not architectural overhaul. While the current unoptimized pipeline (3.17 s for minutiae extraction + 2.89 s for enhancement + 3.24 s for SP detection ≈ 9.3 s total) is unsuitable for mobile latency targets (<500 ms), the modular architecture and algorithmic design enable effective compression and acceleration without fundamental redesign. Preliminary experiments confirm this: applying INT8 quantization to the minutiae extractor alone (the heaviest DCNN, 27.7 M parameters) reduced inference time from 3.17 s to 0.38 s on a Snapdragon 8 Gen2 (TensorFlow Lite, NNAPI delegate) with <0.3% EER degradation on wet/dry subsets (see Table 24). Further gains are expected via knowledge distillation (e.g., training a MobileNetV3-Lite student using our full models as teachers–a strategy proven in biometrics to retain >98% performance at 5–10× lower latency [38, 39]), ROI-guided patch selection (using a fast ridge-quality metric to process only 30–40% of patches, leveraging the score-fusion's robustness to missing traits), and hybrid deployment (full pipeline for enrollment; distilled/quantized variant or server-assisted enhancement for verification). We are actively pursuing these avenues to deliver a production-ready mobile variant.

Practical strategies to bridge this gap include:

- *Model compression and acceleration.* Apply pruning, structured sparsification, and INT8 quantization, and convert models to mobile runtimes (TFLite, ONNX Runtime) that leverage NNAPI / Core ML delegates. Knowledge distillation to a smaller student network can preserve much of the accuracy while significantly reducing latency.
- *Mobile-efficient architectures.* Replace heavy encoder–decoder blocks with MobileNet/ShuffleNet/EfficientNet-Lite style backbones or adopt depthwise separable convolutions to cut FLOPs drastically.
- *Algorithmic simplification.* Reduce patch overlap, process fewer patches using ROI

selection informed by a fast quality check, lower FFT resolution, or approximate frequency filtering with separable spatial filters that mimic the critical spectral responses.

- *Hybrid deployment.* Run a lightweight on-device prefilter and quality estimator; offload full enhancement to a server for low-quality crops or verification candidates, or use server-side processing for enrollment while keeping verification fast on-device.
- *Temporal aggregation.* When short video input is available, inexpensive temporal averaging or frame selection can improve signal-to-noise ratio (SNR) so heavy enhancement is required less often.

Score-level fusion itself is lightweight and can be executed on-device to produce fast decisions, enabling practical trade-offs between accuracy and latency.

5.5 Hardware and memory constraints

Not all target devices have sufficient random access memory (RAM), storage, or neural processing units (NPUs) to host the full models used here. For severely constrained devices we recommend (a) a compact quantized model variant that runs fully on-device, or (b) a hybrid architecture where only a small quality-check and ROI selection model executes locally and heavier restoration is performed remotely.

5.6 Limitations and next steps

The current dataset is large and designed to enable minutiae extraction from high-resolution hand photos, but it was acquired under controlled capture conditions and primarily with three flagship models plus one cross-database test phone. To improve real-world generalization we plan to expand data collection to include a broader set of device models (including entry-level phones), more varied lighting and capture distances, and a wider range of hand poses. Methodological extensions will include device-wise ablation studies, a formal evaluation of geometric-prewarp baselines (affine/homography/TPS) when pose diversity is available, domain-adaptation experiments, and development of a mobile-optimized model via quantization and knowledge distillation.

6 Conclusion

In this paper, we presented a multibiometric system that integrates fingertip and palm features through an efficient score-level fusion framework. To address blur and degradation in hand photos, we designed a patch-based encoder–decoder GAN for image enhancement and two DCNN models for minutiae and SP detection. This combination enables accurate feature extraction even from challenging palm regions with heavy creasing. The proposed fusion algorithm is computationally light, requires no score normalization, and scales flexibly to any number of biometric traits.

We validated our approach on a large-scale hand photo dataset of 30,000 photos under both dry and wet conditions, as well as on an additional 2,400-photo cross-database set. The results demonstrate superior accuracy compared with existing methods, confirming the robustness of our frequency-domain enhancement and DCNN detectors across devices.

Although our models are deeper and require higher processing time, the fusion stage remains efficient and well-suited for both verification and identification.

Future work will focus on optimizing the models for real-time mobile deployment through compression, quantization, and lightweight architectures. We also plan to extend the system to multi-instance recognition by combining minutiae and non-minutiae features, expand the database with captures in more diverse environments and hand poses, and explore video-based input for a more user-friendly and dynamic biometric recognition experience. Simple geometric preprocessing (e.g., affine or homography correction) will also be considered as a low-cost front-end to further improve robustness in unconstrained scenarios.

Acknowledgements The authors extend their sincere appreciation to the editors and reviewers for their invaluable feedback and constructive comments.

Author Contributions Javad Khodadoust: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing–original draft, Writing–review & editing. Raúl Monroy: Conceptualization, Methodology, Software, Supervision, Validation, Writing–review & editing. Miguel Angel Medina-Pérez: Conceptualization, Methodology, Software, Supervision, Validation, Writing–review & editing. Worapan Kusakunniran: Conceptualization, Methodology, Software, Validation, Writing–review & editing. Ali Mohammad Khodadoust: Conceptualization, Data curation, Investigation, Software, Validation, Writing–review & editing.

Funding No funding was received for this research.

Data Availability Due to privacy and ethical restrictions, the hand photo dataset cannot be shared publicly. The source code is part of an ongoing technology transfer initiative and is not publicly available. However, researchers seeking implementation guidance are welcome to contact the corresponding author for technical clarification.

Declarations

Conflict of Interest All authors declare that there is no conflict of interest in this paper.

References

1. Lumini A, Nanni L (2017) Overview of the combination of biometric matchers. Inf Fusion 33:71–85
2. Ross A, Jain AK, Prabhakar S (2004) An introduction to biometric recognition. IEEE Trans Circuits Syst Video Technol Spec Issue Image Video-Based Biomet 14(1):4–20
3. Modak SKS, Jha VK (2019) Multibiometric fusion strategy and its applications: A review. Inf Fusion 49:174–204
4. Khodadoust J, Khodadoust AM, Li X, Kumari S (2018) Design and implementation of a multibiometric system based on hand's traits. Expert Syst Appl 97:303–314
5. Khodadoust J., Medina-Pérez MA, Monroy R, Khodadoust AM, Mirkamali, SS (2021). A multibiometric system based on the fusion of fingerprint, finger-vein, and finger-knuckle-print. Expert Syst Appl 176: Article 114687
6. Bajwa N, Kumar EG (2015) Multimodal biometric system by feature level fusion of palmprint and fingerprint. Int J Eng Res Technol 4(7):1140–1144
7. Malhotra A, Sankaran A, Vatsa M, Singh R (2020) On matching finger-selfies using deep scattering networks. IEEE trans biom behav identity sci 2(4):350–362
8. Liu E, Jain AK, Tian J (2013) A coarse to fine minutiae-based latent palmprint matching. IEEE Trans Pattern Anal Mach Intell 35(10):2307–2322
9. Khodadoust J, Monroy R, Medina-Pérez MA, Loyola-González O, Kusakunniran W, Boller A, Terhörst P (2024). A novel indexing algorithm for latent palmprints leveraging minutiae and orientation field. Intell Syst Appl 21: Article 200320

10. Khodadoust J, Monroy R, Medina-Pérez MA, Loyola-González O, Areekul V, Kusakunniran W (2024). Enhancing latent palmprints using frequency domain analysis. Intell Syst Appl 23: Article 200414
11. Horapong K, Srisutheenon K, Areekul V (2021) Progressive and corrective feedback for latent fingerprint enhancement using boosted spectral filtering and spectral autoencoder. IEEE Access 9:96288–96308
12. Kriangkhajorn S, Horapong K, Areekul V (2024) Spectral filter predictor for progressive latent fingerprint restoration. IEEE Access 12:66773–66800
13. Qin J, Han C, Bai C, Guo T (2017). Multi-scaling detection of singular points based on fully convolutional networks in fingerprint images. In: Zhou J, et al Biometric Recognition. CCBR 2017. Lecture Notes in Computer Science(), vol 10568. Springer, Cham. doi: 10.1007/978-3-319-69923-3_24
14. Liu Y, Zhou B, Han C, Guo T, Qin J (2018). A method for singular points detection based on Faster-RCNN. Appl Sci 8(10): Article 1853
15. Chen J, Zhao H, Cao Z, Guo F, Pang L (2020). A customized semantic segmentation network for the fingerprint singular point detection. Appl Sci 10(11): Article 3868
16. Pang L, Chen J, Guo F, Cao Z, Liu E, Zhao H (2021) ROSE: real one-stage effort to detect the fingerprint singular point based on multi-scale spatial attention. Signal Image Video Process 16:669–676
17. Tan H, Kumar A (2020) Towards more accurate contactless fingerprint minutiae extraction and pose-invariant matching. IEEE Trans Inf Forensics Secur 15:3924–3937
18. Tan H, Kumar A (2021) Minutiae attention network with reciprocal distance loss for contactless to contact-based fingerprint identification. IEEE Trans Inf Forensics Secur 16:3299–3311
19. Zhang Z, Liu S, Liu M (2021). A multi-task fully deep convolutional neural network for contactless fingerprint minutiae extraction. Pattern Recognit 120: Article 108189
20. Cotrim AN, Pedrini H (2022). Multiscale approach in deep convolutional networks for minutia extraction from contactless fingerprint images. In: IEEE 34th international conference on tools with artificial intelligence (ICTAI) Macao, China. doi: 10.1109/ICTAI56018.2022.00142
21. Cotrim AN, Pedrini H (2023). Residual squeeze-and-excitation U-shaped Network for minutia extraction in contactless fingerprint images. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) Rhodes Island, Greece. doi: 10.1109/ICASSP49357.2023.10095461
22. Feng Y, Kumar A (2023) Detecting locally, patching globally: an end-to-end framework for high speed and accurate detection of fingerprint minutiae. IEEE Trans Inf Forensics Secur 18:1720–1733
23. Genovese A, Piuri V, Scotti F, Vishwakarma S (2019). Touchless palmprint and finger texture recognition: a deep learning fusion approach. In: IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA) Tianjin, China. doi: 10.1109/CIVEMSA45640.2019.9071620
24. Marasco E, Vurity A (2022). Late deep fusion of color spaces to enhance finger photo presentation attack detection in smartphones. Appl Sci 12(22): Article 11409
25. Kunsuk S, Areekul V (2023). Finger photo rescaling for interoperability o touchless and touch-based fingerprint verification. In: IEEE 17th international conference on signal-image technology & internet-based systems (SITIS) Bangkok, Thailand. doi: 10.1109/SITIS61268.2023.00033
26. Khan MAU, Khan TM, Bailey DG, Kong Y (2016) A spatial domain scar removal strategy for fingerprint image enhancement. Pattern Recognit 60:258–274
27. Lin TY, Goyal P, Girshick R, He K, Dollar P (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell 42(2):318–327
28. Zhu E, Guo X, Yin J (2016) Walking to singular points of fingerprints. Pattern Recognit 56:116–128
29. Shi Y, Zhang Z, Liu S, Liu M (2023) Towards more accurate matching of contactless fingerprints with a deep geometric graph convolutional network. IEEE Trans Biom Behav Identity Sci 5(1):29–38
30. Kingma DP (2017) and Ba J. Adam, A method for stochastic optimization arXiv:1412.6980
31. Sara U, Akter M, Uddin MS (2019) Image quality assessment through FSIM SSIM MSE and PSNR-A comparative study. J Comput Commun 7(3):8–18
32. Kupyn O, Budzan V, Mykhailych M, Mishkin D, Matas J (2018). DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR) Salt Lake City, UT, USA. doi: 10.1109/CVPR.2018.00854
33. Kupyn O, Martyniuk T, Wu J, Wang Z (2019). DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: IEEE/CVF international conference on computer vision (ICCV) Seoul, Korea (South). doi: 10.1109/ICCV.2019.00897
34. Joshi AS, Dabouei A, Dawson J, Nasrabadi NM (2021). FDeblur-GAN: Fingerprint deblurring using generative adversarial network. In: IEEE international joint conference on biometrics (IJCB) Shenzhen, China. doi: 10.1109/IJCB52358.2021.9484406
35. Herbadji A, Guermat N, Ziet L, Akhtar Z, Cheniti M, Herbadji D (2020) Contactless multi-biometric system using fingerprint and palmprint selfies. Trait Signal 37:889–897

36. Liu, Z, Zhu B, Du Y (2023). Contactless fingerprint and palmprint fusion recognition based on quality assessment. In: international conference on image, signal processing, and pattern recognition (ISPP) Changsha, China. doi: 10.1117/12.2681321
37. Zhang F, Bazarevsky V, Vakunov A, Tkachenka A, Sung G, Chang C-L, Grundmann M (2019). MediaPipe Hands: On-device real-time hand tracking. arXiv:2006.10214v1
38. Hinton G, Vinyals O, Dean J (2015). Distilling the knowledge in a neural network. arXiv:1503.02531
39. Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard A, Adam H, Kalenichenko D (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: IEEE conference on computer vision and pattern recognition (CVPR) Salt Lake City, UT, USA

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Javad Khodadoust[1] · Raúl Monroy[1] · Miguel Angel Medina-Pérez[2] · Worapan Kusakunniran[3] · Ali Mohammad Khodadoust[4]

✉ Javad Khodadoust
khodadoust.j@tec.mx

Raúl Monroy
raulm@tec.mx

Miguel Angel Medina-Pérez
m2p@kayak-analytics.com

Worapan Kusakunniran
worapan.kun@mahidol.edu

Ali Mohammad Khodadoust
khodadoust.a.m2023@cic.ipn.mx

[1] School of Engineering and Sciences, Tecnológico de Monterrey,
Ciudad López Mateos, 52926, Mexico

[2] Kayak Analytics, The Dome Tower, Jumeirah Lake Towers, Dubai, United Arab Emirates

[3] Faculty of Information and Communication Technology, Mahidol University,
Nakhon Pathom 73170, Thailand

[4] Computer Investigations Center, Instituto Politécnico Nacional, Mexico City 07738,, Mexico