IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 March 2024, accepted 12 April 2024, date of publication 16 April 2024, date of current version 25 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3389701

RESEARCH ARTICLE

Exploring the Power of Simplicity: A New State-of-the-Art in Fingerprint Orientation Field Estimation

RAFFAELE CAPPELLI
Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy
e-mail: raffaele.cappelli@unibo.it

ABSTRACT Fingerprint recognition is an important tool for personal identification due to its versatility, user-friendliness, and accuracy. Fingerprint orientation field estimation, a crucial step in fingerprint feature extraction, significantly impacts recognition performance. While numerous methods have been proposed, achieving state-of-the-art accuracy often comes at the cost of increased complexity, hindering practical implementation. This work addresses this challenge by introducing two novel, simple, and efficient fingerprint orientation field estimation methods: GBFOE and SNFOE. Both methods adhere to the KISS (Keep It Simple and Straightforward) principle, achieving remarkable performance on publicly available benchmarks. GBFOE outperforms all local methods and rivals more complex approaches, while SNFOE establishes itself as a new state-of-the-art, achieving the highest accuracy on all datasets in both evaluated benchmarks. Surpassing methods designed specifically for latent fingerprints, SNFOE demonstrates exceptional performance on this challenging task within the evaluated benchmarks, highlighting its generalizability despite not being trained on such data. These results underline the potential of simple and efficient methods in fingerprint orientation field estimation, paving the way for practical and resource-efficient fingerprint recognition systems. An open-source Python implementation of both methods is available, fostering further research and development in this field.

INDEX TERMS Fingerprints, orientation field estimation, KISS principle, FOE benchmarks, NIST SD27.

I. INTRODUCTION

Fingerprint recognition, initially employed solely within the criminal justice system, has undergone a significant evolution, becoming a widely implemented feature in contemporary society. Its versatility, user-friendliness, and accuracy have made it a cornerstone of many applications, revolutionizing the realm of personal identification [1], [2].

Within the fingerprint recognition process, orientation field estimation (Fig. 1) plays a pivotal role in feature extraction. Accurate orientation information is essential for subsequent tasks, such as fingerprint enhancement. Conversely, deviations between the estimated and actual orientation field can lead to the extraction of incorrect or spurious fingerprint features, including minutiae with erroneous position or direction, or false minutiae. These errors can significantly degrade the performance of fingerprint recognition systems.

Over the past five decades, numerous fingerprint orientation field estimation techniques have emerged, ranging from traditional methods utilizing handcrafted local features in the spatial or frequency domains, to more recent approaches leveraging sophisticated global mathematical models and machine learning. While these advancements have demonstrably enhanced estimation accuracy, particularly for low-quality and latent fingerprints, they have also introduced a layer of complexity that can impede their practical implementation and adoption. This study adheres to the KISS (Keep It Simple and Straightforward[1])

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Jin.

[1]While other interpretations of the KISS acronym are more common, this paper employs a less colloquial but more academically appropriate definition in the interest of maintaining scientific precision and avoiding potential misinterpretations.

principle [3], [4] by proposing fingerprint orientation field estimation methods that achieve state-of-the-art performance while maintaining simplicity.

The primary contributions of this paper are as follows:

1. A novel learning-based fingerprint orientation estimation method that surpasses all previous methods evaluated on public benchmarks and can deal both with plain fingerprints acquired through online sensors, and with latent fingerprints, without requiring any fine-tuning.
2. A simple fingerprint orientation estimation method based on traditional image processing techniques with minimal computational resource requirements that achieves performance comparable to much more complex methods.
3. An open-source implementation of both methods to facilitate further research and development in this domain.

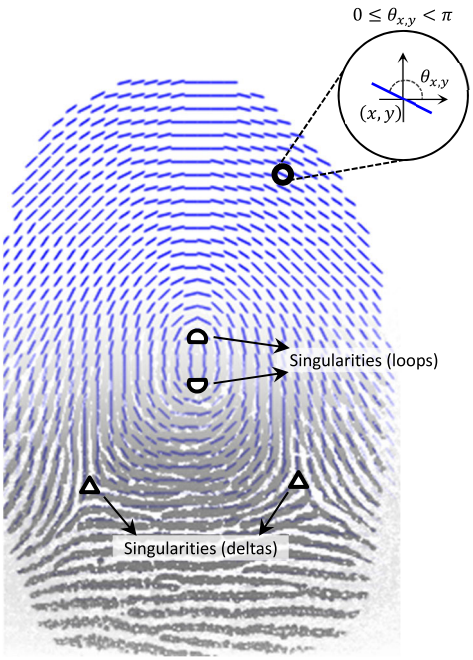

FIGURE 1. A fingerprint faded into its orientation field, with the four singularity regions (two loops and two deltas) highlighted. For clarity, the orientation field is downsampled to $1/64^{th}$ of its original resolution, with line segments indicating local ridge orientations every eight pixels. Each orientation $\theta_{x,y}$ is an angle in $[0, \pi)$.

The rest of this paper is organized as follows. Section II reviews the main fingerprint orientation field estimation methods proposed in the literature. Section III and IV describe the two novel orientation estimation methods. Section V reports experiments aimed at evaluating the performance of the proposed methods and comparing them to the state-of-the-art on public benchmarks. Finally, section VI draws some concluding remarks.

II. RELATED WORKS

Over the past half-century, a diverse spectrum of over 100 fingerprint orientation field estimation methods have been proposed, reflecting the challenging nature and profound importance of this task [1], [5].

Fingerprint orientation field estimation algorithms can be broadly classified into three categories: *local methods*, *global methods*, and *learning-based*.

Local methods analyze image information within a small region around a pixel to determine its orientation. The simplest and most common are gradient-based approaches: since the gradient phase angle denotes the direction of the maximum intensity change at each pixel, the ridge-line orientation can be assumed to be orthogonal to that angle [6], [7], [8], [9], [10], [11]. Other local estimation methods include slit-based techniques, which analyze the pixel intensities along a set of directions (slits) and choose the best orientation according to some measures [12], [13], [14], [15], [16], [17], [18]. Finally, some local estimation methods apply a set of directional filters in the spatial domain [19], [20], [21], while others work in the frequency domain [22], [23], [24]. These methods rely on local information and can generate noisy outputs when faced with low-quality fingerprints. To mitigate this, a common practice is to incorporate a local averaging or voting operator within the processing pipeline. For instance, in [25], the size of the smoothing window was chosen according to a hierarchical coherence analysis. Other multi-scale approaches can be found in [11] and [15]. In [26], a nonlinear approach was proposed to reduce the impulsive noise caused by scars and breaks in the ridges. In [27], a neural network was trained to detect wrong orientation elements, which are then corrected using the orientation of the neighbouring elements. In [28], interpolation with a Delaunay triangulation was used to recover the orientation in noisy regions.

Global methods aim to overcome the limitations of local analysis, leveraging the fact that the orientation field exhibits global consistency, except for a few singularity regions (Fig. 1). Recognizing the inherent smoothness of fingerprint orientation fields, researchers have thus explored utilizing mathematical models to infer local structure while simultaneously imposing a global constraint on each point within the orientation field, resulting in a more holistic and accurate representation. One of the first global models was proposed by Sherlock and Monro [29], where singular points (loops and deltas) are modeled as zeros and poles in the complex plane (zero-pole model). Building upon this model, researchers explored various other mathematical models based on the positions of the singularities. A piecewise linear approximation was proposed in [30] to improve the model near to the singularities. A rational complex model, which generalizes the zero-pole model by adding some pseudozeros and pseudopoles as control points for increased flexibility, was described in [31], [32], and [33]. Other models were proposed in [34], focusing on prediction capabilities where no ridge information is available, and in [35] and [36], using quadratic differentials. A key challenge for the above global methods lies in the reliable detection of singular points in low-quality fingerprints. To address this issue, researchers proposed models that are independent of singularity positions. These models estimate parameters

through data fitting, starting with an initial, coarse estimation of fingerprint orientations. One such approach, presented in [37], utilizes a 2D Fourier series expansion (FOMFE) in the phase plane. This model is able to capture global features, including the high curvature regions surrounding singularities. An enhanced version, introduced in [38], allows for the reconstruction of the global orientation field even from a partial fingerprint. Subsequent refinements (see [39], [40]) sought to further improve reconstruction quality in areas of high curvature and low quality. Beyond the FOMFE approach, alternative models employed various mathematical bases: Legendre polynomials [41], [42], Discrete Cosine Transform [43], Chebyshev polynomials [44], and orthogonal polynomials in two discrete variables [18], [45]. Finally, some authors proposed diffusion models to regularize the coarse local orientations. In [46], a variational framework was proposed to model the orientations and singularity positions. In [47], a div-curl model was introduced to minimize the combination of weighted divergence, curl, and a data fidelity term, yielding a nonlinear partial differential equation for orientation modelling.

One of the earliest learning-based methods was proposed in [48], where a hierarchical neural network was trained to estimate the orientation of 16×16 fingerprint patches. Another approach, described in [49], relies on a Markov Random Field with two components: one incorporates a global mixture model of orientation fields learned from training fingerprints, while the other enforces a smoothness constraint. Reference [50] introduced the Active Fingerprint Ridge Orientation Model (AFROM), a statistical model that constrains fingerprint orientation fields to realistic variations learned from a training set. AFROM employs a vectorially linear regression using Legendre Polynomials. An adaptive approach, which relies on local estimation but replaces orientations in bad quality regions with those provided by a learning-based technique like AFROM, was proposed in [44].

Inspired by spelling correction techniques used in natural language processing, Feng et al. [51] introduced a global orientation patch dictionary. This dictionary consists of typical real orientation patches and is designed to correct the noisy orientation field estimated by a local method. Subsequently, several variants of this approach were introduced, including multi-scale dictionaries [52], dictionaries combined with sparse coding [53], dictionaries of ridge-line patches instead of orientation patches [54], localized orientation patch dictionaries with smaller patch sizes instead of a global dictionary [55], and even a large dictionary of entire orientation fields rather than patches [56].

More recent methods are mostly based on deep learning. In [57], a convolutional neural network (CNN) was trained to classify the orientation field of a fingerprint patch into one of 128 reference orientation patches created by unsupervised clustering of a database of high-quality patches. Reference [58] described a unified fingerprint processing framework implemented by a single network that includes fingerprint segmentation, orientation field estimation, fingerprint enhancement, and minutiae extraction. Reference [59] introduced a regression convolutional neural network that can estimate the orientation field of an entire fingerprint. A modified version of the same network, based on classification with soft fusion of output labels (deep expectation), was reported to achieve better results in [60]. Other convolutional neural networks for orientation field estimation were described in [61] and [62]. Finally, [63] introduced a rather elaborated technique for orientation field estimation, based on residual learning with prior knowledge of fingerprint patterns. Statistical distribution models of orientation fields, for different fingerprint patterns, are obtained by clustering a large database of good-quality fingerprints. The residual orientation fields and reliability scores, which indicate the level of consistency with each statistical orientation model, are estimated using a convolutional neural network. Then the final orientation field is obtained by fusing the estimations according to their corresponding reliability scores.

Analyzing the orientation estimation methods proposed over the years and considering the performance of some of them measured against public benchmarks (see section V), some observations can be made. Local methods remain a valid option for medium to high-quality fingerprints, as they can be implemented efficiently and run even on hardware with limited computational resources. Global methods, despite high expectations, have not proven sufficiently effective on low-quality fingerprints. Global models based on singularity positions are limited by the difficulty of identifying singularities, while similarity-independent approaches often become mere approximations lacking specific a-priori knowledge about fingerprints. For low-quality fingerprints, learning-based methods appear to be the most promising choice. From these considerations, the idea emerges to explore two novel approaches in this study:

- A local method that is as simple as possible yet achieves performance comparable to more complex local methods and, ideally, even global methods.
- A deep-learning-based method that achieves state-of-the-art accuracy while being much simpler than recently proposed alternatives.

III. GBFOE

GBFOE (Gradient-Based Fingerprint Orientation Estimation) is the result of a research effort aimed at designing a local orientation estimation method based on traditional image processing techniques and inspired by the KISS principle. In particular, its development was guided by the following design goals:

- A short sequence of well-known and efficient image processing steps.
- Estimation of orientations at each pixel.
- A minimal number of parameters to configure.

- Adaptation to the fingerprint quality, but without relying on complex regularization techniques or global models.

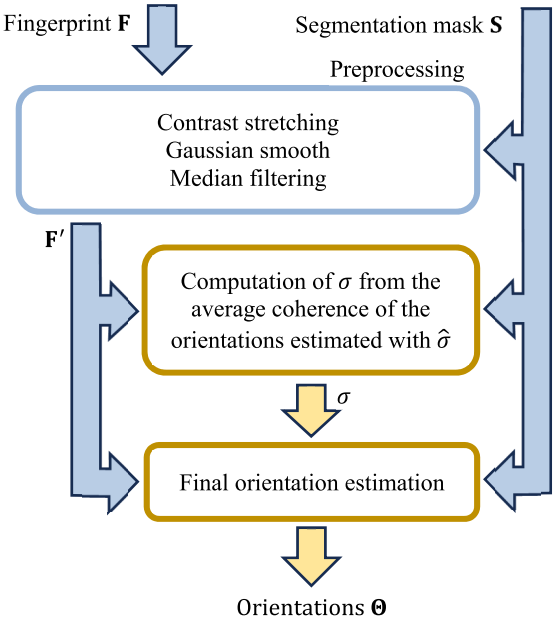

FIGURE 2. A visual summary of the proposed GBFOE method.

Fig. 2 illustrates the functional schema of GBFOE, while Fig. 3 provides an example of its application to a fingerprint sample. Given a fingerprint $\mathbf{F}$ and its segmentation mask $\mathbf{S}$ (a binary image where background pixels have a value of zero and foreground pixels have a value of one [64]), the first step of the method involves the following preprocessing operations aimed at reducing fingerprint noise: 1) contrast stretching, clipping $s_p\%$ values at both ends of the gray level histogram, 2) Gaussian smoothing through convolution with a 2D Gaussian filter with standard deviation σ_p, 3) Median filtering applied with a filter size of m_p. Let $\mathbf{F}'$ be the fingerprint image obtained after preprocessing:

$$\mathbf{F}' = \text{preprocessing}\left(\mathbf{F}, \mathbf{S}, s_p, \sigma_p, m_p\right) \tag{1}$$

The orientation estimation is based on the computation of the local gradients. Let $\mathbf{G}_x = \frac{\partial \mathbf{F}'}{\partial x}$ and $\mathbf{G}_y = \frac{\partial \mathbf{F}'}{\partial y}$ be the two matrices which, for each pixel, contain the horizontal and vertical derivative approximations of $\mathbf{F}'$. $\mathbf{G}_x$ and $\mathbf{G}_y$ can be computed by convolution with the Sobel filters $\mathbf{S}_x$ and $\mathbf{S}_y$ [65]: $\mathbf{G}_x = \mathbf{F}' * \mathbf{S}_x$, $\mathbf{G}_y = \mathbf{F}' * \mathbf{S}_y$.

The gradients computed as described above reflect the orientation of ridges and valleys at a very fine a scale and are highly sensitive to noise in the fingerprint image. To address this, it is necessary to average the gradients. But directly averaging gradient vectors in a local neighborhood poses a challenge: opposite gradient vectors cancel each other out, even though they represent the same ridge/valley orientation. In [6], a solution was proposed: before averaging, double the angles of the gradient vectors. By doing so, opposite gradient vectors align in the same direction and reinforce each other, while perpendicular gradients cancel out. After averaging, the gradient vectors are converted back to their single-angle representation. The local orientation can then be estimated as orthogonal to the angle of the average gradient vector. Additionally, as suggested in [10], the magnitude of the gradient is also squared, to enhance the contribution of stronger gradients to the average. Conceptually, this operation treats the two components of the gradient as the real and imaginary parts of a complex number: squaring the complex number results in doubling its phase angle and squaring its magnitude. Let $\mathbf{G}_{s,x}$ and $\mathbf{G}_{s,y}$ be the two matrices containing the two components of the squared gradients. $\mathbf{G}_{s,x}$ and $\mathbf{G}_{s,y}$ can be computed as follows (see [10]):

$$\mathbf{G}_{s,x} = \mathbf{G}_{xx} - \mathbf{G}_{yy}, \mathbf{G}_{s,y} = 2 \cdot \mathbf{G}_{xy} \tag{2}$$

with $\mathbf{G}_{xx} = \mathbf{S} \odot \mathbf{G}_x^{\circ 2}$, $\mathbf{G}_{yy} = \mathbf{S} \odot \mathbf{G}_y^{\circ 2}$, and $\mathbf{G}_{xy} = 2\left(\mathbf{S} \odot \mathbf{G}_x \odot \mathbf{G}_y\right)$.

$\mathbf{X} \odot \mathbf{Y}$ denotes elementwise matrix multiplication (Hadamard product) and $\mathbf{X}^{\circ 2}$ denotes elementwise matrix square (second Hadamard power). Note that the matrices are also multiplied (elementwise) by the segmentation mask $\mathbf{S}$: this step is taken to discard gradient information from background pixels.

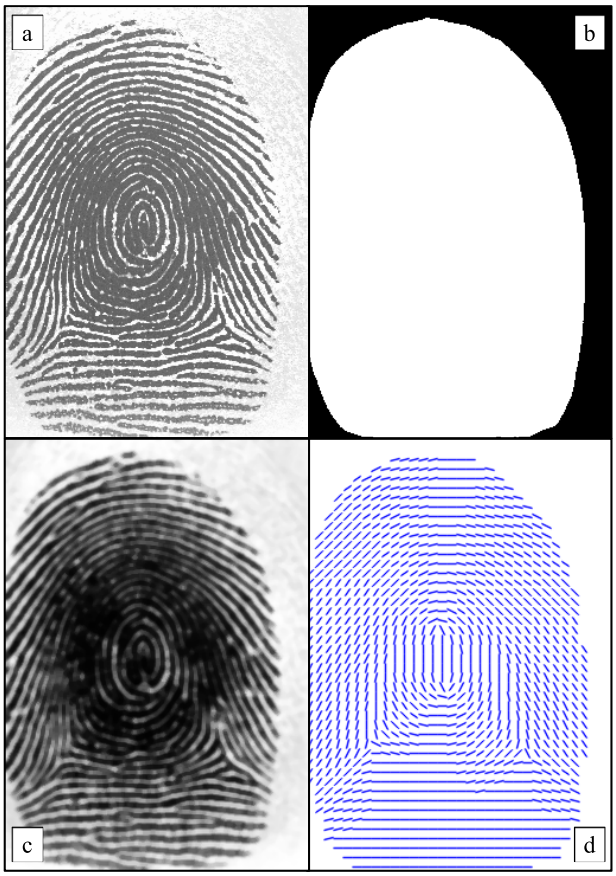

FIGURE 3. An example of orientation field estimation using GBFOE: (a) Fingerprint image F, (b) Segmentation mask S, (c) Preprocessed fingerprint F′, (d) Orientation field Θ, with line segments indicating orientations every eight pixels.

GBFOE averages the squared gradients by convolution with a 2D Gaussian filter:

$$\overline{\mathbf{G}_{s,x}} = \overline{\mathbf{G}_{xx}} - \overline{\mathbf{G}_{yy}}, \overline{\mathbf{G}_{s,y}} = 2\cdot\overline{\mathbf{G}_{xy}} \quad (3)$$
$$\text{with } \overline{\mathbf{G}_{xx}} = \mathbf{G}_{xx} * \mathbf{G}_\sigma, \overline{\mathbf{G}_{yy}} = \mathbf{G}_{yy} * \mathbf{G}_\sigma, \overline{\mathbf{G}_{xy}} = \mathbf{G}_{xy} * \mathbf{G}_\sigma$$

where $\mathbf{G}_\sigma$ is a 2D Gaussian filter with standard deviation σ. From the averaged square gradients, each element $\mathbf{\Theta}_{i,j}$ of the orientation field $\mathbf{\Theta}$ can be then computed as follows:

$$\mathbf{\Theta}_{i,j} = \frac{1}{2}\text{atan2}\left(2\cdot\left(\overline{\mathbf{G}_{xy}}\right)_{i,j}, \left(\overline{\mathbf{G}_{xx}} - \overline{\mathbf{G}_{yy}}\right)_{i,j}\right) + \frac{\pi}{2} \quad (4)$$

The orientation estimation is actually performed twice by GBFOE. Initially, a default value $\hat{\sigma}$ (a parameter of the method) is used for the standard deviation of the Gaussian filter. Subsequently, the local orientation coherence (see [10]) is computed using the following formula:

$$\hat{\mathbf{C}} = \left(\left(\overline{\mathbf{G}_{xx}} - \overline{\mathbf{G}_{yy}}\right)^{\circ 2} + 4\cdot\left(\overline{\mathbf{G}_{xy}}\right)^{\circ 2}\right)^{\circ\frac{1}{2}} \oslash \left(\overline{\mathbf{G}_{xx}} + \overline{\mathbf{G}_{yy}}\right) \quad (5)$$

where $\mathbf{X}^{\circ\frac{1}{2}}$ denotes elementwise matrix square root (Hadamard square root) and $\mathbf{X} \oslash \mathbf{Y}$ denotes elementwise matrix division (Hadamard division).

The local orientation coherence $\hat{\mathbf{C}}$ is then averaged (only over foreground pixels from the segmentation mask $\mathbf{S}$) and used to determine the standard deviation of the Gaussian filter for the final estimation.

$$\sigma = k_\sigma \cdot \left(1 - \frac{\sum_{i,j}\left(\mathbf{S}_{i,j}\cdot\hat{\mathbf{C}}_{i,j}\right)}{\sum_{i,j}\mathbf{S}_{i,j}}\right) \quad (6)$$

where k_σ is a parameter of the method.

This approach leverages the observation that low-quality fingerprints exhibit lower coherence. By employing equation (6), GBFOE adjusts the standard deviation σ, effectively averaging gradients with a wider Gaussian kernel for low-quality fingerprints compared to high-quality ones. This simple strategy enhances robustness against low-quality fingerprints while preserving precision for high-quality ones.

The GBFOE method is governed by five parameters in total, the first three controlling preprocessing steps, and the last two controlling orientation estimation:

- s_p, the contrast stretching clipping percentage,
- σ_p, the standard deviation of the Gaussian smooth filter,
- m_p, the size of the median filter,
- $\hat{\sigma}$, the standard deviation of the Gaussian filter used for the first orientation estimation,
- k_σ, the multiplicative factor to determine the standard deviation of the Gaussian filter used for the final orientation estimation.

The specific values used for the above parameters are documented in section V-B.

IV. SNFOE

SNFOE (Simple Network for Fingerprint Orientation Estimation) is a novel method based on a fully convolutional neural network. Drawing inspiration from the KISS principle, SNFOE is designed to achieve the following goals:

- *End-to-end orientation estimation* – the entire orientation estimation task is carried out within a single network, eliminating the need for separate preprocessing or postprocessing steps.
- *Purely convolutional architecture* – the network exclusively employs convolutional layers, allowing it to directly process fingerprints of any size[2] without requiring image resizing or mosaicking.
- *Dense pixel-wise estimation* – the output of the network is a dense orientation field, providing an estimate of the local orientation at each pixel within the fingerprint.
- *Compact network architecture* – the network is relatively compact, minimizing hardware requirements and computational cost.
- *Framework-agnostic implementation* – the network architecture employs standard layers and leverages a common optimization technique with a straightforward loss function. This facilitates easy implementation across various deep learning frameworks.

Fig. 4 illustrates the network architecture of SNFOE, meticulously chosen as the most promising design after evaluating several alternatives inspired by successful convolutional models like ResNet [66], U-Net [67], DeepLabv3+ [68], and ConvNext [69]. Appendix A details this selection process. The network comprises three key building blocks: the Encoder, the Decoder, and the Head block for final orientation estimation. It takes a two-channel image as input, containing the fingerprint $\mathbf{F}$ and its segmentation mask $\mathbf{S}$. Both the Encoder and Decoder are symmetrical, each with five levels. Each encoder level employs a 5×5 convolution with padding and ReLU activation, followed by batch normalization and 2×2 max pooling for downsampling. Conversely, each decoder level utilizes a 5×5 convolution with padding and ReLU activation, batch normalization, and a 2×2 spatial upsampling layer. Skip connections play a crucial role in information flow. They deliver feature maps obtained after the batch normalization layer of each encoder level to the corresponding decoder level, where they are concatenated with the upsampled features. An additional skip connection directly concatenates the input segmentation mask $\mathbf{S}$ to the features produced by the last decoder level. The Head block operates at the same resolution of the input fingerprint and applies 16 5×5 convolution filters with padding, ReLU activation, and batch normalization, generating 16 feature maps. A final 3×3 convolution layer with linear activation yields two feature maps, $\mathbf{X}$ and $\mathbf{Y}$, interpreted as containing the cosine and sine of the doubled orientation angle for each pixel. The DoubleAngleLayer, the

[2]There is a minor constraint on the image size: both the width and height of the image must be multiples of 32. If the dimensions do not already comply, the image needs to be padded to meet this requirement.

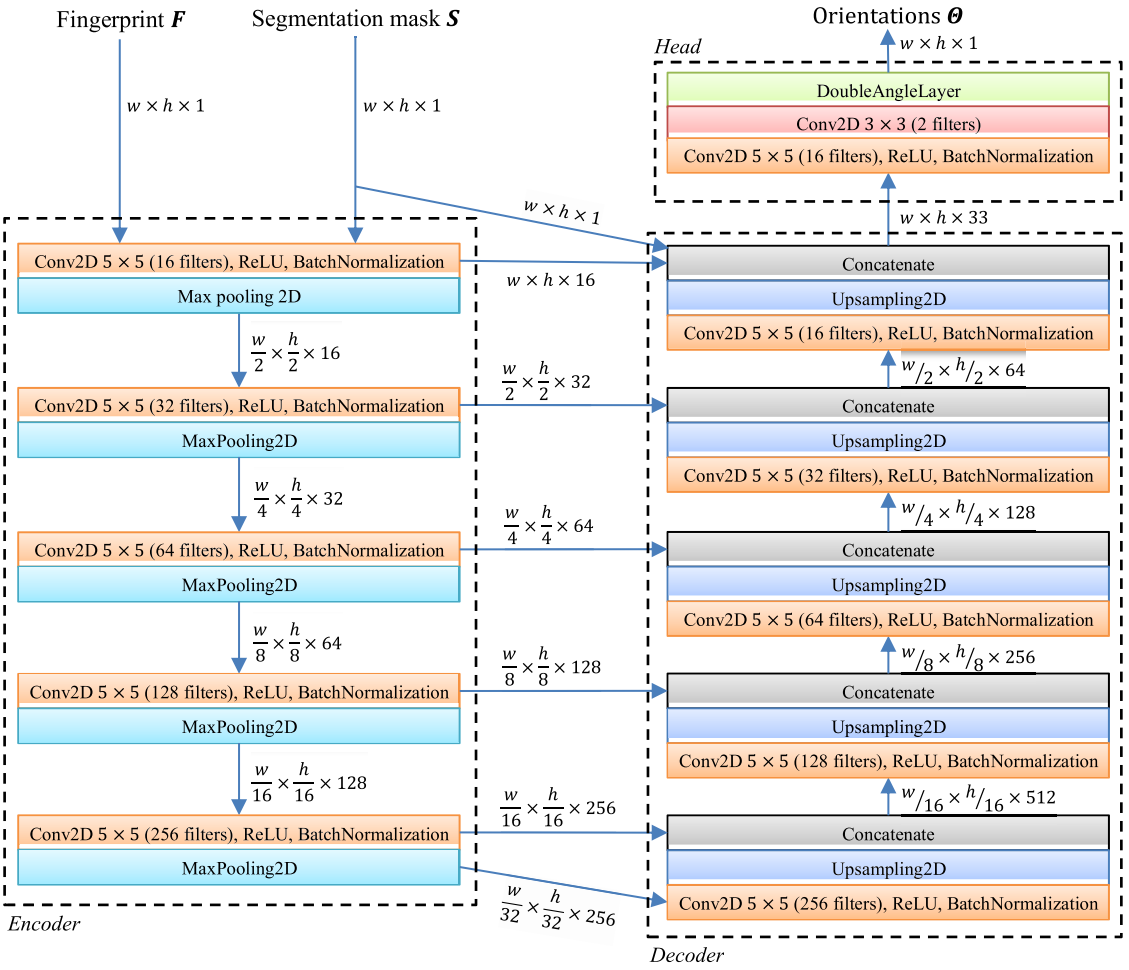

FIGURE 4. The network architecture of SNFOE.

network's final layer, calculates the orientation $\mathbf{\Theta}_{i,j}$ of each pixel using the following formula:

$$\mathbf{\Theta}_{i,j} = \frac{1}{2}\text{atan2}\left(\mathbf{Y}_{i,j}, \mathbf{X}_{i,j}\right) \tag{7}$$

This allows the network to learn the double-angle representation, offering advantages similar to those discussed earlier in section III.

The network is trained using the Huber loss [70] on the difference between the true orientations $\hat{\mathbf{\Theta}}$ and the predicted orientations $\mathbf{\Theta}$. For the precise formula of the loss function please refer to Appendix A.

Fig. 5 presents an illustrative example of orientation field estimation using SNFOE. It visualizes the network's processing pipeline, from the input (fingerprint $\mathbf{F}$ and segmentation mask $\mathbf{S}$) to the final predicted orientations $\mathbf{\Theta}$. The figure showcases the first 16 feature maps for each convolution layer within the Encoder, Decoder, and Head block. It is important to note that while the feature maps are resized to consistent dimensions for visualization purposes, their intrinsic resolutions vary from the original fingerprint size to $1/32^{th}$ of the original resolution. As we move through the encoding path, the feature maps undergo a transformation. Initially, they capture fine-grained details and localized patterns in the input. Gradually, they progress towards representing higher-level contextual information and global structures that increasingly hold relevance to the task of orientation estimation. Conversely, the decoding path progressively refines the feature maps, transforming them from abstract spatial representations back to precise pixel-level maps. It appears that each of these refined maps becomes specialized in representing specific ranges of orientations. Finally, the last convolutional layer generates the double-angle components, which are subsequently converted into the final orientation field by the last layer.

V. EXPERIMENTAL RESULTS

A. BENCHMARKS AND TRAINING DATASETS

Table 1 provides an overview of the fingerprint datasets employed in this study.

Four datasets, each containing 100 fingerprints, were sourced from FVC2002 [71], and FVC2004 [72] databases. These datasets were used for training the convolutional networks (please refer to Appendix A and Section V-C

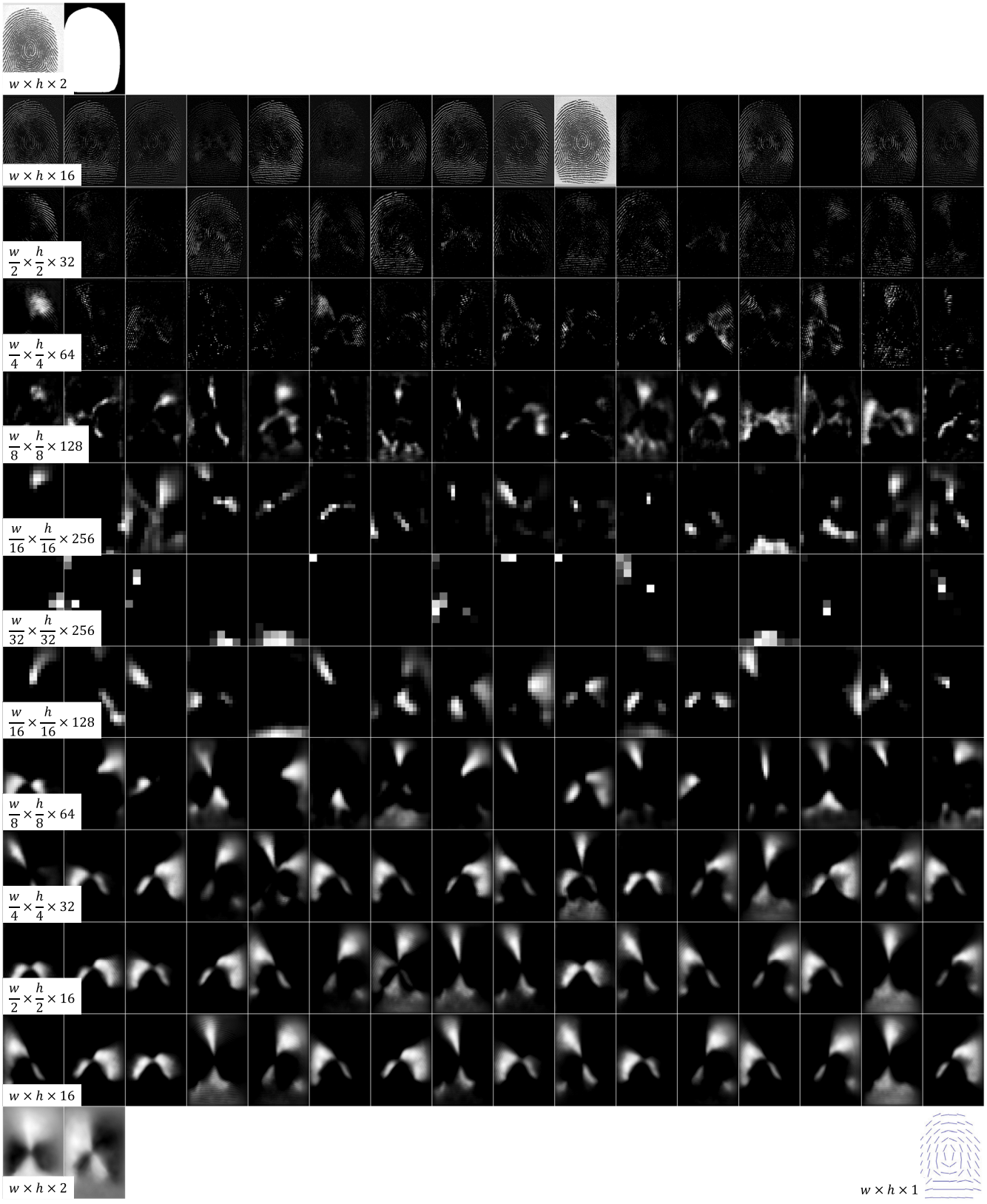

FIGURE 5. An example of input, feature maps, and output of the SNFOE network. The first 16 feature maps produced by each convolution layer are shown, all resized at the same dimensions for visualization purposes. The output orientation field Θ is depicted in the lower right corner, with line segments indicating orientations every 32 pixels.

TABLE 1. Fingerprint datasets used in this study.

Dataset	Fingerprints	Usage for GBFOE	Usage for SNFOE
Subset of FVC2002 DB2 A	100 (third impression of each finger)	-	Training
Subset of FVC2002 DB3 A	100 (third impression of each finger)	-	Training
Subset of FVC2004 DB1 A	100 (first impression of each finger)	-	Training
Subset of FVC2004 DB3 A	100 (third impression of each finger)	-	Training
FOE-TEST	60 (10 Good, 50 Bad)	Parameter tuning	Network model evaluation, training
FOE-STD-1.0	60 (10 Good, 50 Bad)	Evaluation (on FVC-onGoing)	Evaluation (on FVC-onGoing)
NIST SD27	258 (88 Good, 85 Bad, 85 Ugly)	Evaluation	Evaluation

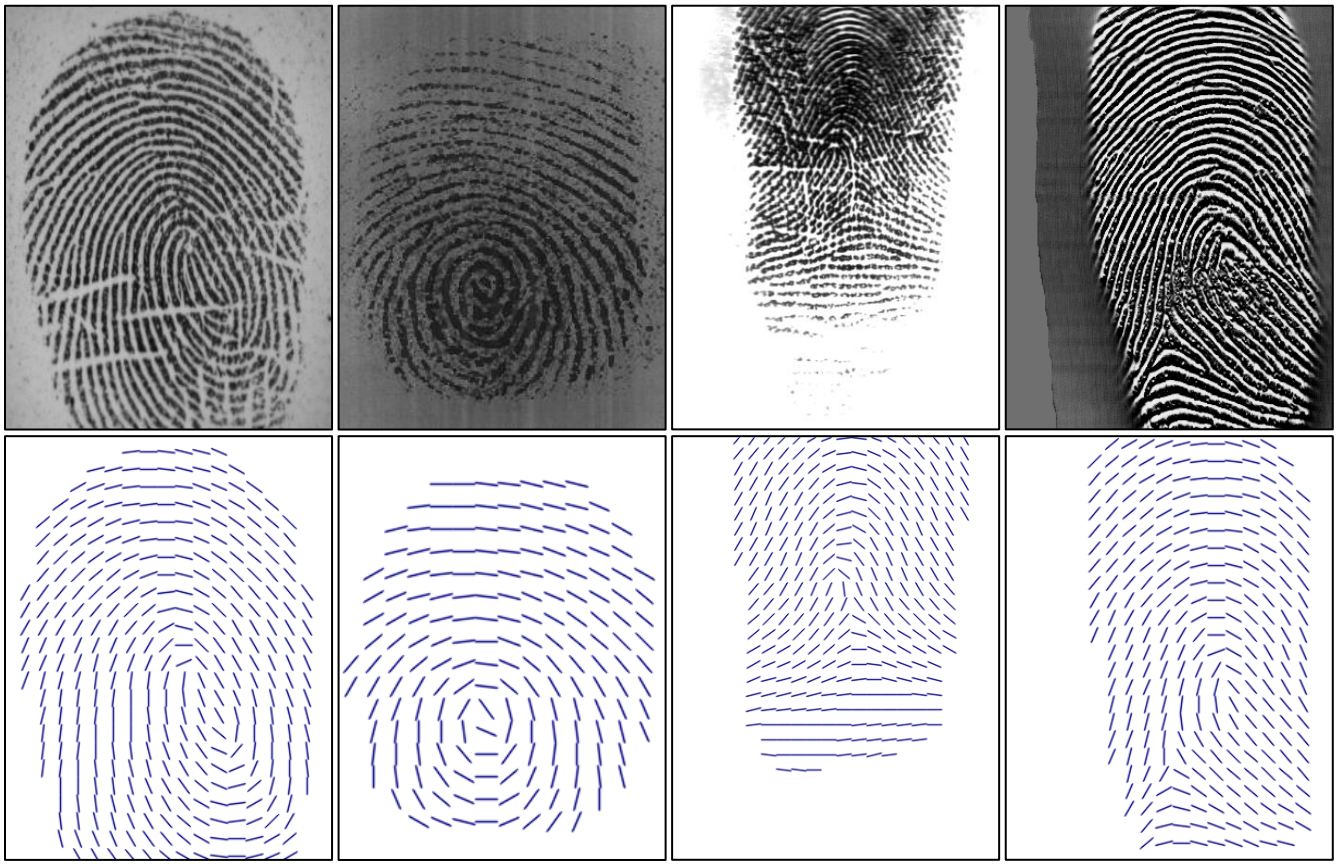

FIGURE 6. From left to right: an example fingerprint from FVC2002 DB2 (optical sensor), FVC2002 DB3 (capacitive sensor), FVC2004 DB1 (optical sensor), and FVC2004 DB3 (thermal sweeping sensor). The corresponding ground truth orientations are depicted with line segments indicating orientations every 16 pixels.

for additional details). These datasets encompass a diverse array of acquisition technologies, including two optical sensors, a capacitive sensor, and a thermal sweeping sensor. The ground truth orientation fields were meticulously annotated using a specifically designed software tool. Fig. 6 illustrates an example fingerprint from each of these datasets, with the corresponding ground truth orientations.

The remaining datasets listed in table 1 correspond to the primary benchmarks used for evaluating fingerprint orientation estimation methods: FVC-onGoing FOE and NIST SD27.

The FVC-onGoing automated evaluation system [73], [74] offers two fingerprint orientation field estimation benchmarks: FOE-TEST and FOE-STD-1.0. Both benchmarks comprise fingerprints captured with optical sensors and manually marked ground truth orientations. Each FOE benchmark is further divided into a "Good" dataset containing ten high-quality fingerprints, and a "Bad" dataset containing 50 low- and very-low-quality fingerprints. The key challenge lies in achieving accurate orientation estimation accuracy on the bad quality dataset while maintaining performance on the good quality dataset. FOE-TEST datasets are publicly available and serve three purposes in this work: i) tuning GBFOE parameters, ii) being the test set for selecting SNFOE network model, and iii) contributing to the training set of the final SNFOE model. Fig. 7 showcases examples of FOE-TEST fingerprints with

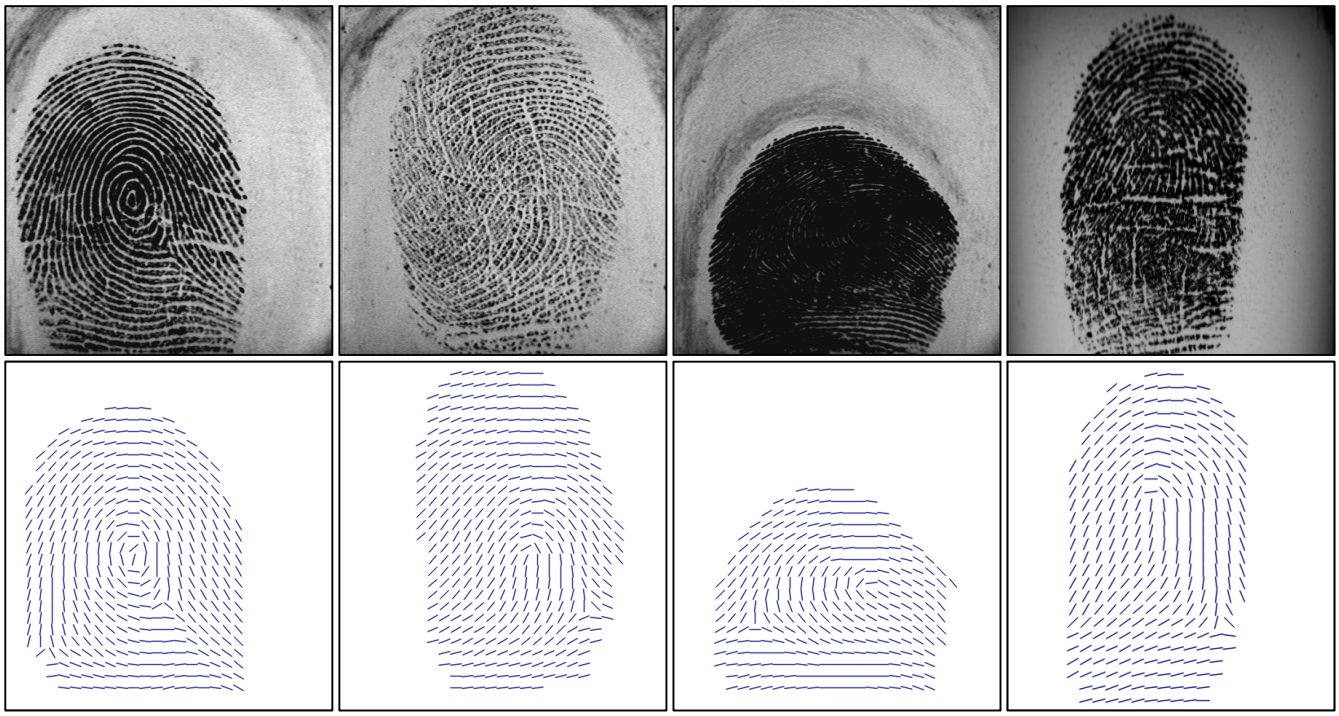

FIGURE 7. From left to right: an example fingerprint from the FOE-TEST "Good" dataset, and three from the "Bad" dataset. The corresponding ground truth orientations are depicted with line segments indicating orientations every 16 pixels.

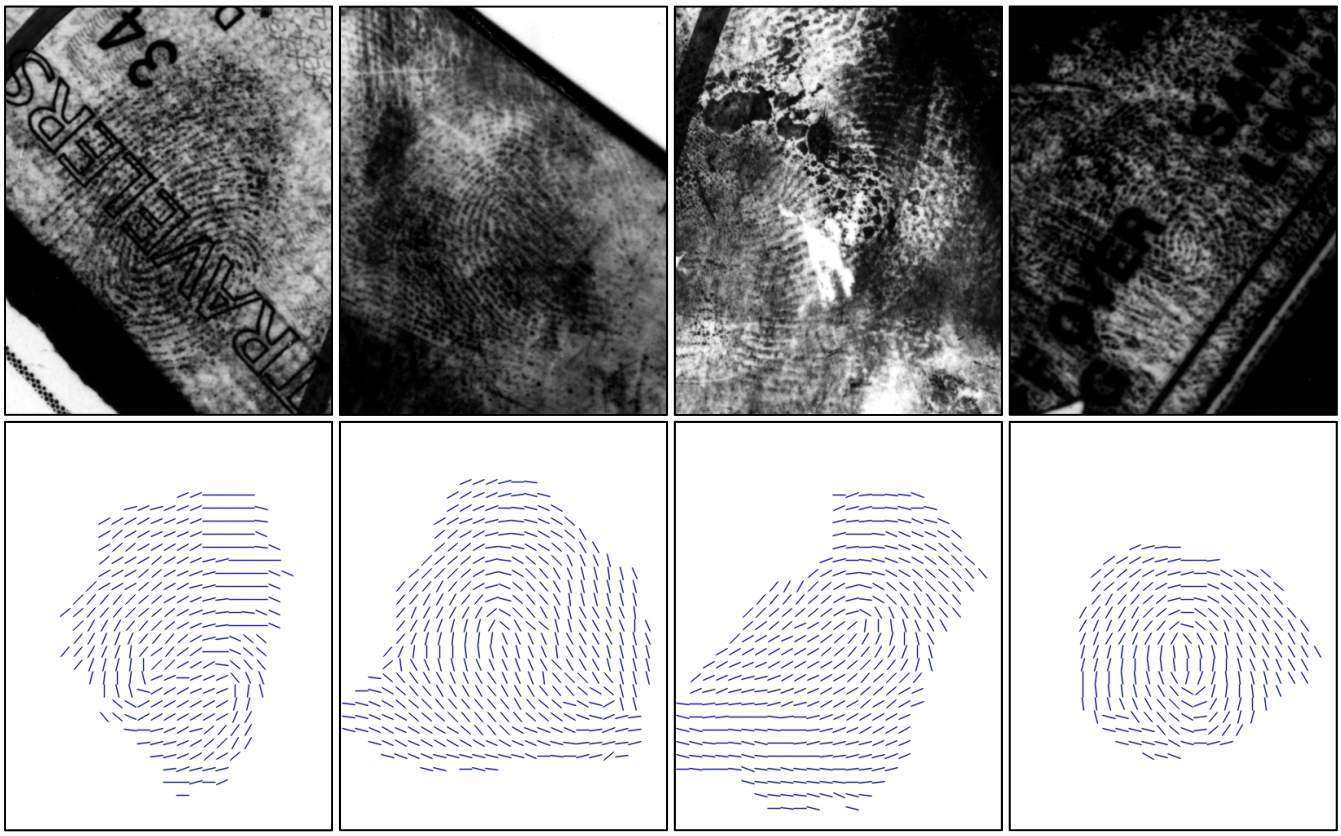

FIGURE 8. From left to right: an example latent fingerprint from the SD27 "Good" dataset, one from the "Bad" dataset, and two from the "Ugly" dataset. The corresponding ground truth orientations are depicted with line segments indicating orientations every 16 pixels.

the corresponding ground truth orientations. FOE-STD-1.0 fingerprints are sequestered, requiring algorithms to be submitted to FVC-onGoing for automated evaluation. Both GBFOE and SNFOE have been evaluated in this manner,

TABLE 2. Average RMSD on FOE-STD-1.0 (in degrees)*.

Category	Method	Name	Good	Bad
Local analysis	Baseline gradient estimation method	Gradient	5.86	21.83
Local analysis	Gradient estimation	AntheusOriEx	5.46	17.06
Local analysis	Bank of filters and iterative regularization	AnGaFRIS [21]	5.75	14.54
Local analysis	Gradient estimation with simple quality-based adjustment	GBFOE *(proposed)*	5.30	14.40
Global analysis	2D Fourier series expansion	FOMFE [37]	6.70	21.44
Global analysis	Parametric model and scar detection	MXR	5.59	11.36
Global analysis	Div-curl model	ROF [47]	5.24	11.20
Learning based	Adaptive learning-based technique	Adaptive-3 v0.1 [44]	5.76	13.45
Learning based	Adaptive learning-based technique (improved version)	Adaptive-3 v0.2	5.93	13.27
Learning based	Localized orientation patch dictionaries	LocalDict [55]	6.08	9.66
Learning based	CNN with orientation regression	ConvNetOF [59]	5.80	8.53
Learning based	Deep learning (no other detail is provided)	ORI-NET	6.94	8.44
Learning based	CNN with orientation deep expectation	DEX-OF v1.0 [60]	4.89	7.52
Learning based	CNN with orientation deep expectation (improved version)	DEX-OF v1.1	4.84	6.73
Learning based	Simple but effective end-to-end CNN	SNFOE *(proposed)*	**4.30**	**6.37**

*Best values are in bold

FIGURE 9. Examples of the data augmentation techniques applied to the training fingerprints. Multiple augmentation techniques are randomly applied to each fingerprint.

to facilitate comparison with other methods on the same benchmark.

NIST SD27 [75] contains 258 latent fingerprints categorized into three datasets of decreasing quality: ''Good'' dataset (88 fingerprints), ''Bad'' dataset (85 fingerprints), ''Ugly'' dataset (85 fingerprints). Fig. 8 presents examples of NIST SD27 fingerprints alongside their corresponding ground truth orientations provided by Feng et al. in [51]. Although GBFOE and SNFOE are not specifically designed to deal with latent prints, they were also evaluated on this benchmark to allow comparison with recent orientation field estimation methods that were tested on NIST SD27.

Both FOE and SD27 benchmarks employ the Root Mean Square Deviation (RMSD) to assess the accuracy of orientation field estimation.

$$RMSD\left(\boldsymbol{\Theta},\hat{\boldsymbol{\Theta}},\mathbf{S}\right)=\sqrt{\frac{\sum_{i,j}\left(\mathbf{S}_{i,j}\cdot d\phi\left(\boldsymbol{\Theta}_{i,j},\hat{\boldsymbol{\Theta}}_{i,j}\right)\right)^2}{\sum_{i,j}\mathbf{S}_{i,j}}} \tag{8}$$

where $\boldsymbol{\Theta}$ are the estimated orientations, $\hat{\boldsymbol{\Theta}}$ the ground truth orientations, $\mathbf{S}$ the ground truth segmentation mask, and $d\phi\left(\theta,\hat{\theta}\right)$ is the difference between two orientations θ and $\hat{\theta}$:

$$d\phi\left(\theta,\hat{\theta}\right)=\left(\theta-\hat{\theta}+\frac{\pi}{2}\right)\bmod\pi-\frac{\pi}{2} \tag{9}$$

The accuracy on a whole dataset is measured as the average RMSD and is expressed in degrees.

B. GBFOE PARAMETER SELECTION

GBFOE is controlled by five parameters (see section III). In this experimentation, the following values were used: $s_p = 19$, $\sigma_p = \frac{5}{4}$, $m_p = 5$, $\hat{\sigma} = 27$, and $k_\sigma = 43$. These values were chosen on FOE-TEST as the ones that minimize the average RMSD on the ''Bad dataset'' among a set of reasonable combinations of values. The same values were used to evaluate GBFOE on both FOE-STD-1.0 and NIST SD27 benchmarks.

C. SNFOE TRAINING

The selection of SNFOE architecture, loss function, and hyperparameters was performed using the 400 FVC fingerprints as the training set, and the 60 FOE-TEST fingerprints as the test set (see Appendix A for the details). When training the final model, FOE-TEST fingerprints were also included in the training set, resulting in a total of 460 fingerprints. The entire training process took about 15 minutes on a PC with an NVIDIA GeForce RTX™ 3080 Ti GPU. The training

TABLE 3. Average RMSD on NIST SD27 (in degrees)*.

Category	Method	Name	All	Good	Bad	Ugly
Local analysis	Local Fourier analysis	STFT [23]	32.51	27.27	34.10	36.36
Local analysis	Gradient estimation with simple quality-based adjustment	GBFOE *(proposed)*	28.44	22.49	30.02	33.01
Global analysis	2D Fourier series expansion	FOMFE [37]	28.12	22.83	29.09	32.63
Learning based	Orientation patch dictionary	GlobalDict [51]	18.44	14.40	19.18	21.88
Learning based	Ridge patch dictionary	RidgeDict [54]	19.53	15.34	20.70	22.68
Learning based	Localized orientation patch dictionaries	LocalDict [55]	14.35	11.15	15.15	16.85
Learning based	Localized orientation patch dictionaries (manual fingerprint pose)	LocalDict-M [55]	13.76	10.87	14.12	16.40
Learning based	Patch-based CNN orientation estimation	ConvNet [57]	13.51	10.76	13.94	16.00
Learning based	Multiscale orientation patch dictionaries	MultiScaleDict [52]	16.09	12.35	16.94	19.11
Learning based	CNN with Dilated Spatial Pyramid Pooling (DSPP)	FingerNet [58]	17.82	13.67	18.42	21.50
Learning based	Dictionaries combined with sparse coding	SparseCoding [53]	16.38	12.57	16.88	20.22
Learning based	Exhaustive search on a large database (boosting version)	ExSearch-B [56]	13.54	11.21	14.20	14.95
Learning based	Exhaustive search on a large database	ExSearch [56]	13.01	10.85	13.99	14.27
Learning based	CNN for residual learning on predefined patterns	PriorK [63]	12.16	9.87	12.83	13.85
Learning based	CNN for residual learning on predefined patterns (manual f. pose)	PriorK-M [63]	12.10	9.60	12.71	14.07
Learning based	Simple but effective end-to-end CNN	SNFOE *(proposed)*	**11.73**	**9.38**	**12.43**	**13.47**

*Best values are in bold

procedure closely followed the approach used during the network architecture search, as described in Appendix A. Here is a summary of the key elements:

- Input image size – all training fingerprints were padded to a uniform size of 512 × 512 pixels.
- Data augmentation – various augmentation techniques were applied, including random translation, rotation, scale, horizontal flip, gamma correction, contrast reduction, morphology operations, simulation of scratches and abrasions (see Fig. 9).
- Optimization algorithm – the Adam optimizer [76] with Nesterov momentum [77] was employed. The learning rate was dynamically adjusted between 10^{-5} and 0.025 using a cosine decay strategy with warmup. The momentum parameters were set to $\beta_1 = 0.2$ and $\beta_2 = 0.5$.
- Training epochs – The model underwent a fixed number of 25 training epochs, with each epoch consisting of 120 batches, each containing 16 fingerprints.

D. RESULTS

GBFOE is implemented in Python using the OpenCV library [78]. On a PC with an Intel® Xeon® Silver 4112 CPU at 2.60GHz, the average orientation field estimation time for a single fingerprint is approximately 60ms.

SNFOE is implemented in Python with the Keras library [79]. On a PC equipped with an NVIDIA GeForce RTX™ 3080 Ti GPU, the average orientation field estimation time for a single fingerprint is approximately 44ms.

Table 2 compares GBFOE and SNFOE to the 13 approaches whose results on FOE-STD-1.0 are published on FVC-onGoing [74]. These methods can be categorized into three groups:

- Local analysis – Gradient, AntheusOriEx, and AnGaFRIS [21],
- Global analysis – FOMFE [37], MXR, and ROF [47],
- Learning-based – Adaptive-3 [44] (and its improved version), LocalDict [55], ConvNetOF [59], ORI-NET, DEX-OF [60], and its improved version.

GBFOE achieves the best performance among local methods on both the "Good" (5.30° RMSD) and "Bad" (14.40° RMSD) datasets. It even outperforms the global method FOMFE and comes close to the performance of the learning-based method Adaptive-3. SNFOE surpasses all other methods on this benchmark, achieving a remarkable result of 4.30° RMSD on the "Good" dataset and 6.37° RMSD on the "Bad" dataset.

Table 3 compares GBFOE and SNFOE to 14 existing methods on NIST SD27:

- Local analysis – STFT [23],
- Global analysis – FOMFE [37],
- Learning-based – GlobalDict [51], RidgeDict [54], LocalDict [55] (with and without manual fingerprint pose marking), ConvNet [57], MultiScaleDict [52], FingerNet [58], SparseCoding [53], ExSearch [56] (and its faster, slightly less accurate "boosting" variant), PriorK [63] (with and without manual fingerprint pose marking).

Following previous works, the table reports average RMSD on individual datasets ("Good", "Bad", and "Ugly") as well as the average RMSD over all fingerprints ("All").

Based on local analysis, GBFOE is obviously not well-suited for latent fingerprint processing. However, it outperforms the other local method (STFT) and exhibits performance close to the global method (FOMFE).

SNFOE, once again, demonstrates exceptional performance by achieving the lowest average RMSD on every

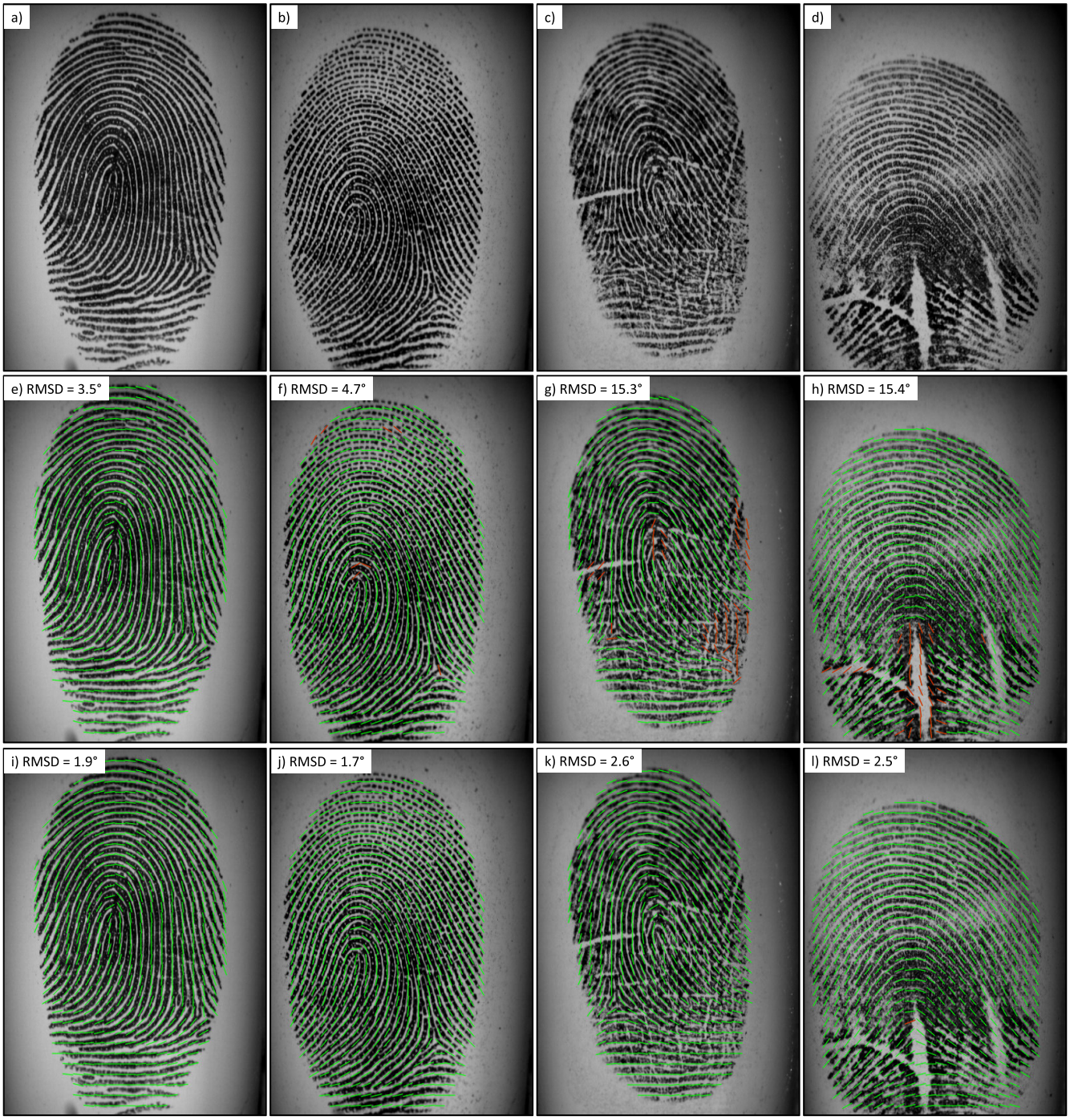

FIGURE 10. Examples of orientation field estimation on four fingerprints from FVC2006 DB2 [80]. Each column shows a fingerprint (a, b, c, d), and the corresponding results using GBFOE (e, f, g, h) and SNFOE (i, j, k, l). The estimated orientations are visualized by line segments indicating the orientation at every 16 pixels. Red segments highlight errors exceeding 15°. The RMSD is reported for each estimated orientation field.

dataset. It even achieves slightly better results than PriorK-M [63], which relies on human intervention for manual alignment of latent fingerprints. SNFOE's results on this benchmark are truly astonishing, especially considering that, unlike other methods in the table, it was not specifically trained to handle latent fingerprints. Instead, it was simply trained on a small set of plain fingerprints acquired using online sensors.

E. ILLUSTRATIVE CASES

Fig. 10 shows examples of orientation field estimation from fingerprints captured with an optical sensor. Notably, neither fingerprints from FOE-STD-1.0 nor those from FOE-TEST could serve this purpose. The former dataset is sequestered, and the latter was used for training SNFOE. Consequently, four fingerprints were chosen from a distinct dataset (FVC2006 DB2 [80]). Fig. 10.a shows a high-quality

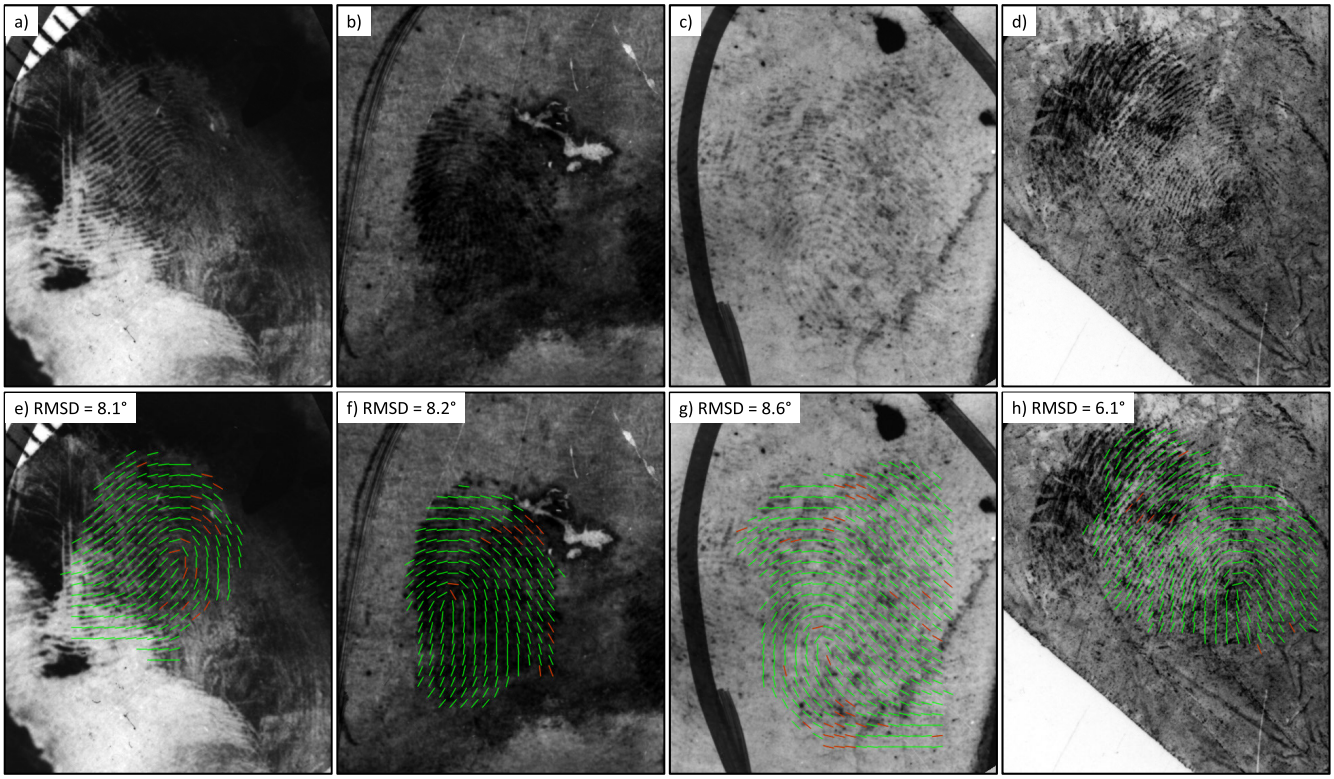

FIGURE 11. Successful SNFOE orientation estimation examples on latent fingerprints. From left to right: a fingerprint from the SD27 "Good" dataset (a, e), one from the "Bad" dataset (b, f), and two from the "Ugly" dataset (c, g; d, h). The estimated orientations are visualized by line segments indicating the orientation at every 16 pixels. Red segments highlight errors exceeding 15° compared to the ground truth. The RMSD is reported for each estimated orientation field.

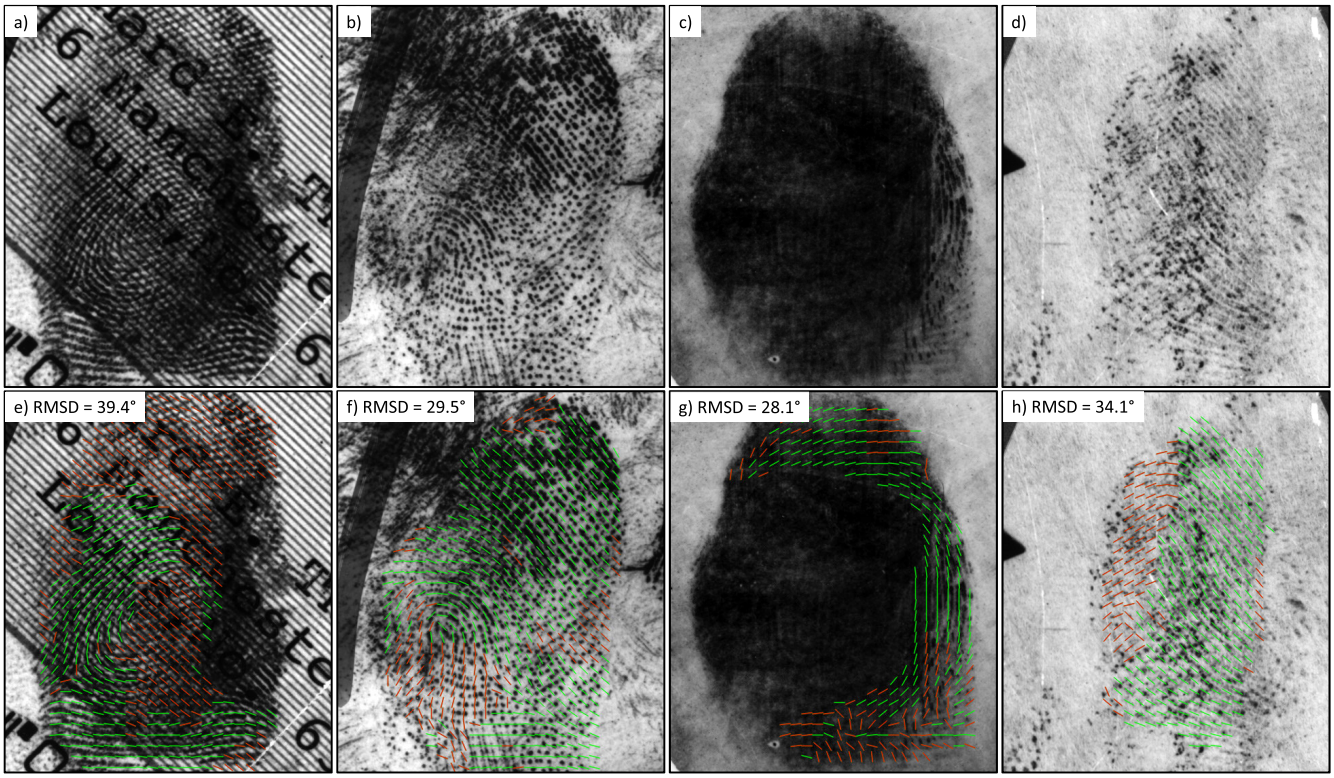

FIGURE 12. Examples of SNFOE's failure cases on latent fingerprints. From left to right: a fingerprint from the SD27 "Good" dataset (a, e), one from the "Bad" dataset (b, f), and two from the "Ugly" dataset (c, g; d, h). The estimated orientations are visualized by line segments indicating the orientation at every 16 pixels. Red segments highlight errors exceeding 15°. The RMSD is reported for each estimated orientation field.

TABLE 4. Comparison between the key features of SNFOE and PriorK.

Element	SNFOE (proposed)	PriorK [63]
Network architecture	Symmetrical encoder and decoder blocks with five convolutional and spatial downsampling layers, skip connections, and a final head block with two convolutional layers and a double-angle orientation representation layer.	Feature extraction module with three convolutional and spatial downsampling blocks, followed by several residual blocks. Dilated Spatial Pyramid Pooling (DSPP) module that contains convolutional layers with various dilation rates to capture multiscale information.
Loss function	Huber loss on the difference between ground truth and predicted orientations (one hyperparameter).	Weighted sum of three loss functions (five hyperparameters in total): 1) Orientation loss – based on the focal loss. 2) Smoothness loss – a regularization loss to constrain the smoothness of estimated residual orientations. 3) Reliability loss – based on focal loss, to predict reliability score.
Training set	460 plain fingerprints (from FOE-TEST and some FVC databases) with manually annotated ground truth orientations.	Simulated latent fingerprints created using: i) the first 14,000 fingerprints in NIST SD14 [81], ii) a database of images containing text (MSRA-TD500 [82]), and iii) various plain fingerprints from FVC2004 [72]. Ground truth orientations generated using FingerNet [58].
Training procedure	Adam optimizer with Nesterov momentum, data augmentation.	1) Generation of the four statistical orientation patterns (arch, left loop, right loop, and whorl): all the orientation fields of training fingerprints are aligned to the same coordinates, based on their pose, using the method proposed in [83]. After alignment, a clustering method is applied to classify the orientation fields into the four patterns and generate the center of each clustering, i.e. the average orientation field. 2) Training of the CNN: Adam optimizer, data augmentation.
Orientation estimation	Network input: fingerprint and segmentation mask. Network output: orientation field.	1) Estimate fingerprint coordinate system using the pose estimation network proposed in [83]. 2) Network inference is repeated four times: one for each reference orientation pattern. Network input: fingerprint and the reference orientation pattern transformed according to the coordinates estimated at the previous step. Network output: residual orientation field and reliability score. 3) Fuse the four orientation estimations (reference + residual), weighted by reliability scores, into a single orientation field. 4) Apply a final smoothing by replacing each orientation element with the average of its 3×3 neighborhood.

fingerprint. Both GBFOE and SNFOE provide accurate estimations (Fig. 10.e and 10.i), with RMSD of 3.5° and 1.9°, respectively. The fingerprint in Fig. 10.b is still of good quality, but it has numerous ridge discontinuities in the top region, likely due to dry skin or low pressure on the sensor. GBFOE, during the initial estimation (see section III), detects lower orientation coherence and, consequently, chooses a larger Gaussian filter to average the squared gradients. While this reduces sensitivity to noise, it compromises accuracy in the core region, which has high curvature, leading to an RMSD of 4.7° (Fig. 10.f). Conversely, SNFOE handles this case gracefully, achieving an RMSD of 1.7° (Fig. 10.j). The fingerprint in Fig. 10.c is of lower quality, featuring several scratches, especially in the bottom right region. GBFOE employs a large Gaussian filter, aiding in scratch recovery, but sacrificing accuracy in the core region. Despite this, it fails to estimate the correct orientations in the bottom right (Fig. 10.g). In contrast, SNFOE successfully estimates the orientation field (Fig. 10.k). Finally, the fingerprint in Fig. 10.d presents a wide vertical scratch, which poses a significant challenge for GBFOE, resulting in inaccurate estimations (Fig. 10.h). SNFOE, however, handles this case effectively as well (Fig. 10.l).

Fig. 11 presents four examples demonstrating SNFOE's accurate estimation of the orientation field for latent fingerprints from NIST SD27. Despite the fingerprints' very low quality, the proposed method achieves impressive results. Most of the estimated orientations are correct, and the RMSD remains remarkably low.

However, within this benchmark there still exist latent fingerprints where SNFOE's performance falls short. Fig. 12 illustrates four such cases. In these instances, overlapped patterns or other noise elements confound the proposed method. Consequently, in many regions, SNFOE struggles to estimate the correct orientations, leading to significantly high RMSD values.

F. ABOUT COMPLEXITY

While evaluating the accuracy of orientation estimation is crucial, we must also consider the overall complexity of the

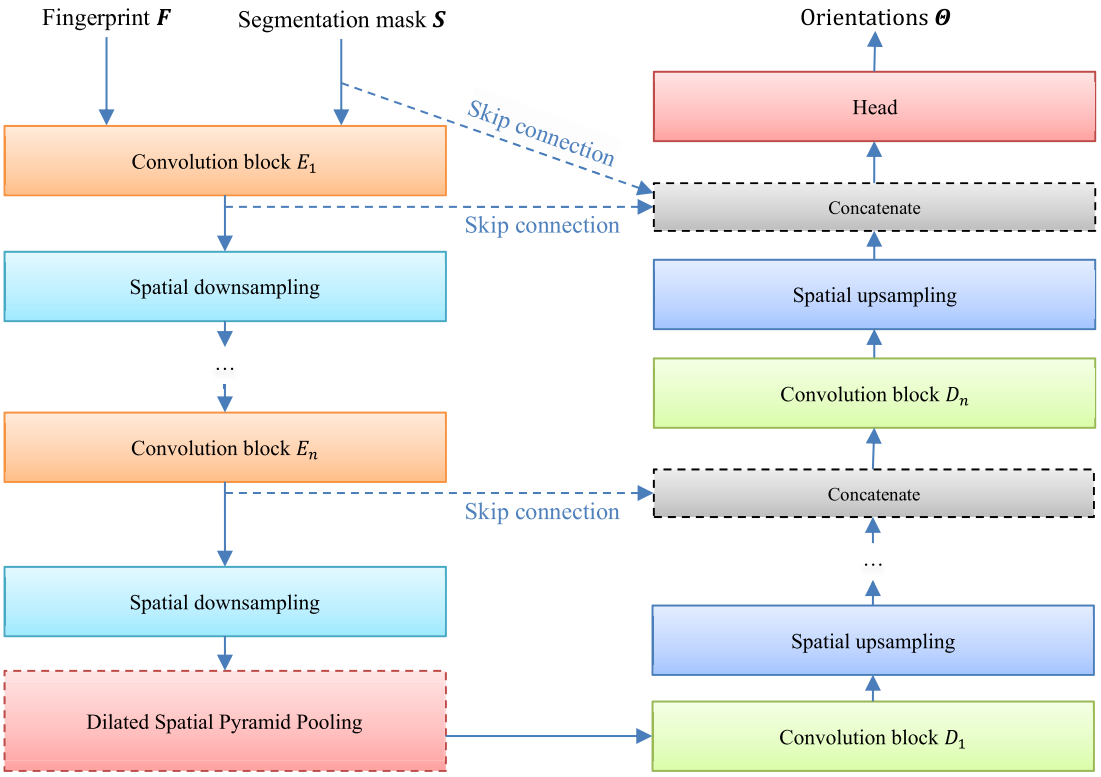

FIGURE 13. The generic architecture used during the search of the best network architecture for SNFOE. Dashed blocks and connections denote optional elements.

methods to fully assess their value. From this perspective, GBFOE becomes even more attractive compared to the other local and global methods it competes with in terms of performance. In fact, GBFOE only requires simple basic image processing operations and repeating the orientation estimation twice based on gradient components: it can be implemented in less than 50 lines of Python code. Similarly, SNFOE, although undeniably more complex than GBFOE, emerges as incredibly simple when compared to the most recent learning-based methods. Table 4 compares the key features of SNFOE and the second most accurate method in Table 3. The significant difference in complexity is evident.

VI. CONCLUSION

This paper introduces two novel methods for estimating the fingerprint orientation field: GBFOE and SNFOE. Both approaches adhere to the KISS principle, emphasizing simplicity in design and computational requirements.

The evaluation conducted on publicly available benchmarks demonstrates the effectiveness of the proposed methods. Despite its simplicity, GBFOE outperforms all local techniques and even approaches the performance of some significantly more complex global and learning-based methods on the FOE benchmark. SNFOE, on the other hand, stands out as a truly remarkable development. Trained exclusively on a small set of plain fingerprints, it surpasses all existing methods on both the FOE and NIST SD27 benchmarks, achieving exceptional accuracy even for challenging latent fingerprints. This achievement underscores the potential of simple, learning-based approaches in achieving superior performance in the field.

These findings not only significantly contribute to the advancement of orientation field estimation techniques but also advocate for the value of simplicity in scientific research. By demonstrating that effective solutions can be achieved through straightforward methodologies, this work paves the way for further exploration of efficient and practical approaches in fingerprint recognition.

While SNFOE demonstrates exceptional performance, additional studies are warranted to address its limitations in handling certain latent fingerprint cases. The open-source implementation[3] of both GBFOE and SNFOE facilitates further research and development, encouraging future exploration of these promising methods.

APPENDIX A
THE QUEST FOR THE OPTIMAL NETWORK MODEL

This appendix details the choices and experiments that led to defining the network model used in SNFOE.

The desired convolutional neural network has to determine the orientation of the fingerprint ridges at each pixel location, resulting in a matrix matching the input image dimensions.

[3]https://github.com/raffaele-cappelli/pyfing

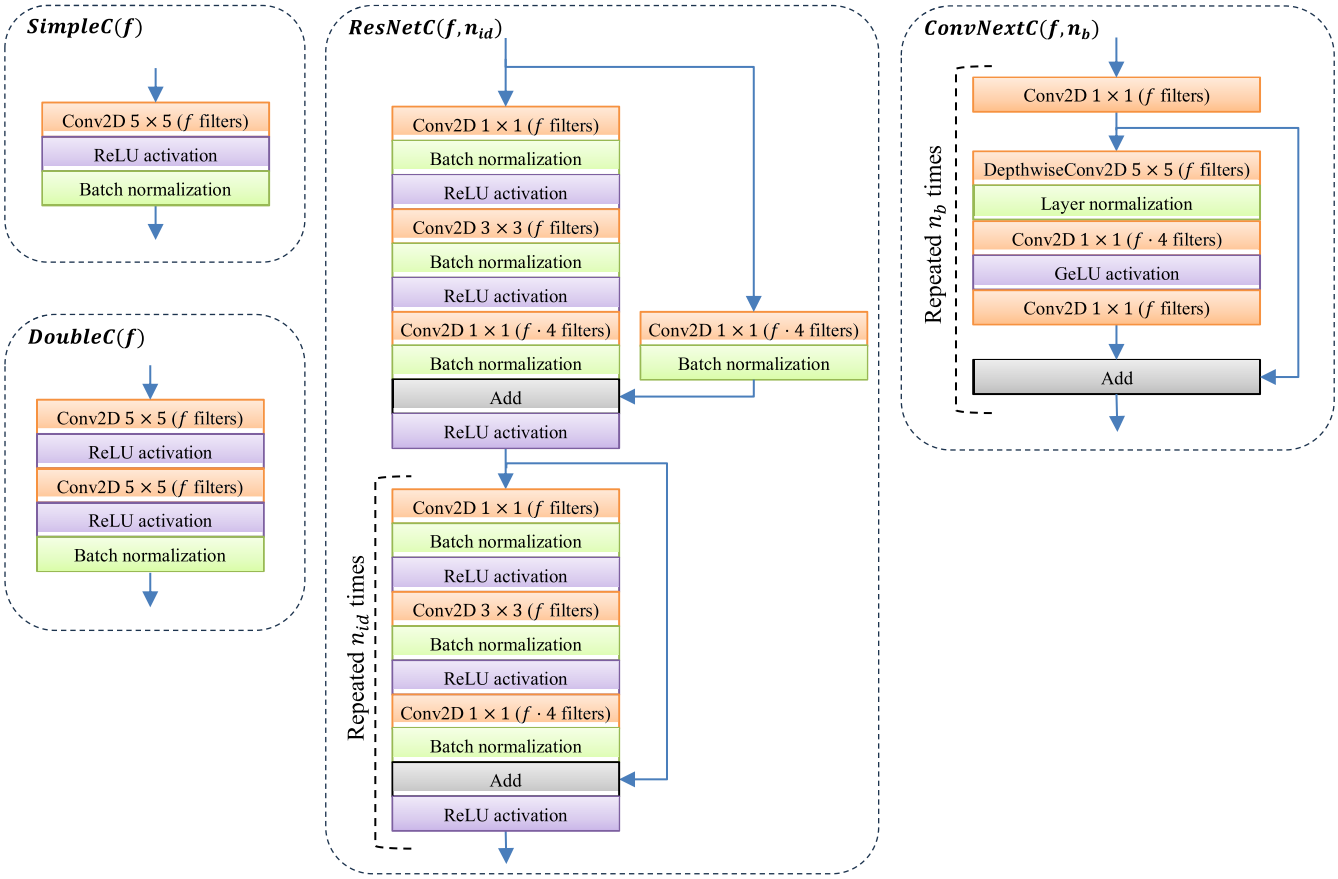

FIGURE 14. The four parametric convolution blocks used for building the various network architectures.

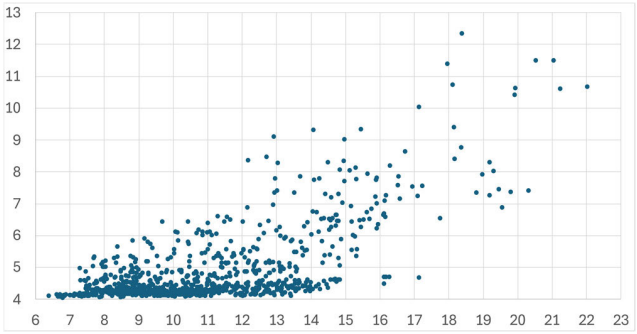

FIGURE 15. Results of the 960 models evaluated on FOE-TEST. The x and y coordinates of each point are the corresponding RMSD on the "Bad" and "Good" dataset, respectively.

Therefore, it resembles a semantic segmentation network more than a classification network. Consequently, successful convolutional architectures in segmentation, like U-net [67] and DeepLabv3+ [68], were analyzed.

A. INITIAL EXPERIMENTS

A series of exploratory experiments were conducted with early architecture prototypes to identify the most promising training procedure and data augmentation strategy. Based on these initial trials, the following procedures and hyperparameters were chosen:

- *Input image size* – all training fingerprints were padded to a uniform size of 512 × 512 pixels.
- *Data augmentation* – fingerprint-specific augmentation techniques were implemented, including random translation, rotation, scale, horizontal flip, gamma correction, contrast reduction, morphological operations, simulated scratches and abrasions (Fig. 9).
- *Optimization algorithm* – the Adam optimizer [76] with Nesterov momentum [77] was employed. The learning rate was dynamically adjusted between 10^{-5} and 0.025 using a cosine decay strategy with warmup. The momentum parameters were set to $\beta_1 = 0.2$ and $\beta_2 = 0.5$.
- *Training epochs* – Due to the limited availability of fingerprints with orientation ground truth, a validation set was not used to determine training termination. Instead, a fixed number of 25 training epochs was established, with each epoch consisting of 120 batches of 16 or 8 fingerprints, depending on the architecture's memory requirements.

B. THE GENERIC NETWORK ARCHITECTURE

To select the most promising network model, a generic architecture was established, as illustrated in Fig. 13. This architecture comprises the following components:

TABLE 5. The 12 base network architectures.

Name	n	Encoder convolution blocks	Decoder convolution blocks	Parameters[4]	Layers
U01	4	$E_1 = DoubleC(12)$ $E_2 = DoubleC(24)$ $E_3 = DoubleC(48)$ $E_4 = DoubleC(96)$	$D_1 = DoubleC(96)$ $D_2 = DoubleC(48)$ $D_3 = DoubleC(24)$ $D_4 = DoubleC(12)$	1 M	43
U05	5	$E_1 = DoubleC(12)$ $E_2 = DoubleC(24)$ $E_3 = DoubleC(48)$ $E_4 = DoubleC(96)$ $E_5 = DoubleC(192)$	$D_1 = DoubleC(192)$ $D_2 = DoubleC(96)$ $D_3 = DoubleC(48)$ $D_4 = DoubleC(24)$ $D_5 = DoubleC(12)$	5 M	52
U20	6	$E_1 = DoubleC(12)$ $E_2 = DoubleC(24)$ $E_3 = DoubleC(48)$ $E_4 = DoubleC(96)$ $E_5 = DoubleC(192)$ $E_6 = DoubleC(372)$	$D_1 = DoubleC(372)$ $D_2 = DoubleC(192)$ $D_3 = DoubleC(96)$ $D_4 = DoubleC(48)$ $D_5 = DoubleC(24)$ $D_6 = DoubleC(12)$	20 M	61
S01	4	$E_1 = SimpleC(16)$ $E_2 = SimpleC(32)$ $E_3 = SimpleC(64)$ $E_4 = SimpleC(128)$	$D_1 = SimpleC(128)$ $D_2 = SimpleC(64)$ $D_3 = SimpleC(32)$ $D_4 = SimpleC(16)$	1 M	35
S05	5	$E_1 = SimpleC(16)$ $E_2 = SimpleC(32)$ $E_3 = SimpleC(64)$ $E_4 = SimpleC(128)$ $E_5 = SimpleC(256)$	$D_1 = SimpleC(256)$ $D_2 = SimpleC(128)$ $D_3 = SimpleC(64)$ $D_4 = SimpleC(32)$ $D_5 = SimpleC(16)$	5 M	42
S20	6	$E_1 = SimpleC(16)$ $E_2 = SimpleC(32)$ $E_3 = SimpleC(64)$ $E_4 = SimpleC(128)$ $E_5 = SimpleC(256)$ $E_6 = SimpleC(512)$	$D_1 = SimpleC(512)$ $D_2 = SimpleC(256)$ $D_3 = SimpleC(128)$ $D_4 = SimpleC(64)$ $D_5 = SimpleC(32)$ $D_6 = SimpleC(16)$	20 M	49
R01	4	$E_1 = SimpleC(12)$ $E_2 = SimpleC(24)$ $E_3 = ResNetC(48,2)$ $E_4 = ResNetC(96,2)$	$D_1 = SimpleC(96)$ $D_2 = SimpleC(48)$ $D_3 = SimpleC(24)$ $D_4 = SimpleC(12)$	1 M	99
R05	5	$E_1 = SimpleC(12)$ $E_2 = SimpleC(24)$ $E_3 = ResNetC(48,2)$ $E_4 = ResNetC(96,3)$ $E_5 = ResNetC(192,2)$	$D_1 = SimpleC(192)$ $D_2 = SimpleC(96)$ $D_3 = SimpleC(48)$ $D_4 = SimpleC(24)$ $D_5 = SimpleC(12)$	5 M	149
R20	6	$E_1 = SimpleC(12)$ $E_2 = SimpleC(24)$ $E_3 = ResNetC(48,2)$ $E_4 = ResNetC(96,2)$ $E_5 = ResNetC(192,3)$ $E_6 = ResNetC(384,2)$	$D_1 = SimpleC(384)$ $D_2 = SimpleC(192)$ $D_3 = SimpleC(96)$ $D_4 = SimpleC(48)$ $D_5 = SimpleC(24)$ $D_6 = SimpleC(12)$	20 M	188
N01	4	$E_1 = SimpleC(16)$ $E_2 = SimpleC(32)$ $E_3 = ConvNextC(64,3)$ $E_4 = ConvNextC(128,3)$	$D_1 = SimpleC(128)$ $D_2 = SimpleC(64)$ $D_3 = SimpleC(32)$ $D_4 = SimpleC(16)$	1 M	85
N05	5	$E_1 = SimpleC(14)$ $E_2 = SimpleC(28)$ $E_3 = ConvNextC(56,3)$ $E_4 = ConvNextC(112,5)$ $E_5 = ConvNextC(224,3)$	$D_1 = SimpleC(224)$ $D_2 = SimpleC(112)$ $D_3 = SimpleC(56)$ $D_4 = SimpleC(28)$ $D_5 = SimpleC(14)$	5 M	135
N20	6	$E_1 = SimpleC(14)$ $E_2 = SimpleC(28)$ $E_3 = ConvNextC(56,2)$ $E_4 = ConvNextC(112,4)$ $E_5 = ConvNextC(224,6)$ $E_6 = ConvNextC(448,3)$	$D_1 = SimpleC(448)$ $D_2 = SimpleC(224)$ $D_3 = SimpleC(112)$ $D_4 = SimpleC(56)$ $D_5 = SimpleC(28)$ $D_6 = SimpleC(14)$	20 M	176

[4] The reported number of parameters excludes any additional parameters required for downsampling/upsampling layers, or for the DSPP module.

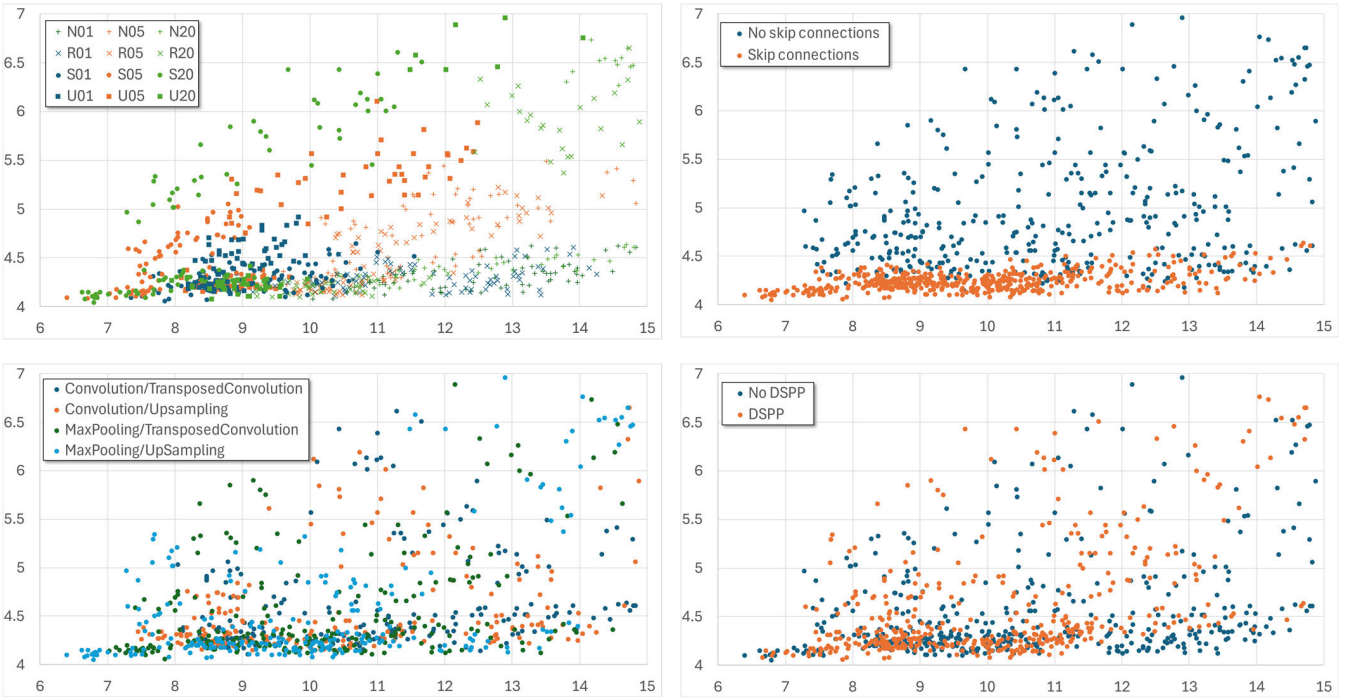

FIGURE 16. Results of the 960 models on the FOE-TEST benchmark. All graphs plot the accuracy on the "Bad" and "Good" datasets on the x and y axis, respectively. A different variable is used for labeling the points in each graph. From top to bottom and from left to right: base architecture, presence of skip connections, downsampling and upsampling techniques, and presence of DSPP.

- *Encoding path* – a series of n levels that extract features while progressively reducing spatial resolution.
- Optional *Dilated Spatial Pyramid Pooling (DSPP)* – inspired by DeepLabv3+ [68].
- *Decoding path* – similar to the encoding path, it has the same number of n levels. It extracts features while gradually increasing their resolution and optionally concatenates features from *skip connections*.
- *Head* – the final block responsible for estimating the orientations.

This generic architecture served as the foundation for a series of specific architectures, as detailed below.

The head block implementation consists of a simple convolutional layer (16 5×5 filters with padding) followed by ReLU activation, batch normalization, another convolutional layer (two 3×3 filters with padding), and finally the DoubleAngleLayer (see section IV) to compute the orientations from the double-angle representation. A graphical representation is shown in Fig. 4.

The optional DSPP module draws inspiration from DeepLabv3+ [68] and utilizes 256 filters of size 3×3 with padding and ReLU activation for each dilation rate in $\{1, 6, 12, 18\}$, followed by batch normalization. The resulting 1024 features are concatenated and reduced to 256 features by 1×1 convolution with ReLU activation.

Four parametric basic convolution blocks were defined for the encoder and decoder levels (Fig. 14): *SimpleC* (f), *DoubleC*(f) (inspired by U-net [67]), *ResNetC*(f, n_{id}) (inspired by ResNet [66]), and *ConvNextC*(f, n_b) (inspired by ConvNext [69]). These blocks were combined to form the 12 base architectures listed in Table 5.

C. CANDIDATE ARCHITECTURES

Each of the 12 base architectures could be further customized with various options:

- 2×2 max pooling or 3×3 convolution with stride two for downsampling,
- 2×2 upsampling or 3×3 transposed convolution with stride two for upsampling,
- inclusion or exclusion of skip connections,
- inclusion or exclusion of the DSPP module.

This resulted in a total of 192 diverse candidate architectures.

D. LOSS FUNCTION SELECTION

Three potential loss function were evaluated: Mean Square Error (MSE), Mean Absolute Error (MAE) and Huber with parameter δ (HUBER$_\delta$):

$$\mathrm{MSE}\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}, \mathbf{S}\right) = \frac{\sum_{i,j}\left(\mathbf{S}_{i,j} \cdot d\phi\left(\boldsymbol{\Theta}_{i,j}, \hat{\boldsymbol{\Theta}}_{i,j}\right)\right)^2}{w \cdot h} \quad (10)$$

$$\mathrm{MAE}\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}, \mathbf{S}\right) = \frac{\sum_{i,j}\left|\mathbf{S}_{i,j} \cdot d\phi\left(\boldsymbol{\Theta}_{i,j}, \hat{\boldsymbol{\Theta}}_{i,j}\right)\right|}{w \cdot h} \quad (11)$$

$$\mathrm{HUBER}_\delta\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}, \mathbf{S}\right) = \frac{\sum_{i,j} Huber\left(\mathbf{S}_{i,j} \cdot d\phi\left(\boldsymbol{\Theta}_{i,j}, \hat{\boldsymbol{\Theta}}_{i,j}\right), \delta\right)}{w \cdot h} \quad (12)$$

TABLE 6. Models with RMSD < 7° on the "Bad" dataset.

Base architecture	Skip connections	Downsampling	Upsampling	DSPP	Loss function	RMSD ("Good")	RMSD ("Bad")
S05	Yes	2 × 2 max pooling	2 × 2 upsampling	No	$\text{HUBER}_{0.03}$	4.10	6.39
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	No	$\text{HUBER}_{0.06}$	4.15	6.62
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	Yes	$\text{HUBER}_{0.06}$	4.08	6.66
S05	Yes	2 × 2 max pooling	2 × 2 upsampling	No	MAE	4.09	6.70
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	Yes	$\text{HUBER}_{0.09}$	4.15	6.71
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	No	MAE	4.11	6.77
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	No	$\text{HUBER}_{0.09}$	4.05	6.79
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	Yes	MAE	4.10	6.80
S05	Yes	2 × 2 max pooling	2 × 2 upsampling	No	$\text{HUBER}_{0.09}$	4.12	6.80
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	Yes	$\text{HUBER}_{0.03}$	4.13	6.82
S05	Yes	2 × 2 max pooling	2 × 2 upsampling	No	$\text{HUBER}_{0.06}$	4.12	6.83
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	No	MSE	4.10	6.84
S20	Yes	2 × 2 max pooling	2 × 2 upsampling	No	$\text{HUBER}_{0.03}$	4.15	6.88

TABLE 7. Results of the ablation study.

Element removed	RMSD ("Good")	RMSD ("Bad")
None	4.10	6.39
First convolution layer in the Head block	4.12	6.52
Skip connection from the input segmentation mask to the Head block	4.20	6.91
DoubleAngleLayer (modifying the previous convolution layer to produce a single-channel output)	4.67	8.30
5 × 5 filters (replaced by 3 × 3 filters) in the convolution blocks	4.14	9.30
Data augmentation	4.27	10.73
Batch normalization in the convolution blocks	4.25	12.58

where $\boldsymbol{\Theta}$ are the estimated orientations, $\hat{\Theta}$ the ground truth orientations, $\mathbf{S}$ the ground truth segmentation mask, $d\phi\left(\theta, \hat{\theta}\right)$ is the difference between two orientations defined in equation (9), and $Huber\ (x, \delta)$ is the Huber loss [70]:

$$Huber\ (x, \delta) = \begin{cases} \frac{1}{2}x^2 & |x| \leq \delta \\ \delta(|x| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{13}$$

The HUBER_δ loss was evaluated with three different values for δ: 0.03, 0.06, e 0.09. Five loss functions were then considered for each of the 192 architectures, for a total of 960 models.

E. EVALUATION AND ANALYSIS

Each model was trained on the 400 FVC fingerprints and tested on the FOE-TEST benchmark (see section V-A). The entire experiment was conducted on a PC with an NVIDIA GeForce RTX™3080 Ti GPU, taking approximately ten days.

The graph in Fig. 15 shows one point for each of the 960 models evaluated: the x and y coordinates of each point are the corresponding RMSD on the "Bad" and "Good" dataset of the FOE-TEST benchmark, respectively. The graph highlights significant performance variations among the models: the RMSD on the "Good" dataset ranges from 4.05° to 12.34°, while the RMSD on the "Bad" dataset ranges from 6.39° to 22.02°. However, most models achieve an RMSD lower than 7° and 15° on the "Good" and "Bad" dataset, respectively. Further analysis focused on this range.

The four graphs in Fig. 16 depict the same points from Fig. 15 with different labels based on a specific variable. This allows for analysis from different perspectives. Some observations emerge:

- Skip connections are crucial for accuracy, especially for good quality fingerprints.
- Models based on the *SimpleC* convolution block (S01, S05, and S20) appear more effective, particularly when combined with skip connections.
- 2 × 2 max pooling seems to outperform stride-2 convolution for downsampling.
- 2 × 2 upsampling appear slightly better than transposed convolution.

These observations are reinforced by analyzing the most accurate models on the "Bad" dataset (table 6). All models in the table are based on either S05 or S20, incorporating skip connections, 2 × 2 max pooling, and simple 2 × 2 upsampling. The final choice between the base models (S05 and S20) prioritized the architecture with fewer parameters (S05), adhering to the KISS principle. For the same reason, the DSPP module was excluded. As for the loss function, the one used by the model with the lowest RMSD on the "Bad" set ($\text{HUBER}_{0.03}$) was selected.

F. SELECTED MODEL AND ABLATION STUDY

The selected model is therefore characterized by:

- S05 base architecture,
- Skip connections included,
- 2 × 2 max pooling for downsampling,
- 2 × 2 upsampling,
- No DSPP module,
- $\text{HUBER}_{0.03}$ loss function.

As a final experiment to validate the selected model's simplicity and the essentiality of its components, several ablation tests were performed. The model was evaluated after removing each of the following elements individually:

- The first convolution layer of the Head block (16 5×5 filters).
- The individual skip connection between then input segmentation mask and the Head block.
- The DoubleAngleLayer (forcing the network to learn the final orientations directly instead of using the double-angle representation).
- The 5×5 filters in the convolution blocks (replacing them with 3×3 filters).
- Data augmentation on the training fingerprints.
- The batch normalization layers in the *SimpleC* convolution blocks.

Table 7 shows the results of this ablation study. All evaluated models exhibited performance degradation, albeit to varying degrees. This confirms that all the considered components are essential, and the model selected for SNFOE is the simplest possible configuration with optimal performance.

ACKNOWLEDGMENT

The author is grateful to Dr. Davide Maltoni for his insightful comments.

REFERENCES

[1] D. Maltoni, D. Maio, A. K. Jain, and J. Feng, *Handbook of Fingerprint Recognition*. Cham, Switzerland: Springer, 2022.

[2] D. Maltoni and R. Cappelli, "Fingerprint recognition," in *Handbook of Biometrics*, A. K. Jain, P. Flynn, and A. A. Ross, Eds. Boston, MA, USA: Springer, 2008, pp. 23–42.

[3] T. Dalzell, *The Routledge Dictionary of Modern American Slang and Unconventional English*. Evanston, IL, USA: Routledge, 2009.

[4] Wikipedia Contributors. (2023). *KISS Principle—Wikipedia, The Free Encyclopedia*. [Online]. Available: https://en.wikipedia.org/w/index.php?title=KISS_principle

[5] W. Bian, D. Xu, Q. Li, Y. Cheng, B. Jie, and X. Ding, "A survey of the methods on fingerprint orientation field estimation," *IEEE Access*, vol. 7, pp. 32644–32663, 2019.

[6] M. Kass and A. Witkin, "Analyzing oriented patterns," *Comput. Vis., Graph., Image Process.*, vol. 36, no. 1, p. 133, Oct. 1986.

[7] A. R. Rao and R. C. Jain, "Computerized flow field analysis: Oriented texture fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 693–709, Jul. 1992.

[8] N. K. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern Recognit.*, vol. 28, no. 11, pp. 1657–1672, Nov. 1995.

[9] M. Donahue, "On the use of level curves in image analysis," *Comput. Vis. Image Understand.*, vol. 57, no. 2, pp. 185–203, Mar. 1993.

[10] A. M. Bazen and S. H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 905–919, Jul. 2002.

[11] Y. Mei, H. Sun, and D. Xia, "A gradient-based combined method for the computation of fingerprints' orientation field," *Image Vis. Comput.*, vol. 27, no. 8, pp. 1169–1177, Jul. 2009.

[12] R. M. Stock and C. W. Swonger, "Development and evaluation of a reader of fingerprint minutiae," Cornell Aeronaut. Lab., Tech. Rep. XM-2478-X-1:13-17, 1969.

[13] B. M. Mehtre, N. N. Murthy, S. Kapoor, and B. Chatterjee, "Segmentation of fingerprint images using the directional image," *Pattern Recognit.*, vol. 20, no. 4, pp. 429–435, Jan. 1987.

[14] Y. He, J. Tian, X. Luo, and T. Zhang, "Image enhancement and minutiae matching in fingerprint verification," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1349–1360, Jun. 2003.

[15] M. A. Oliveira and N. J. Leite, "A multiscale directional operator and morphological tools for reconnecting broken ridges in fingerprint images," *Pattern Recognit.*, vol. 41, no. 1, pp. 367–377, Jan. 2008.

[16] B. G. Sherlock, "Computer enhancement and modeling of fingerprint images," in *Automatic Fingerprint Recognition Systems*, N. Ratha and R. Bolle, Eds. New York, NY, USA: Springer, 2004, pp. 87–112.

[17] L. Ji and Z. Yi, "Fingerprint orientation field estimation using ridge projection," *Pattern Recognit.*, vol. 41, no. 5, pp. 1491–1503, May 2008.

[18] W. Bian, S. Ding, and Y. Xue, "Combining weighted linear project analysis with orientation diffusion for fingerprint orientation field reconstruction," *Inf. Sci.*, vol. 396, pp. 55–71, Aug. 2017.

[19] L. Hong, A. Jian, S. Pankanti, and R. Bolle, "Fingerprint enhancement," in *Proc. 3rd IEEE Workshop Appl. Comput. Vis.*, Dec. 1996, pp. 202–207.

[20] T. Nakamura, M. Hirooka, H. Fujiwara, and K. Sumi, "Fingerprint image enhancement using a parallel ridge filter," in *Proc. 17th Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 536–539.

[21] P. Maponi, R. Piergallini, and F. Santarelli, "Fingerprint orientation refinement through iterative smoothing," *Signal Image Process., Int. J.*, vol. 8, no. 5, pp. 29–43, Oct. 2017.

[22] T. Kamei and M. Mizoguchi, "Image filter design for fingerprint enhancement," in *Proc. Int. Symp. Comput. Vis.*, Nov. 1995, pp. 109–114.

[23] S. Chikkerur, V. Govindaraju, and A. N. Cartwright, "Fingerprint image enhancement using STFT analysis," in *Pattern Recognition and Image Analysis*. Berlin, Germany: Springer, 2005.

[24] K. G. Larkin, "Uniform estimation of orientation using local and nonlocal 2-D energy operators," *Opt. Exp.*, vol. 13, no. 20, pp. 8097–8121, 2005.

[25] X. Jiang, M. Liu, and A. C. Kot, "Reference point detection for fingerprint recognition," in *Proc. 17th Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 540–543.

[26] M. Liu, X. Jiang, and A. C. Kot, "Fingerprint reference point detection," in *Biometric Authentication*. Berlin, Germany: Springer, 2004, pp. 272–279.

[27] E. Zhu, J. Yin, C. Hu, and G. Zhang, "A systematic method for fingerprint ridge orientation estimation and image segmentation," *Pattern Recognit.*, vol. 39, no. 8, pp. 1452–1472, Aug. 2006.

[28] Q. Zhang and H. Yan, "Fingerprint orientation field interpolation based on the constrained Delaunay triangulation," *Int. J. Inf. Syst. Sci.*, vol. 3, pp. 438–452, Jul. 2007.

[29] B. G. Sherlock and D. M. Monro, "A model for interpreting fingerprint topology," *Pattern Recognit.*, vol. 26, no. 7, pp. 1047–1055, Jul. 1993.

[30] P. R. Vizcaya and L. A. Gerhardt, "A nonlinear orientation model for global description of fingerprints," *Pattern Recognit.*, vol. 29, no. 7, pp. 1221–1231, Jul. 1996.

[31] J. Zhou and J. Gu, "Modeling orientation fields of fingerprints with rational complex functions," *Pattern Recognit.*, vol. 37, no. 2, pp. 389–391, Feb. 2004.

[32] J. Zhou and J. Gu, "A model-based method for the computation of fingerprints' orientation field," *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 821–835, Jun. 2004.

[33] J. Gu, J. Zhou, and C. Yang, "Fingerprint recognition by combining global structure and local cues," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1952–1964, Jul. 2006.

[34] J. Li, W.-Y. Yau, and H. Wang, "Constrained nonlinear models of fingerprint orientations with prediction," *Pattern Recognit.*, vol. 39, no. 1, pp. 102–114, Jan. 2006.

[35] S. Huckemann, T. Hotz, and A. Munk, "Global models for the orientation field of fingerprints: An approach based on quadratic differentials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1507–1519, Sep. 2008.

[36] C. Gottschlich, B. Tams, and S. Huckemann, "Perfect fingerprint orientation fields by locally adaptive global models," *IET Biometrics*, vol. 6, no. 3, pp. 183–190, May 2017.

[37] Y. Wang, J. Hu, and D. Phillips, "A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 573–585, Apr. 2007.

[38] Y. Wang and J. Hu, "Global ridge orientation modeling for partial fingerprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 72–87, Jan. 2011.

[39] A. Tashk, M. S. Helfroush, and M. Muhammadpour, "Improvement of fingerprint orientation estimation by a modification of fingerprint orientation model based on 2D Fourier expansion (M-FOMFE)," in *Proc. 2nd Int. Conf. Comput., Control Commun.*, Feb. 2009, pp. 1–6.
[40] X. Tao, X. Yang, K. Cao, R. Wang, P. Li, and J. Tian, "Estimation of fingerprint orientation field by weighted 2D Fourier expansion model," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1253–1256.
[41] S. Ram, H. Bischof, and J. Birchbauer, "Modelling fingerprint ridge orientation using legendre polynomials," *Pattern Recognit.*, vol. 43, no. 1, pp. 342–357, Jan. 2010.
[42] S. Jirachaweng, Z. Hou, W.-Y. Yau, and V. Areekul, "Residual orientation modeling for fingerprint enhancement and singular point detection," *Pattern Recognit.*, vol. 44, no. 2, pp. 431–442, Feb. 2011.
[43] M. Liu, S. Liu, and Q. Zhao, "Fingerprint orientation field reconstruction by weighted discrete cosine transform," *Inf. Sci.*, vol. 268, pp. 65–77, Jun. 2014.
[44] F. Turroni, D. Maltoni, R. Cappelli, and D. Maio, "Improving fingerprint orientation extraction," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1002–1013, Sep. 2011.
[45] W. Bian, Y. Luo, D. Xu, and Q. Yu, "Fingerprint ridge orientation field reconstruction using the best quadratic approximation by orthogonal polynomials in two discrete variables," *Pattern Recognit.*, vol. 47, no. 10, pp. 3304–3313, Oct. 2014.
[46] Z. Hou and W.-Y. Yau, "A variational formulation for fingerprint orientation modeling," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1626–1629.
[47] K. Cao, J. Liang, and J. Tian, "A div-curl regularization model for fingerprint orientation extraction," in *Proc. IEEE 5th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2012, pp. 231–236.
[48] K. A. Nagaty, "On learning to estimate the block directional image of a fingerprint using a hierarchical neural network," *Neural Netw.*, vol. 16, no. 1, pp. 133–144, Jan. 2003.
[49] K.-C. Lee and S. Prabhakar, "Probabilistic orientation field estimation for fingerprint enhancement and verification," in *Proc. Biometrics Symp.*, Sep. 2008, pp. 41–46.
[50] S. Ram, H. Bischof, and J. Birchbauer, "Active fingerprint ridge orientation models," in *Advances in Biometrics*. Berlin, Germany: Springer, 2009.
[51] J. Feng, J. Zhou, and A. K. Jain, "Orientation field estimation for latent fingerprint enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 925–940, Apr. 2013.
[52] C. Chen, J. Feng, and J. Zhou, "Multi-scale dictionaries based fingerprint orientation field estimation," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2016, pp. 1–8.
[53] S. Liu, M. Liu, and Z. Yang, "Sparse coding based orientation estimation for latent fingerprints," *Pattern Recognit.*, vol. 67, pp. 164–176, Jul. 2017.
[54] K. Cao, E. Liu, and A. K. Jain, "Segmentation and enhancement of latent fingerprints: A coarse to fine RidgeStructure dictionary," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1847–1859, Sep. 2014.
[55] X. Yang, J. Feng, and J. Zhou, "Localized dictionaries based orientation field estimation for latent fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 955–969, May 2014.
[56] Q. Yin, J. Feng, J. Lu, and J. Zhou, "Orientation field estimation for latent fingerprints by exhaustive search of large database," in *Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Oct. 2018, pp. 1–9.
[57] K. Cao and A. K. Jain, "Latent orientation field estimation via convolutional neural network," in *Proc. Int. Conf. Biometrics (ICB)*, May 2015, pp. 349–356.
[58] Y. Tang, F. Gao, J. Feng, and Y. Liu, "FingerNet: An unified deep network for fingerprint minutiae extraction," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 108–116.
[59] P. Schuch, S.-D. Schulz, and C. Busch, "ConvNet regression for fingerprint orientations," in *Image Analysis*. Cham, Switzerland: Springer, 2017.
[60] P. Schuch, S.-D. Schulz, and C. Busch, "Deep expectation for estimation of fingerprint orientation fields," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 185–190.
[61] Z. Qu, J. Liu, Y. Liu, Q. Guan, C. Yang, and Y. Zhang, "OrieNet: A regression system for latent fingerprint orientation field extraction," in *Artificial Neural Networks and Machine Learning*. Cham, Switzerland: Springer, 2018.
[62] W. J. Wong and S.-H. Lai, "Multi-task CNN for restoring corrupted fingerprint images," *Pattern Recognit.*, vol. 101, May 2020, Art. no. 107203.
[63] Y. Duan, J. Feng, J. Lu, and J. Zhou, "Orientation field estimation for latent fingerprints with prior knowledge of fingerprint pattern," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Aug. 2021, pp. 1–8.
[64] R. Cappelli, "Unveiling the power of simplicity: Two remarkably effective methods for fingerprint segmentation," *IEEE Access*, vol. 11, pp. 144530–144544, 2023.
[65] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. London, U.K.: Pearson, 2018.
[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[67] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[68] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 20–29.
[69] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.
[70] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.
[71] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2002, pp. 811–814.
[72] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain, "Performance evaluation of fingerprint verification systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 3–18, Jan. 2006.
[73] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti, and A. Mayoue, "Fingerprint and on-line signature verification competitions at ICB 2009," in *Advances in Biometrics*. Berlin, Germany: Springer, 2009.
[74] (2024). *FVC-onGoing: On-Line Evaluation of Fingerprint Recognition Algorithms*. Accessed: Feb. 26, 2024. [Online]. Available: https://biolab.csr.unibo.it/fvcongoing
[75] M. D. Garris and R. M. McCabe, "NIST special database 27: Fingerprint minutiae from latent and matching tenprint images," Nat. Inst. Standards Technol. (NIST), Tech. Rep. NISTIR 6534, 2000.
[76] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
[77] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–4.
[78] *OpenCV: Open Source Computer Vision Library*. Accessed: Apr. 17, 2024. [Online]. Available: https://opencv.org/
[79] *Keras*. Accessed: Apr. 17, 2024. [Online]. Available: https://keras.io/
[80] R. Cappelli, M. Ferrara, A. Franco, and D. Maltoni, "Fingerprint verification competition 2006," *Biometric Technol. Today*, vol. 15, nos. 7–8, pp. 7–9, Jul. 2007.
[81] C. I. Watson, "NIST special database 14: Fingerprint card pairs 2," Adv. Syst. Division, Image Recognit. Group, Nat. Inst. Standards Technol., USA, 1993. [Online]. Available: https://dl.acm.org/doi/10.1016/j.patrec.2007.04.003
[82] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1083–1090.
[83] Q. Yin, J. Feng, J. Lu, and J. Zhou, "Joint estimation of pose and singular points of fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1467–1479, 2021.

RAFFAELE CAPPELLI received the degree in computer science from the University of Bologna, in 1998, and the Ph.D. degree, in 2002.

He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Bologna, and a member of the Biometric System Laboratory in Cesena. He is one of the organizers of the fingerprint verification competitions (FVC2000–FVC2006 and FVC-onGoing) and one of the authors of the *Synthetic Fingerprint Generator* (SFinGe).

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement