Progressive and Corrective Feedback for Latent Fingerprint Enhancement Using Boosted Spectral Filtering and Spectral Autoencoder

KITTIPOL HORAPONG, KITTINUTH SRISUTHEENON, AND VUTIPONG AREEKUL, (Member, IEEE)
Kasetsart Signal and Image Processing Laboratory (KSIP Lab), Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand
Corresponding author: Vutipong Areekul (fengvpa@ku.ac.th)
This work was supported in part by the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, and in part by the Siew-Sngiem Karnchanachari Research Leadership and Young Professorship Awards.

ABSTRACT The objective of this research is to design an efficient algorithm that can successfully enhance a targeted latent fingerprint from various complex backgrounds under an uncontrolled environment. Most algorithms in literature exploited dictionary learning schemes and deep learning architectures to capture latent fingerprints from complicated backgrounds and noise. However, an algorithm learned from other high-quality fingerprint images may not solve all possible cases within a given unseen image. We propose a new feedback framework to distinguish latent fingerprints from complex backgrounds and gradually improve friction-ridge quality using the information provided inside the given unseen image. We combine two efficient mechanisms. The first mechanism enhances high-quality areas in priority and feeds the enhanced areas back to improve the quality of latent fingerprints in the nearby area. The second mechanism is to verify that the first mechanism works correctly by detecting anomalously enhanced fingerprint patterns. The second mechanism employs a spectral autoencoder that learns from good fingerprint spectra in the frequency domain. The anomalous fingerprint area is sent back to the first mechanism for further improving the enhanced result. We benchmark the proposed algorithm against available state-of-the-art algorithms using two fingerprint matching systems (one commercial off-the-shelf and one open-source) on two public latent fingerprint databases. The experimental results show that the proposed algorithm outperforms most state-of-the-art algorithms in the literature.

INDEX TERMS Fingerprint recognition, image enhancement, image filtering, image forensics, image restoration, machine learning.

I. INTRODUCTION A latent fingerprint must contain several crucial features such as singular points, friction ridges, and minutiae in order to identify a prime suspect [1]. Even though several latent fingerprints are carefully collected from a crime scene and extensively developed in an advanced laboratory, these latent fingerprints may fail to identify or verify any suspect due to four uncontrolled conditions. First, we cannot control the quality and quantity of latent fingerprints unintentionally left at a crime scene. Hence, latent fingerprints are always incomplete, low-quality, and partial. Second, we cannot control the surface and background where fingerprints are deposited. So, we collect latent fingerprint images with uncontrolled background patterns, and friction ridge quality depends on the smoothness of a surface. Third, we cannot control dust, grease, and other substances that can contaminate latent fingerprints. Therefore, these latent fingerprint images are always noisy. Finally, the fingerprint can be overlapped with other fingerprints by multiple touches. We need to separate these overlapped fingerprints before sending each of them to identify the suspect. These four uncontrolled conditions have made our task very difficult for identifying the prime suspect from available latent fingerprints. Hence every genuine feature existing in the latent fingerprint is significant. The goal of this research is to restore...
and preserve any critical features that are hidden in latent fingerprints.

Most fingerprint enhancement methods in the handbook of fingerprint recognition [2] were designed explicitly for rolled and slapped fingerprints. These fingerprints are obtained from various live-scan fingerprint sensors under a controlled environment. The fingerprint quality is usually quite good, with no interfering background. On the contrary, this is not the case for latent fingerprints. Latent fingerprints are always low-quality and partial. In addition, latent fingerprints usually compose of complex background and noise. Hence most methods in [2] fail to restore and preserve essential parameters for enhancement. For example, the classic Gabor filtering method, proposed by Hong et al. [3], suffers from a failure to estimate orientation and frequency parameters correctly. Therefore we cannot rely on these enhanced results.

Since 2008, several researchers have shifted their research to solve the latent fingerprint enhancement enhancement problem [4], [5]. Several latent fingerprint enhancement algorithms have emerged from machine learning tools. The most popular approaches are based on dictionary learning [6]–[13] and deep learning [14]–[22]. The key of these learning approaches is to learn from a large set of high-quality fingerprint images and to use these learning models to enhance friction ridges of low-quality latent fingerprint images. Lately, these learning approaches have been succeeded in enhancing latent fingerprints and significantly improved the identification rate.

A. RESEARCH MOTIVATION
Most previously mentioned algorithms in literature perform latent fingerprint enhancement by using learning models in the spatial domain [6]–[12], [14]–[22]. On the contrary, working in the frequency domain provides us some advantages. Firstly, the Fourier transform decomposes a fingerprint image into a spectral magnitude image and a spectral phase image. We can extract and enhance friction ridges easier in the spectral magnitude image because friction ridge spectra peak and pack in the frequency domain. Secondly, we can eliminate unwanted spectra that unrelated to friction ridges in this spectral magnitude image. Thirdly, the spectral phase image is directly related to minutiae locations in the latent fingerprint image. We leave this spectral phase image untouched to preserve the locations of genuine minutiae in the original image. Hence we only manipulate the spectral magnitude image. Our previous work proposed a dictionary learning in the frequency domain, called spectral dictionary [13]. The enhanced results of this approach are auspicious. Hence our motivation is to explore the possibility of a more complicated learning model in the frequency domain to solve the latent fingerprint enhancement problem.

Performing fingerprint enhancement in the frequency domain gives us another significant advantage. We can design a progressive feedback mechanism. An adaptive boosted spectral filtering (ABSF) technique [24] has succeeded in enhancing rolled and slapped fingerprints because its concept has taken on this mechanism. The ABSF algorithm initially enhances friction ridges with high-quality first in the frequency domain. Then these locally enhanced results are fed back to the original image to improve low-quality friction ridges nearby in the spatial domain. The overlapped block-based Fourier transform allows us to diffuse good spectra into bad spectra, resulting in progressive feedback enhancement. Even though the ABSF can significantly improve the very low-quality of clear friction ridges, it has a problem with the noisy friction ridges on the complex background. If the initial block contains friction ridges with noise on the complicated background, the enhanced result may contaminate with enhanced noise or background instead. Moreover, the feedback operation propagates the enhanced error to the nearby blocks resulting in enhancement failure of the entire image. Hence, the original ABSF algorithm is not suitable for directly enhancing the latent fingerprint.

We aim to bring latent fingerprint enhancement to another level. To achieve this goal, we need to combine two concepts: the progressive feedback mechanism of ABSF and an efficient learning model in the frequency domain. We have done preliminary work with the progressive feedback mechanism on latent fingerprint enhancement in [26]. This approach has shown some impressive results by enhancing corrupted friction ridges in complex backgrounds. However, this algorithm requires human assistance for the manual selection of the initial block location. In addition, this algorithm is sensitive to error propagation. If the algorithm fails to enhance the genuine friction ridges, the enhanced error propagates to the nearby blocks resulting in a domino effect. In another preliminary work [27], we have tried to combine two concepts. We employed an autoencoder to predict a corresponding matched filter with the progressive feedback mechanism. The results are still more room for improvement. In this paper, we provide a novel framework for latent fingerprint enhancement.

B. RESEARCH CONTRIBUTION
There are three key contributions of this paper as follows.
1) We introduce a novel framework for latent fingerprint enhancement, which fully exploits a progressive feedback mechanism incorporated with a new learning model for anomalous fingerprint pattern detection. The seamless integration of two independent mechanisms brings latent fingerprint enhancement to another level.
2) We develop the progressive feedback mechanism for handling automatic latent fingerprint enhancement. We introduce an automatic initial block localization method, which can indicate multiple initial locations. This method helps reduce the risk of initial start at the wrong location resulting in enhanced error propagation.
3) We propose a new learning model based on a stacked autoencoder, called spectral autoencoder, in the frequency domain. We train this spectral autoencoder by a large set of enhanced spectral patches of high-quality
fingerprints. This spectral autoencoder can detect anomalous fingerprint patterns from outputs of the progressive feedback mechanism. The error locations are detected and sent back for re-enhancement in the next iteration. This learning model provides a superior scheme for error detection and error correction of the proposed framework.

II. RELATED WORK

Most latent fingerprint enhancement algorithms have exploited the conventional Gabor filtering concept. These algorithms have difficulty finding reliable parameters such as ridge orientation and frequency for Gabor filters due to corrupted friction ridges in latent fingerprint images. Karimi-Ashtiani and Kuo [4] firstly addressed a latent fingerprint enhancement problem in 2008. They used the Gabor filters to enhance latent fingerprints. However, their parameter estimation method is not suitable for latent fingerprint problems. In 2011, Yoon et al. [5] proposed a robust orientation field estimation for latent fingerprint enhancement using short-time Fourier transform (STFT) and randomized random sample consensus (R-RANSAC) algorithm. Since then, the proposed methods have shifted to exploit machine learning tools to solve this problem. We can categorize algorithms in literature into three main approaches: dictionary learning, deep learning, and progressive feedback. Table 1 shows selected algorithms that specifically address the latent fingerprint enhancement problem. Note that we list only algorithms that we have available enhanced results for our benchmark comparison.

A. DICTIOnARY LEARNING APPROACH

The concept of the dictionary learning approach is to estimate the reliable orientation and frequency parameters for Gabor filters. The dictionary learning can learn good orientation and frequency parameters from high-quality fingerprints. In 2013, Feng et al. [6] firstly proposed a global dictionary of orientation field estimation, called GlobalDict, to retrieve local ridge orientation on latent fingerprints. The estimated local orientation is then applied to the Gabor filter with fixed frequency at 1/9 cycles/pixel and fixed standard deviation at 4. Yang et al. [7] improved the performance of the previous work [6] using local dictionaries, called LocalDict. They presented a location-dependent dictionary relative to the finger pose using different dictionary sets for different fingerprint positions. Then, they used the estimated local orientation with fixed frequency and standard deviation for the Gabor filter parameters. Cao et al. [8] introduced two dictionary sets, including coarse and fine friction ridge structures, called RidgeDict, for ridge frequency and orientation estimation. They applied both estimated parameters to Gabor filters with a fixed standard deviation at 4. Liu et al. [9] presented a dictionary characterized by Gabor function with varying orientations, frequencies, and phases. They used sparse representation with the multi-scale Gabor dictionaries to reconstruct a fingerprint patch. In addition, Chen et al. [10] presented improved multi-scale dictionaries for orientation estimation by covering larger fingerprint areas. Liu et al. [11] developed multi-scale dictionaries with iterative orientation estimation. Xu et al. [12] combined Gabor dictionaries with minutiae dictionaries. On the contrary, Chaidee et al. [13] designed the dictionary learning from Gabor spectral responses via Gabor filter banks and sparse representation in the frequency domain. Instead of using the Gabor filter, the dictionary, called SpectralDict, predicted the shape of the filter in the frequency domain for enhancing latent fingerprints.

B. DEEP LEARNING APPROACH

Since 2017, the deep learning approach has gained attention to solve the latent fingerprint enhancement problem. The deep learning approach aims to transform a latent fingerprint image into an enhanced fingerprint image directly. Tang et al. [14] proposed a unified network architecture named FingerNet, which combines latent fingerprint segmentation, orientation estimation, enhancement, and minutiae extraction. This architecture contains two deep convolutional neural networks (CNN), one for orientation estimation and segmentation and another for minutiae extraction. Then the estimated orientation parameters with fixed frequency shaped Gabor filters for latent fingerprint enhancement. Svoboda et al. [15] and Li et al. [16] independently proposed a deep autoencoder that provided an end-to-end solution for latent fingerprint enhancement. Qian et al. [17] proposed a densely connected UNet (DenseUNet) to produce a high-quality fingerprint patch, not the whole image. Then, the network iteratively enhanced the whole image. Recently, Liu and Qian [18] introduced Deep Nested UNet architecture, called DN-UNets, for latent fingerprint segmentation and enhancement. This network combines nested UNets with dense skip connections, transforming a whole latent fingerprint image into an enhanced image directly.

On the other hand, some researchers have exploited the potential of using generative adversarial networks (GAN) for latent fingerprint enhancement. Dabouei et al. [19] showed that a conditional GAN could reconstruct partial latent fingerprints. Liu et al. [20] introduced a cooperative orientation generative adversarial network (COOGAN) to transform latent fingerprint images using a shared representation of ridge enhancement and orientation features. Xu et al. [21] presented a GAN-based data augmentation in their network structure. The synthesized data can facilitate the network to translate a latent to enhanced fingerprint effectively. Huang et al. [22] proposed a progressive GAN for learning the enhanced result and orientation field. They trained both generator and discriminator with progressively growing from low-resolution.

C. PROGRESSIVE FEEDBACK APPROACH

A progressive feedback approach was first introduced by Sutthiwichaiporn et al. [23]. The ABSF technique [24] fully exploited this approach for enhancing rolled and slapped fingerprints. Deerada et al. [25] applied this approach to improve
TABLE 1. Comparative summary of selected latent fingerprint enhancement algorithms.

| Algorithm               | Objectives                           | Approach                                      | Training database | Benchmark fingerprint matching tools | Benchmark databases                                      |
|-------------------------|--------------------------------------|-----------------------------------------------|-------------------|---------------------------------------|---------------------------------------------------------|
| GlobalDict [6]          | Enhancement                          | Orientation field estimation using dictionary learning | NIST SD4          | VeriFinger SDK 6.2                    | NIST SD27 (258 latent prints), NIST SD14 and SD27 (27,258 rolled prints) |
| LocalDict [7]           | Enhancement                          | Orientation field estimation using dictionary learning | NIST SD4          | VeriFinger SDK 6.2                    | NIST SD27 (258 latent prints), NIST SD14 and SD27 (27,258 rolled prints) |
| RidgeDict [8]           | Segmentation and Enhancement          | Orientation and frequency estimation using dictionary learning | NIST SD4          | 1. VeriFinger SDK 6.3  
2. COTS tenprint matcher  
3. COTS latent matcher | NIST SD27 (258 latent prints), NIST SD14, SD27, and WVU DB (449 latent prints) |
| Spectral Dct [13]       | Enhancement                          | Spectral filtering using dictionary learning | NIST SD4          | 1. VeriFinger SDK 7.1  
2. COTS latent matcher. | NIST SD27 (258 latent prints), NIST SD14 and SD27 (27,258 rolled prints) |
| FingerNet [14]          | Segmentation and Minutiae Extraction | Orientation field estimation using deep CNN | NIST SD14 and background patterns from NIST SD27 | 1. VeriFinger SDK 6.5  
2. Open-source system using MINDTCT minutiae extractor and Bozorth3 matcher | NIST SD27 (258 latent prints), NIST SD14 and SD27 (27,258 rolled prints) |
| Deep NestedUNET [18]    | Segmentation and Enhancement          | Image-to-image transformation using deep CNN | NIST SD4          | VeriFinger SDK 10.0 [30] | NIST SD27 (258 latent prints), NIST SD14 and SD27 (27,258 rolled prints) |
| Semi-Progressive Enh. [26] | Enhancement                           | ABSF [24] with a manual initial block         | No training needed | VeriFinger SDK 10.0 [30] | ITTD-MOLF DB4 (1,000 latent prints), ITTD-MOLF DB2 (1,000 rolled prints) (The ITTD-MOLF is tested on VeriFinger only.) |
| 2-stage Progressive Enh. [27] | Enhancement                           | ABSF [24] with matched filter predicted by the autoencoder | NIST SD4          | VeriFinger SDK 10.0 [30] | NIST SD27 (4,400 latent prints), NIST SD14 and ITTD-MOLF DB3_A (31,000 rolled prints) |
| Progressive and Corrective Feedback (PCF) (Proposed) | Enhancement                           | ABSF [24] + anomaly detection using spectral autoencoder + automatic initial blocks localization | NIST SD4          | 1. VeriFinger SDK 10.0 [30]  
2. Open-source system using MINDTCT minutiae extractor [36] and MCC 2.0 matcher [41] | NIST SD27 (258 latent prints), NIST SD14 and SD27 (27,258 rolled prints) |

the quality of latent fingerprints for reference point detection. This method had shown the potential of latent fingerprint enhancement. Then, Srisutheenon et al. [26] firstly applied this concept for latent fingerprint problems. They designed a simple and effective algorithm to shape a local matched filter. The filtering process firstly enhanced the highest quality block. Then, the enhanced block was inserted back to the input image to improve the quality of fingerprint spectra of neighboring blocks nearby. The drawback of this method is that the highest quality block was manually selected by human assistance. Horapong et al. [27] combined the progressive feedback method [26] with a learning model in a two-stage design. The first stage iteratively applied a matched filter in the high-quality fingerprint region. The second stage used an autoencoder to predict filters in the low-quality region. These preliminary works led us to design a new framework for better performance.

III. PROPOSED METHOD

We introduce a new framework for solving a latent fingerprint enhancement problem. The new framework exploits a feedback mechanism for the handling of this complicated problem. The framework consists of three main processes; A, B, and C, as shown in Fig. 1. The first process, A, is to find the best locations for starting the following enhancement sequence. The best locations should contain the clearest friction ridges in an input latent fingerprint image. The second process, B, adopts the progressive feedback mechanism of ABSF [24] and pushes it to another level. The goal of process B is to enhance the high-quality fingerprint blocks first and feed the enhanced fingerprint blocks back to improve the low-quality fingerprint area nearby. This process gradually improves the quality of latent fingerprint block-by-block until the entire fingerprint segment is enhanced. The third process, C, is to detect anomalous fingerprint patterns. This
process examines the enhanced results and pinpoints the abnormal locations of the enhanced image. Once anomalous blocks are detected, this process will provide feedback of incorrectly enhanced block positions to process B re-enhance again. This process C provides corrective feedback for better enhancement results. Each process is explained as follows.

A. PROCESS A: INITIAL BLOCK LOCALIZATION
Similar to previous works [24], the proposed algorithm needs to start at the best-quality genuine fingerprint location of an input latent fingerprint image. Then the following process can enhance and propagate the genuine spectrum to a nearby area to improve the overall quality of an enhanced output. However, if we start at the corrupted fingerprint location, the enhanced output may be irrelevant to the targeted fingerprint. Moreover, the output result may be contaminated by enhanced background and noise instead. In this work, instead of starting at only one block as [24], [26], we propose an algorithm that can start at multiple blocks, which can mitigate the problem of starting at the wrong location. One of the successes of the proposed algorithm depends on this process. This initial block localization process composes of nine steps shown in Fig. 2. We explain the detail for each step as follows.

1) THE 1ST STEP: BLOCK PARTITIONING
We partition an input image into non-overlapped blocks, \( b(m, n) \), with a block size of \( 16 \times 16 \) pixels. The \( b(m, n) \) is the block at row \( m \) and column \( n \) of this partitioning. We interest in only the area of the manual segment of the input latent fingerprint image. Assume that \( b(m, n) \in BOI_1 \), where the \( BOI_1 \) is a set of blocks of interest covering the manual segment of the input latent fingerprint image.

2) THE 2ND STEP: SMOOTH INTENSITY REJECTION
The second step aims to remove blocks with smooth intensity from the \( BOI_1 \). In general, the blocks with high-quality fingerprints should have a wide range of intensity values (grayscale 0-255). We count a number of the intensity-occurrence for each block in \( BOI_1 \). We reject the block whose intensity-occurrence is less than one-third of the maximum intensity-occurrence in \( BOI_1 \). The residual blocks are in \( BOI_2 \).

3) THE 3RD STEP: INTENSITY OUTLIER REJECTION
The third step aims to remove blocks with too-dark or too-bright intensities in \( BOI_2 \). The blocks with too-dark intensity tend to be smudge, and the blocks with too-bright usually contain no fingerprint. Assume that the block \( b(m, n) \in BOI_2 \) and \( \mu_{b(m,n)} \) is an average intensity of pixels within this block \( b(m, n) \). We calculate \( \mu_{b(m,n)} \) for every \( b(m, n) \) in \( BOI_2 \). We measure the mean (\( \mu_{BOI_2} \)), the standard deviation (\( \sigma_{BOI_2} \)), and the skewness (\( \vartheta_{BOI_2} \)) of all \( \mu_{b(m,n)} \). The three rejection conditions are defined as follows.

- If \( \vartheta_{BOI_2} > 0.1 \) and \( (\mu_{b(m,n)} \geq \mu_{BOI_2} + 1.5\sigma_{BOI_2}) \) or \( (\mu_{b(m,n)} \leq \mu_{BOI_2} - 1.25\sigma_{BOI_2}) \), we reject the \( b(m, n) \), or

FIGURE 2. Flowchart of process A: initial block localization.
The maximum spectral magnitude inside this bandwidth. We reject this block if one of the following conditions is true:

- If \(( R > 0.5)\) and \((\Delta \theta > 30^{\circ})\), we reject this block,
- If \(( R > 0.75)\) and \((\Delta \theta > 20^{\circ})\), we reject this block,
- If \(( R > 0.8)\) and \((\Delta \theta > 10^{\circ})\), we reject this block.

All remaining blocks that passed these conditions are in BOI_7.

8) THE 8TH STEP: HIGH CONTRAST EDGE SPECTRUM REJECTION

Sometimes strong spectral peaks inside fingerprint bandwidth are not related to a fingerprint pattern, but high-contrast edges in the background cause these high peaks in fingerprint bandwidth. In the frequency domain, the fingerprint spectral peaks are at their fundamental frequency and harmonics. On the contrary, the spectral shapes of the high-contrast edges are in the form of a “sinc” function. A center peak of the main lobe of the sinc function is at zero frequency. The first sidelobe peak and the second sidelobe peak of the sinc \((x)\) are located at \(x = 3\pi/2\) and \(x = 5\pi/2\), respectively. In the \(64 \times 64\) spectral patch, the locations of the first and second sidelobe peaks are dependent on the transition width of the high-contrast edge. We notice that the first sidelobe peak of the high contrast edge is usually in the fingerprint bandwidth from our experiments. We can distinguish the high-contrast edge spectra from the fingerprint spectra by detecting the second sidelobe peak. We suspect that these spectra are from high-contrast edges if we can detect the second sidelobe peak in the same direction as the first sidelobe peak. To implement this step, we measure the frequency distance, \(l\) (frequency point), from the center (zero frequency) to the highest peak in the fingerprint band. If this peak is the first sidelobe peak of the sinc function, the second sidelobe peak should be at \((5\pi/2)l\) or \(3\pi/2)l = 1.67l\). We search for the second sidelobe peak at \(1.67l\) within the same direction as the first sidelobe peak in the range of \(\pm 1^{\circ}\). If the second sidelobe peak exists, we reject this block. The residual blocks are in BOI_8.

9) THE 9TH STEP: INITIAL BLOCKS CALCULATION

This step is the final step of process A. We cluster the residual blocks in BOI_8 by grouping connected blocks using an 8-neighbors of a block connectivity operation. Then we calculate a centroid of each cluster. If the centroid is at a block, \((m, n)\), inside the cluster, this block is one of the initial blocks for the next process B. If the centroid is outside the cluster, we find the nearest block to the centroid for the
initial block for the following process. Fig. 3 demonstrates the examples of the residual blocks in BOI8 in the manual fingerprint segment. (Bottom row) The red blocks are the initial blocks obtained from process A.

B. PROCESS B: PROGRESSIVE FEEDBACK LATENT FINGERPRINT ENHANCEMENT

Process B is designed to enhance the targeted latent fingerprint block-by-block starting from the given initial blocks obtained by process A. The concept of this process is similar to the ABSF technique [24]. However, the ABSF technique is designed to handle rolled and slapped fingerprints that contain less noise without complex background. Therefore, we need to modify the ABSF technique to handle the latent fingerprint with background noises. Process B can be divided into three sub-processes: total variation decomposition, initial block enhancement, and iterative blocks enhancement and feedback. Each sub-process is explained in detail as follows.

1) TOTAL VARIATION DECOMPOSITION

The total variation (TV) decomposition is required to reduce high contrast edges in the input latent fingerprint image that create substantial interference within the fingerprint bandwidth in the frequency domain. We use a TV minimization [28] for the decomposition of cartoon and texture components. In this work, we use an anisotropic TV regularization (L1-norm) for TV minimization. That is,

\[
\arg\min_{f_c(x,y)} \lambda \| f_c(x,y) - f(x,y) \|_1 + TV_{\text{ani}}(f_c(x,y)),
\]

where \( f_c(x,y) \) and \( f(x,y) \) are the cartoon-component image and the input latent fingerprint image, respectively. \( \lambda \) is a regularization parameter that is set to 0.45 by our empirical result. \( TV_{\text{ani}}(f_c(x,y)) \) is the 2-D anisotropic total variation of an image, defined by

\[
TV_{\text{ani}}(f_c(x,y)) = \sum_{x,y} \left( |D_x(f_c(x,y))| + |D_y(f_c(x,y))| \right),
\]

This equation is a sum of the L1-norm of a first-order forward finite-difference along the horizontal direction \( D_x \) and vertical direction \( D_y \) for all pixel locations \( (x,y) \). In this work, the maximum iteration for solving the TV minimization is set to 20. Lastly, the texture-component image \( f_t(x,y) \) is obtained by

\[
f_t(x,y) = f(x,y) - f_c(x,y).
\]

2) INITIAL BLOCK ENHANCEMENT

Given the initial blocks from process A, we need to enhance these blocks with priority because the following sub-processes rely on the correctness of these initial block enhancements. Firstly, we divide the texture-component image, \( f_t(x,y) \), into 16 x 16 non-overlapped blocks. This partitioning is similar to the first step of process A. Assume that the \( b(m,n) \) is one of the initial blocks provided by process A. With the initial block at the center, we crop a 128 \( \times \) 128 window from the \( f_t(x,y) \) by

\[
f_{w,b(m,n)}(x,y) = f_t(i,j) |(i,j)\in128x128 \text{ with } b(m,n)\text{at center},
\]

where \( f_{w,b(m,n)}(x,y) \) is the cropped window. This window size of 128 \( \times \) 128 is empirically appropriate for fingerprint spectral analysis in the frequency domain. We also apply a Gaussian window, \( g_\sigma(x,y) \), with a standard deviation equal to 16 pixels (\( \sigma = 16 \)). Note that we employ the Gaussian windowing to prevent discontinuity of signal at the window boundary. Then we take the FFT of this window by

\[
F_{b(m,n)}(u,v) = \mathcal{F} \left[f_{w,b(m,n)}(x,y) \ast g_\sigma_{16}(x,y)\right].
\]

Next, we build a matched filter from this 128 \( \times \) 128 spectral patch. Because process A selects the initial block from the high-quality blocks of the input latent fingerprint image, we can exploit the high-quality spectra of this latent fingerprint. Assume that the highest spectral magnitude within the fingerprint bandwidth, which the radius frequencies for a spectral patch of 128 \( \times \) 128 are between 10 to 22 frequency points, is

\[
F_{\text{max}} = \max_{10 \leq \sqrt{u^2+v^2} \leq 22} |F_{b(m,n)}(u,v)|.
\]

Then, we select the absolute spectral magnitude greater than or equal to half of the highest magnitude within the fingerprint bandwidth as a matched filter, as shown in (7).

\[
|S_{b(m,n)}(u,v)| = \begin{cases} |F_{b(m,n)}(u,v)|, & \text{if } |F_{b(m,n)}(u,v)| \geq \frac{F_{\text{max}}}{2} \\ 0, & \text{otherwise} \end{cases}
\]

In the design, we need to smooth the matched filter to increase the filter’s bandwidth by convolving with a Gaussian smoothing filter with a standard deviation of 2.75,
as in [26], [27]. Finally, the magnitude of the matched filter is given by

$$|H_{b(m,n)}(u,v)| = |S_{b(m,n)}(u,v)| * G_{a=2.75}(u,v). \quad (8)$$

Then we multiply the magnitude of the spectral patch by the magnitude of the matched filter, and we raise the multiplication result to the power of 1.25. The spectral boosted magnitude is given by

$$\hat{F}_{b(m,n)}(u,v) = \left( |F_{b(m,n)}(u,v)| |H_{b(m,n)}(u,v)| \right)^{1.25}. \quad (9)$$

Finally, we take a 128 × 128 inverse fast Fourier transform onto the spectral boosted magnitude with the original phase of its corresponding spectral patch, as shown in (10).

$$\hat{f}_{w,b(m,n)}(x,y) = F^{-1}\left\{ \left| \hat{F}_{b(m,n)}(u,v) \right|, \angle F_{b(m,n)}(u,v) \right\}. \quad (10)$$

where $\hat{F}_{b(m,n)}(u,v)$ is the spectral boosted magnitude, $\angle F_{b(m,n)}(u,v)$ is the original phase of the spectral patch, $F^{-1}\{ \}$ is the inverse FFT operator, and $\hat{f}_{w,b(m,n)}(x,y)$ is the enhanced window with the $(m,n)$ at the center. Then we crop the enhanced block $(16 \times 16)$ pixels from the center of the enhanced window $(128 \times 128)$ pixels. The enhanced block is placed on two images at the corresponding location of the original block, $b(m,n)$. One is the enhanced image, $f_e(x,y)$, and another one is the feedback image, $f_f(x,y)$. The enhanced image has the same size as the original latent image with all zero-value pixels for initialization. We insert the enhanced block into the enhanced image by

$$f_e(x,y)\big|_{(x,y)\in b(m,n)} = \hat{f}_{w,b(m,n)}(i,j)\big|_{(i,j)\in 16\times16} = f_f(x,y)\big|_{(x,y)\in b(m,n)} \quad (11).$$

The feedback image is obtained by modifying the texture-component image from the TV decomposition. The intensity of the enhanced block is scaled to the range of $[-1,1]$. This scaled block is inserted into the texture-component image by

$$\hat{f}_f(x,y)\big|_{(x,y)\in b(m,n)} = \hat{f}_{w,b(m,n)}(i,j)\big|_{(i,j)\in 16\times16} \quad (12).$$

The $\hat{f}_f(x,y)$ is the input image for the next sub-process. We called it the feedback image because it contains the initial enhanced block that can improve the fingerprint quality for the neighboring blocks nearby in the next sub-process.

For other initial blocks, we redo the initial block enhancement sub-process using (4) through (12) in parallel. Similarly, the other initial enhanced blocks are placed back in both the enhanced image simultaneously by (11) and the feedback image by (12). Fig. 4 (b) demonstrates this action.

3) ITERATIVE BLOCKS ENHANCEMENT AND FEEDBACK

In this sub-process, we iteratively enhance the feedback image, $\hat{f}_f(x,y)$, block-by-block by starting from the surrounding blocks of the initial blocks from the previous sub-process. Our concept is to enhance the blocks with a genuine fingerprint spectrum first. Once we put these enhanced blocks back into the feedback image, the enhanced fingerprint spectra can improve the weak fingerprint spectrum of the nearby blocks in the next iteration. This sequence is one of the crucial concepts of the proposed algorithm, as shown in Fig. 4 (c)-(g).

The enhancement sequence in this sub-process is different from the previous sub-process with two aspects. First, we use a $64 \times 64$ FFT window instead of the $128 \times 128$ FFT window. The smaller spectral patches are more suitable for enhancing weak latent fingerprints in noisy areas. Second, the matched filter in (8) is applied if the peak of the fingerprint spectra is strong enough. Otherwise, the bandpass filter is used to preserve the original spectra instead. The magnitude of the ideal bandpass filter is defined by

$$|H_{bp}(u,v)| = \begin{cases} 1 & \text{if } 5 \leq \sqrt{u^2+v^2} \leq 13, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Note that the fingerprint spectra range from 5 to 13 frequency-points for the $64 \times 64$ FFT window.

We use the same measure as our preliminary works [26], [27]. The spectral peak ratio (SPR) is defined by

$$SPR = \frac{p_1}{p_1 + p_2} \left| \frac{1}{p_1,p_2\in \max_{5\leq \sqrt{u^2+v^2}\leq 13} |F_{b(m,n)}(u,v)|} \right|, \quad (14)$$

where $p_1$ and $p_2$ are the first and second highest peaks of spectral magnitude in the fingerprint bandwidth. From our experiments, we found that the first peak usually represents the potential fingerprint spectrum, and the second peak represents other spectra such as high sharp edges, noise, or background. For the best case, if $p_2$ is equal to zero, the SPR is equal to one. For the worst case, if $p_1$ is equal to $p_2$, the SPR is equal to 0.5. Hence the range value of $SPR$ is from 0.5 to 1. Based on the $SPR$ value, we divide the strength of the fingerprint signal for each block into three classes: strong, moderate, and weak, as shown in Table 2.

| Spectral classes | Relative signal level | Lower bound of $SPR$ | Threshold level (p) |
|------------------|-----------------------|----------------------|---------------------|
| Strong           | $p_1 \geq 2p_2$       | $SPR = \frac{1}{1 + \frac{1}{2}} = \frac{2}{3}$ | 0.67                |
| Moderate         | $p_1 \geq 1.5p_2$     | $SPR = \frac{1}{1 + \frac{1}{1.5}} = \frac{3}{5}$ | 0.6                 |
| Weak             | $p_1 \geq p_2$        | $SPR = \frac{1}{1 + 1} = \frac{1}{2}$          | 0.5                 |

We divide this sub-process into three tiers based on three threshold levels, as shown in Table 2. We aim to enhance all blocks with an intense fingerprint spectrum in the first tier. The sequence of block enhancement begins with the nearest neighbors of the initial enhanced blocks. We calculate the nearest neighbors using the Euclidean distance transform in [29]. We arrange an enhancement priority of neighboring blocks ascending order from the nearest distance to the
K. Horapong et al.: Progressive and Corrective Feedback for Latent Fingerprint Enhancement

**FIGURE 4.** Demonstration of process B: Progressive feedback latent fingerprint enhancement and examples of the spectral patch improvement in sub-processes. Note that padding enhanced block back can improve the weak fingerprint spectra nearby, for example, the targeted block (18,24). (a). Receiving two initial blocks at (row, col) = (23,28) and (35,16). (b). Boosting the spectral patch at the block (23,28). (c),(d),(e). Progressive boosting the spectral patches from the surrounding blocks (in the cyan area) of the two initial blocks to the targeted block (18,24) with an overlapped window of 64 × 64. (f). Boosting the spectral patch at the targeted block (18,24). (g). The feedback image after finished with η = 0.67.

Further distance [29]. Hence the nearest neighboring blocks, attached to the four sides of the initial enhanced block, are enhanced first using (4) through (10), except that the feedback image, \( \hat{f}(x, y) \), is used instead of the texture-component image, \( f(x, y) \), in (4). Note that we use the 64 × 64 FFT window instead of the 128 × 128 FFT window in (4). We calculate SPR for each block and compare it with a threshold, \( \eta = 0.67 \). If the SPR is greater than or equal to the threshold \( \text{SPR} \geq 0.67 \), we enhance this block using the matched filter (8). Otherwise, we use the ideal bandpass filter in (13) to keep the fingerprint spectra for the next tier. We enhance all four blocks in parallel. Then we insert all four enhanced blocks into the feedback image simultaneously, as in (12). However, only the enhanced blocks that passed the matched filter are placed into the enhanced image (11). Next, the four corner blocks of the initial enhanced block are selected to be enhanced, as shown in Fig. 4. The sequence repeats until all blocks in the given manual fingerprint segment are enhanced. Then the first tier is finished.

For the second tier, we repeat this sub-process for the moderate spectrum blocks using (4) through (12) by changing the threshold \( \eta \) to 0.6. This sub-process is similar to the first tier. Finally, we repeat the sub-process in the third tier by changing the threshold \( \eta = 0.5 \). At the third tier, we output all enhanced blocks to the enhanced image (11). The enhanced image is the final output from process B.

C. PROCESS C: ANOMALOUS FINGERPRINT PATTERN DETECTION

The enhanced fingerprint image from process B may have some defects because strong spectra from noise and background may overcome the weak latent fingerprint spectra. We need to check anomalous fingerprint patterns from the enhanced image and correct these errors. Process C is to detect the locations of anomalous blocks from the enhanced image of process B. These anomalous block positions are then sent back to process B as corrective feedback for our fine-tuning enhancement. In this process, we employ a hierarchical autoencoder to measure the quality of spectral magnitudes of fingerprint patterns. We split this process into four parts; (1) fingerprint spectral autoencoder architectures, (2) training of fingerprint spectral autoencoders, (3) data preparation for spectral autoencoder training, and (4) anomalous fingerprint pattern detection. We give a clear explanation for each part as follows.

1) FINGERPRINT SPECTRAL AUTOENCODER ARCHITECTURES

We design two networks to capture fingerprint spectrum patterns in the frequency domain. The first network, called a **locally spectral autoencoder**, is aimed to learn from good fingerprint spectral shapes of a 64 × 64 spectral patch. The 64 × 64 spectral patch is the magnitude of the FFT.
coefficients. Hence this network can estimate the local fingerprint spectrum shape in a $64 \times 64$ FFT window. Note that the original block size for enhancement is $16 \times 16$ pixels in the spatial domain. The block size is extended to $64 \times 64$ pixels for the FFT window size in the frequency domain. The second network, called a regionally spectral autoencoder, is hierarchical. It combines nine locally spectral autoencoders for inputs and outputs. This network learns from $3 \times 3$ fingerprint spectral patches—the nine neighboring spectral patches contain regional ridge-flow patterns of fingerprints. Hence this network can estimate and predict regional spectrum patterns of fingerprint in the frequency domain. Note that the $3 \times 3$ spectral patches are extracted from the $3 \times 3$ enhanced blocks in the spatial domain. Similarly, each enhanced block size is $16 \times 16$ pixels and extended to $64 \times 64$ pixels for the FFT window size.

The locally spectral autoencoder architecture comprises three fully connected hidden layers, as shown in Fig. 5 (a). The first and the third internal layer contain 1,024 nodes, and the most internal layer has 512 code-size. The inputs of this autoencoder are the spectral magnitudes of the $64 \times 64$ spectral patch. Because the 2-D FFT of a real image is conjugate symmetry, we select only the top-half spectral magnitudes from the 4,096 2-D FFT coefficients as the input vector. We combine the 2,048 coefficients from the top half of a spectral patch with the additional 33 coefficients from the next half horizontal line of the spectral patch. In other words, only spectral magnitudes of 2,081 2-D FFT coefficients out of 4,096 are chosen as the input nodes. This input size reduction decreases the computational complexity and training time of this autoencoder. We rearrange the 2,081 spectral magnitudes into a 1-D vector with a length of 2,081. The first network layer reduces the input size from 2,081 to 1,024. The second layer compresses 1,024 code-size into 512 code-size. The decoder reverses the compression process and reconstructs the 1-D vector of 2,081 spectral magnitudes. Finally, the reconstructed spectral patch can be estimated at the output of this autoencoder.

The regionally spectral autoencoder architecture has only one hidden layer with 1024 code-size, as shown in Fig. 5 (b). The input nodes are from $3 \times 3$ fingerprint spectral patches, encoded to $9 \times 512 = 4,608$ using the nine encoders of locally spectral autoencoders. We rearrange nine of the 512 codes obtained from each encoder as an input vector following the order number shown in Fig. 5 (b). Each input from each encoder is concatenated and rearranged into a 1-D vector with a length of 4,608. The decoder reverses the encoder process to reconstruct $3 \times 3$ fingerprint spectral patches in the same manner.
2) TRAINING OF FINGERPRINT SPECTRAL AUTOENCODERS
The locally spectral encoder-decoder learning uses an input-output pair from two corresponding \(64 \times 64\) spectral patches, as shown in Fig. 5 (a). We obtain a spectral patch for input training from the magnitude of the FFT coefficients of an original high-quality fingerprint image with a size of \(64 \times 64\) pixels. We also obtain a modified spectral patch for output training from a spectral patch modification process. In this process, the corresponding enhanced block, which its location is corresponding to the input patch, is extracted from the enhanced fingerprint image. We apply the VeriFinger10.0 [30] to enhance high-quality fingerprint images. Then we crop the corresponding enhanced fingerprint block (\(64 \times 64\) pixels) from this enhanced image, and we perform the 2-D FFT operation to obtain the enhanced spectral patch. The enhanced spectral patch is converted into a modified spectral patch using three steps. Fig. 6 shows an example of this conversion. Firstly, we eliminate the center frequency magnitude (zero frequency) by placing zeroes at the four center points of the \(64 \times 64\) enhanced spectral patch, as shown in Fig. 6 (b). Second, the foreground spectra are segmented using the Chan-Vese segmentation algorithm [31]. The initial boundary is a square at the boundary of a spectral patch. We set a maximum iteration for 300 iterations. As a result of this active contour segmentation, we obtain several spectral objects, as shown in Fig. 6 (c). Third, we keep only the strong fingerprint spectral object closest to the center (zero frequency). We reject undesired spectra (harmonic spectra) that are far away from the center. To achieve this goal, we calculate a Euclidean distance between the center and each object’s peak. We keep the spectral object with a minimum distance to the center, as shown in Fig. 6 (d). Finally, the modified spectral patch comprises the original spectral magnitude and the obtained spectral object magnitude divided by two. Fig. 6 reveals the step-by-step results of the spectral patch modification process.

![Image](image_url)

**FIGURE 6.** Step-by-step results of the spectral patch modification process for locally spectral autoencoder training.

We use two levels of sparse autoencoder with 1,024 and 512 hidden neurons for the locally spectral autoencoder. We employ the same method as the stacked denoising autoencoder architecture [32], and we train the network based on pairs of spectral patches in supervised learning. The loss function is a mean squared error function with two regularization terms. We implement our training on the MATLAB toolbox [33] by setting the coefficient of \(l_2\) norm regularization and the sparsity regularization to 0.01 and 4, respectively. We choose a scale conjugate gradient-descent algorithm [34] to update weights and bias values during the training. Also, the encoder transfer function is a positive saturating linear function, and the decoder transfer function is linear.

The regionally spectral encoder and decoder learn from encoded vectors extracted from \(3 \times 3\) spectral patches, as shown in Fig. 5 (b). The input vector is a concatenation of the \(512\)-length encoded vectors from nine locally spectral autoencoders. The \(4,608\)-length vector is used as the input and output for training the network. We use another sparse autoencoder to learn essential features in unsupervised learning. In this layer, we select the number of hidden neurons as 1,024. Other training parameters are the same as the locally spectral autoencoder parameters.

3) DATA PREPARATION FOR SPECTRAL AUTOENCODER TRAINING
We extract the high-quality fingerprint blocks from the NIST-SD4 database [35]. This database contains 4,000 rolled fingerprints of 8-bit grayscale images. Each image is \(512 \times 512\) pixels with 500 dpi stored in a modified JPEG lossless format. In this database, most images have a clear friction-ridge structure sufficiently for training our proposed networks. We use the “mindict” function in the NBIS software package [36] to collect high-quality fingerprint blocks. We randomly choose the blocks with the averaged quality value of 2-by-2 local cells [37] more than 3.75, except around the core and delta point since these are rare data. In addition, we extract the core and delta points of NIST-SD4 images using the VeriFinger 10.0 extractor [30]. Fig. 7 (a) shows example pairs of \(3 \times 3\) spectral patches from corresponding original/enhanced pairs. Fig. 7 (b) demonstrates example pairs of an individual spectral patch from fingerprint/enhancement pairs available in our training dataset.

Our work creates 51,350 pairs of \(3 \times 3\) spectral patches and 241,107 pairs of individual spectral patches for network training. In particular, the training spectral patches created for the friction ridge area are taken from several directions distributed around the core point. We collect these spectral patches from eight sectors around the core point. More particularly, we perform the MCAR [38] setting in the regionally spectral autoencoder network dataset during training. We aim to improve the ability of the regionally spectral autoencoder network to deal effectively with the missing spectral patches. The number of missing spectral patches is randomly selected from 1 up to 4 patches during the training. Finally, we split the entire dataset into three different sets: training, validation, and testing in the ratio of 70:15:15. Fig. 8 visualizes a set of bases learned by our network model and reveals how the basis attempts to capture the fingerprint spectral peaks in various patterns. The advantage of the proposed autoencoder is that these spectral patches are sparse in the frequency domain, and they can represent patterns of friction ridges in the spatial domain.
4) ANOMALOUS FINGERPRINT PATTERN DETECTION

We apply the regionally spectral autoencoder for anomalous fingerprint pattern detection. The enhanced image from process B is the input of this process C. We analyze the enhanced image block-by-block of $16 \times 16$ pixels inside the segmented area. Each targeted block and its eight neighboring blocks form a group of $3 \times 3$ enhanced blocks, as shown in Fig. 9. With each block at the center, the block size is extended to a window size of $64 \times 64$ pixels to cover the nearby enhanced area. Each spectral patch can be obtained by calculating the magnitudes of the FFT coefficients of each window. Finally, we obtain $3 \times 3$ overlapped spectral patches from each targeted block, as shown in Fig. 9. These $3 \times 3$ spectral patches are an input of the regionally spectral autoencoder.

As shown in Fig. 9, we generate nine groups of $3 \times 3$ spectral patches for the targeted block and its eight neighboring blocks. The spectral patch of the targeted block is located at a different position in each group, resulting in a different prediction result. Nine groups of $3 \times 3$ spectral patches are fed into the regionally spectral autoencoder. Here we obtain nine output spectral patches of the same targeted block from nine output groups. Then we average these nine output spectral patches of the same targeted block to predict the spectral patch of the enhanced block, as shown in Fig. 9.

We use the Jensen–Shannon divergence [39] to measure the difference between the enhanced spectral patch and the predicted spectral patch of the same targeted block. The Jensen–Shannon divergence is a symmetrized version of the Kullback-Leibler divergence, defined by

$$
D_{J(m,n)}(S_E \parallel S_P) = \frac{1}{2}D_{K(m,n)}(S_E \parallel S_M) + \frac{1}{2}D_{K(m,n)}(S_P \parallel S_M),
$$

(15)

where $S_E$ and $S_P$ are the enhanced spectral patch and the predicted spectral patch from the same targeted block $b(m,n)$, respectively. $S_M = 0.5(S_E + S_P)$ is an arithmetic mean of the two spectral patches. $D_{K(m,n)}(S_E \parallel S_P)$ is the Kullback-Leibler divergence between two spectral patches $S_E$ and $S_P$ of the targeted block $b(m,n)$, given by

$$
D_{K(m,n)}(S_E \parallel S_P) = \sum_{i,j} S_{E(m,n)}(i,j) \ln \left( \frac{S_{E(m,n)}(i,j)}{S_{P(m,n)}(i,j)} \right)
$$

(16)

for all frequency-point $(i,j)$ in both spectral patches.

We convert the divergence between enhanced and predicted spectral patches into a similarity score, defined by

$$
SS_{FSP}(m,n) = \frac{\min_{b(m,n) \in BOI_1} D_{J(m,n)}}{D_{J(m,n)}},
$$

(17)

where $SS_{FSP}(m,n)$ is the fingerprint spectral pattern similarity score at the targeted block position $(m,n)$ in the manual fingerprint segment (BOI_1). We calculate the minimum divergence for all blocks inside the manual fingerprint segment.
Hence the maximum similarity score is one. Note that the higher the similarity score, the lower the divergence between enhanced and predicted spectral patches.

With this similarity score, we can distinguish between normally and abnormally enhanced blocks. The enhanced blocks, in which their similarity score is less than a threshold level, are classified as abnormally enhanced blocks. Otherwise, they are classified as normally enhanced blocks. We set the variable threshold depending on the number of refinement iterations \((k)\). The threshold level is gradually decreased \((\eta FSP)\), for each ongoing refinement iteration, given by

\[
\eta FSP(k) = \mu SS + \left(\frac{3 - k}{2}\right) \sigma SS
\]

where \(\eta FSP(k)\) is a variable threshold level at the \(k\) refinement iteration, and \(k\) is the number of refinement iterations varying in the range of \(1 \leq k \leq 5\). A mean value and a standard deviation value of all similarity scores in the fingerprint segment are denoted by \(\mu SS\) and \(\sigma SS\), respectively.

After each refinement iteration is complete, the positions of abnormally enhanced blocks are sent back to process B instead of initial block locations from process A. We assign all block positions of normally enhanced blocks as the new initial blocks, with no more enhancement for these blocks. An input image for the next refinement iteration is composed of enhanced blocks at the locations of normally enhanced blocks and the texture-component image at the locations of abnormally enhanced blocks. Then we repeat process B and process C until \(k\) is reached the assigned number of the refinement iteration. The final output is the enhanced image.

**IV. EXPERIMENTAL RESULTS**

We perform benchmark tests with two automatic fingerprint identification systems (AFIS), one commercial-off-the-shelf (COTS), and one open-source. We employ the VeriFinger 10.0 SDK [30] for COTS, which is the latest unchanged version; on revision: 186632. For an open-source fingerprint matching system, we use a combination of the MINDTCT minutiae extractor from NBIS SDK 5.0.0 [36] and the Minutiae Cylinder-Code (MCC) matcher [40] from MCC SDK 2.0 [41]. In our experiments, we use MCC16 floating-point version in the MCC SDK 2.0. Two different automatic fingerprint identification systems (AFIS) give us two different fingerprint matching approaches. The COTS uses all features available in an input enhanced image, but the open-source uses only minutiae for matching.

We experiment on two public latent fingerprint databases; NIST-SD27 [42] and IITD-MOLF DB4 [43]. Even though the NIST-SD27 dataset has been withdrawn from NIST, it is a fundamental choice for benchmark due to rich published enhancement results. Latent fingerprint images in this dataset were collected from real solved cases. This dataset contains latent fingerprints with different qualities on various complex backgrounds. On the other hand, the IITD-MOLF DB4 dataset is currently available with limited published enhancement results. Latent fingerprints in this dataset are usually on a clear background. With these two databases, we can evaluate the performance of latent fingerprint enhancement algorithms in two different environments.

**A. BENCHMARK TESTS ON NIST-SD27 DATABASE**

The NIST-SD27 database contains 258 latent fingerprint images from actual crime scenes and corresponding ten-print image pairs. Latent fingerprints in this database are classified into three classes of quality: good, bad, and ugly. There are 88 images for good quality, 85 images for bad quality, and 85 images for ugly quality. For a real-world scenario, we insert the NIST-SD14 [44] for an additional background database to extend the fingerprint gallery in our latent fingerprint identification experiments. We use only 27,000 images from file cards (with prefix f-), which are in the first subset of NIST-SD14. As a result, a total of 27,258 fingerprints are available for identification testing on the NIST-SD27 database.

Currently, we can compare with only seven published algorithms due to available latent fingerprint enhancement results from this NIST-SD27 database. We obtained the enhanced results from authors of GlobalDict [6], LocalDict [7], RidgeDict [8], SpectralDict [13], DN-Unets [18], and Semi-Prgs [26]. In addition, we used the released code available in [45], the FingerNet algorithm [14], to enhance latent fingerprint images. The proposed algorithm (called progressive and corrective feedback (PCF)) and the other algorithms in [6], [7], [13], [26] employ the same manual fingerprint segmentation, released by [6]. On the other hand, the algorithms [8], [14], [18] can automatically segment latent fingerprints from various backgrounds.

1) **BENCHMARK TEST WITH COTS VERIFINGER 10.0**

We use a Cumulative Matching Characteristic (CMC) curve, which reports identification rate vs. ranking, as a performance metric for our benchmark tests. We start the performance testing by probing the enhanced images obtained from each enhancement algorithm to the COTS VeriFinger 10.0 system. The CMC curve can be calculated from the ranking output of this COTS matcher. Fig. 10 shows the CMC curves from the COTS VeriFinger10.0. We provide precise identification rates for rank-1, -5, -10, -20, and -30 in Table 3.

The proposed algorithm outperforms most state-of-the-art algorithms except in the good-quality case. It achieves the best accuracy for rank-1 and rank-30. However, it is inferior to our preliminary work, the Semi-Prgs algorithm [26] for rank-5, rank-10, and rank-20. The reason is that this algorithm needs human assistance to choose the initial block location, but the proposed algorithm performs automatic initial block localization. Hence starting at the right location is very crucial for the proposed algorithm to achieve high accuracy.
FIGURE 10. Comparison of CMC curves of eight latent fingerprint enhancement algorithms using COTS VeriFinger10.0 for both minutiae extractor and matcher. All 258 latent fingerprints from the NIST-SD27 database are probed. The database background comprises the corresponding 258 rolled fingerprints from the NIST-SD27 database and the 27,000 fingerprints from the NIST-SD14 database.

TABLE 3. Identification rates using verifinger10.0 for both minutiae extractor and matcher on NIST-SD27 database (overall).

| Algorithms     | Rank-1 | Rank-5 | Rank-10 | Rank-20 | Rank-30 |
|----------------|--------|--------|---------|---------|---------|
| GlobalDict [6] | 48.84  | 53.88  | 57.36   | 58.53   | 58.53   |
| LocalDict [7]  | 47.29  | 54.26  | 55.81   | 57.36   | 58.53   |
| SpectralDict [13] | 46.12  | 53.10  | 54.26   | 56.59   | 57.75   |
| LocalDict [7]  | 47.29  | 54.26  | 55.81   | 57.36   | 58.53   |
| GlobalDict [6] | 48.84  | 53.88  | 57.36   | 58.53   | 58.53   |
| FingerNet [14] | 51.55  | 58.14  | 60.47   | 62.02   | 62.40   |
| Semi-Prgs [26] | 54.65  | 63.18  | 66.28   | 68.22   | 68.60   |
| Proposed PCF  | **56.98** | **62.02** | **64.73** | **66.67** | **68.60** |

2) BENCHMARK TEST WITH MCC 2.0
In this experiment, we focus on minutiae detection and matching using open-source algorithms. The enhanced latent fingerprints are sent to the MINDTCT minutiae extractor from NBIS SDK 5.0.0 [36] to create a minutiae template. Then, the template matching is performed by the MCC SDK 2.0 [41]. Fig. 11 shows the CMC curves from this open-source matching system. We also report the identification rate for rank-1 to rank-30 in Table 4. Our proposed method still outperforms most algorithms for the overall case. However, if the rank is greater than 20, the DN-Unets [18] achieves the best accuracy.

Experiencing with two AFISs, we found that different AFIS provides different identification results. Changing the AFIS alters the ranking of identification performance of latent enhancement algorithms. For the good-quality case, our proposed method is inferior to a few published algorithms using COTS, but, by contrast, it is the best using minutiae-based matching. The contradiction happens for the bad-quality case. However, the proposed method provides robust latent enhanced results to be used with different AFISs.

B. BENCHMARK TESTS ON IITD-MOLF DATASET
The IITD-MOLF DB4 database [43] contains 4,400 latent fingerprint images from all ten fingers of 100 persons. Each
finger can have at least one image, up to five images depending on the recording session. We treat each latent fingerprint image as an individual query fingerprint. Hence, we probe 4,400 unique queries for the test. However, we refer to the IITD-MOLF DB3_A [42] database for their corresponding ten-print image pairs. It has 4,000 live-scan slap fingerprint images captured by the CrossMatch L-Scan Patrol sensor for the same 1,000 fingers. Combined with 27,000 fingerprints from NIST-SD14, we have 31,000 fingerprints of a fingerprint gallery for identification tests.

In this benchmark test, we obtained only enhanced results from three published algorithms; SpectralDict [13], FingerNet [14], and 2-Stage-Prgs [27]. Unfortunately, the state-of-the-art authors [18] could not provide us with the enhanced results in this database. Therefore we cannot include results from [18] in this evaluation. Because [13] and [27] are our previous works, we can reproduce enhanced results from our source codes. For [14], we use the released code [45] to enhance latent fingerprint images. Note that the enhanced results from [13], [27], and the proposed algorithm used manual segments from [27], while the enhanced results from [14] used their auto-segments. We did not have enhanced results from [26] in this dataset because it needs manually selected initial block locations of 4,400 images.
K. Horapong et al.: Progressive and Corrective Feedback for Latent Fingerprint Enhancement

FIGURE 12. Comparison of CMC curves of four latent fingerprint enhancement algorithms using COTS and the open-source matching system. All 4,400 latent fingerprints from the IITD-MOLF DB4 database are probed. The database background comprises the corresponding 4,000 plain fingerprints from the IITD-MOLF DB3A database and the 27,000 fingerprints from the NIST-SD14 database.

TABLE 5. Identification rates using verifinger 10.0 for both minutiae extractor and matcher on IITD-MOLF DB4 database.

| Algorithms       | Ranking from VeriFinger 10.0 |
|------------------|-------------------------------|
|                  | Rank-1 | Rank-5 | Rank-10 | Rank-20 | Rank-30 |
| FingerNet [14]   | 30.64  | 33.52  | 35.05   | 36.80   | 37.98   |
| SpectralDict [13]| 33.70  | 37.59  | 39.23   | 41.41   | 42.80   |
| 2-StagePrgs [27] | 35.18  | 39.16  | 40.73   | **42.77** | **43.84** |
| Proposed PCF     | **35.48** | **39.20** | **40.86** | 42.66   | 43.68   |

1) BENCHMARK TEST WITH COTS VERIFINGER 10.0
Using COTS VeriFinger 10.0, we plot the CMC curves for the IITD-MOLF DB4 database in Fig. 12 (a). The identification rates versus ranking are reported in Table 5. Our proposed method and our preliminary work [27] outperform the deep learning FingerNet [14] approximately 5% identification accuracy, a significant gap for improvement.

2) BENCHMARK TEST WITH MCC 2.0
Using the MINDTCT minutiae extraction [36] and the MCC minutiae matcher [41], we obtain the CMC curves for the IITD-MOLF DB4 database in Fig. 12 (b). The identification rates versus ranking are shown in Table 6.

The identification results are comparable. The FingerNet slightly outperforms our proposed method for rank-1 and rank-5. Nevertheless, our proposed method gains better performance for the rank-10, rank-20, and rank-30.

C. ABSTRACTION ON THE REFINEMENT ITERATION
We perform an ablation study of the refinement iteration in process C, anomalous fingerprint pattern detection. In this process, we set a maximum number \( k \) of the refinement iterations to five. Without feedback or no iteration \( (k = 0) \), we activate only process A and process B without process C. This case is called “feedforward.” Fig. 13 demonstrates the CMC curves by varying feedbacks or refinement iterations; \( k = 0, 1, 2, \ldots, 5 \). Similar to the previous experiments, we operate the same benchmark tests with two AFISs and two latent fingerprint databases. The experiment results show that the refinement iteration of process C always improves the performance of the proposed algorithm. However, increasing the number of refinement iterations does not guarantee better performance. For example, as shown in Fig 13 (a), using COTS VeriFinger with the NIST-SD27 database, the best performance is \( k = 2 \). The performance result of \( k = 5 \) is inferior to \( k = 2 \) or 3. For other cases, as shown in Fig 13 (b), (c), (d), the proposed method tends to gain better performance while \( k \) is increasing. Hence, we set \( k = 5 \) as our best performance results for benchmarking against the other algorithms, as shown in Fig 10, 11, and 12. Note that our preliminary work, the Semi-Prgs method [26], is similar to process B without process A. The difference is that we need to choose the initial block location for starting process B manually in [26].

VOLUME 9, 2021

FIGURE 13. CMC curves using COTS VeriFinger 10.0 on latent fingerprints from IITD-MOLF DB4 database.
We demonstrate why increasing refinement iterations may result in better or worse enhancement results in Fig. 14. Fig. 14 (a) shows some successful cases. The proposed algorithm can remedy some mistakes from the feedforward or previous feedback processes. However, the later iterations cannot further improve enhanced results. On the other hand, Fig. 14 (b) demonstrates unsuccessful cases. The proposed algorithm is confused by overlapped latent fingerprints in the B146 image from NIST-SD27. Another failure is caused by the large missing area of friction ridges in the 23_L_6_1 image from IITD-MOLF. The enhancement process goes wrong and cannot recover from corrective feedback. These examples can tell us why higher iteration may not yield the best result.

D. VISUAL INSPECTION AND COMPARISON
To understand the strength and weaknesses of the proposed algorithm, we visually inspect and compare enhanced results. We select enhanced results from the top-5 algorithms in Table 3 and the top-4 algorithms in Table 5 for visual comparisons. Fig. 15 shows two latent fingerprint examples from the NIST-SD27 database. The B180 image in Fig. 15 (a) contains a bad-quality latent fingerprint deposited on a knife blade. The Verifinger 10.0 can identify three enhanced results at the first rank and the other two within the third rank. The MCC 2.0 provides ranking differentiation. Our proposed algorithm achieves the best MCC ranking for this image. Most algorithms suffer from combining the latent fingerprint with a strong edge from the blade. Both FingerNet [14] and DN-UNets [18] reconstruct some fake ridges due to their over-segmentation around the actual latent fingerprint boundary. Our proposed result can enhance unclear ridges in the upper left zone, which are a failure from our preliminary work, Semi-Prgs [26].

The U270 image in Fig. 15 (b) is an ugly quality from the NIST-SD27 database. This image contains unclear ridges around a core point. Rankings from the VerifiFinger 10.0 are the same for all algorithms, but rankings from the MCC 2.0 are different. The proposed algorithm preserves ridges around the core point area. Most algorithms could not correctly reconstruct ridges around singular point areas.

Fig. 16 shows two other latent fingerprint examples from the IIT-D MOLF DB4 database. The 33_R_4_2 image in Fig. 16 (a) has a large area of unclear ridges in the middle area of a whorl-type fingerprint. On the other hand, there are clear ridges in the upper and lower areas. The proposed
algorithm can reconstruct ridges in the central area while the other algorithms fail. The reason is that the proposed algorithm can diffuse high-quality ridge spectrum into the low-quality area, resulting in better-enhanced results.

Another showcase in the IIT-D MOLF DB4 database, the 24_R_8_4 image, is a right loop type in Fig. 16 (b). This image contains a weak fingerprint pattern for the entire segment. Moreover, there are very unclear ridges in the bottom zone. The proposed algorithm yields the rank-1 identification result for both AFIS systems. Our 2-StagePrgs algorithm [27] has a problem around a core point, resulting in a slightly shifted location of the detected core point. The FingerNet enhanced result [14] excludes the bottom zone from its segmentation and fails to produce its identification rank for the MINDTCT and MCC AFIS system. Note that the full enhanced results from both databases are available upon request.

**E. EXECUTION TIME**

We implement our works using the Matlab2018a toolbox and Microsoft Visual C# 2017, which runs on an Intel Core i7 CPU 2.2GHz with 8GB RAM and NVIDIA GeForce GTX 1060 6GB GPU. Table 7 reports the execution times for each process of the proposed method for two benchmark databases. The NIST-SD27 image size is $800 \times 768$ pixels and the IITD-MOLF DB4 image size is $320 \times 448$ pixels. Note that the later iteration is faster than the former iteration because most blocks have already been enhanced. Therefore, only remaining anomalous blocks are in this process for re-enhancement.
We train two separate components of the sparse autoencoder for the locally spectral autoencoder. The training time is 36.7 hours. In addition, it consumes 36.8 hours for retraining the full stacking. The regionally spectral autoencoder requires 16.2 hours for training and 12.6 hours for retraining the missing data.

Our proposed algorithm’s execution time may be slow compared to those deep learning approaches, which require less than a second [18] or only a few seconds [14] for enhancement. However, this is not a critical issue. Instead, the more critical issue is how the algorithm can correctly enhance most hidden features in the input latent fingerprint image. The goal of the latent fingerprint enhancement is to increase the AFIS hit rate and identify the prime suspect.

V. CONCLUSION AND FUTURE RESEARCH

We combine two powerful mechanisms to solve the latent fingerprint enhancement problem. The first mechanism uses boosted spectral filtering to improve high-quality friction ridges in priority. Then the enhanced friction ridges are inserted back as feedback to improve low-quality ridges nearby. This mechanism provides progressive feedback that can exploit intra-image correlation. The second mechanism...
employs machine learning for corrective feedback. This mechanism uses locally and regionally spectral autoencoders for anomalous fingerprint pattern detection. Note that this mechanism explores inter-image correlation. The combination of the two mechanisms gives us a novel framework for latent fingerprint enhancement.

There are some drawbacks to the proposed framework. First, the proposed framework is quite complicated. Second, the proposed autoencoder cannot detect incorrectly enhanced results of global fingerprint patterns, as shown in Fig. 14 (b). Third, the manual segmentation of latent fingerprints is necessary for the input of the proposed framework.

Most deep learning approaches provide automatic latent fingerprint segmentation and enhancement in the same package. In contrast, the proposed algorithm requires manual latent fingerprint segmentation. Nevertheless, we think that human-guided segmentation is still required in the practical forensic routine if multiple latent prints are in one image. Moreover, latent fingerprint examiners need to focus on the targeted fingerprint or the best-quality latent fingerprint in the given image. The fully automatic system requires several automatic processes, including segmentation, quality assessment, enhancement, and targeted fingerprint selection. Therefore, we need a solution that has not been thoroughly answered yet.

Unfortunately, we cannot compare our enhanced results with the GAN approaches due to a lack of those results. We have already requested GAN’s enhanced results from some authors in the literature, but we currently receive no results. Therefore, the question to be answered is that whether the GAN’s results can surpass our results. In our opinion, the progressive feedback of our proposed framework is responsible for the most improvement of latent fingerprint enhancement. On the other hand, the exiting GAN approaches cannot exploit and boost weak friction ridges from the enhanced ridges (no intra-feedback mechanism). So we left this task for future research. We shall provide our enhanced results from both public databases for future comparison upon request to answer this question.

For future research, there is plenty of room for improvement. The spectral autoencoder with fully connected layers is simple, but it may not be efficient. We shall extend our learning model by exploring deep CNN architectures in the frequency domain for globally anomalous fingerprint pattern detection. Combining progressive feedback with a deep learning model such as CNN will bring us new hope for a better latent fingerprint enhancement scheme.

ACKNOWLEDGMENT
The contribution of Kittinuth Srisutheenon is based on his past thesis work at Kasetsart University.

REFERENCES
[1] K. Wertheim, “Embryology and morphology of friction ridge skin,” in Fingerprint Sourcebook, A. McRoberts, Ed. Washington, DC, USA: National Institute of Justice, 2011, ch. 3, pp. 1–26.
[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, “Fingerprint analysis and representation,” in Handbook of Fingerprint Recognition, 2nd ed. New York, NY, USA: Springer, 2009, pp. 96–166.
[3] L. Hong, Y. Wan, and A. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 777–789, Aug. 1998.
[4] S. Karimi-Ashtiani and C. C. J. Kuo, “A robust technique for latent fingerprint image segmentation and enhancement,” in Proc. Int. Conf. Image Process. (ICIP), Oct. 2008, pp. 1492–1495.
[5] S. Yoon, J. Feng, and A. K. Jain, “Latent fingerprint enhancement via robust orientation field estimation,” in Proc. Int. Joint Conf. Biometrics (IJCB), Oct. 2011, pp. 1–8.
[6] 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.
[7] 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.
[8] K. Cao, E. Liu, and A. K. Jain, “Segmentation and enhancement of latent fingerprints: A coarse to fine ridge structure dictionary,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 9, pp. 1847–1859, Sep. 2014.
[9] M. Liu, X. Chen, and X. Wang, “Latent fingerprint enhancement Via multi-scale patch based sparse representation,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 1, pp. 6–15, Jan. 2015.
[10] 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.
[11] S. Liu, M. Liu, and Z. Yang, “Sparse coding based orientation estimation for latent fingerprints,” Pattern Recognit., vol. 67, pp. 164–176, Jul. 2017.
[12] M. Xu, J. Feng, J. Lu, and J. Zhou, “Latent fingerprint enhancement using Gabor and minutia dictionaries,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2017, pp. 3540–3544.
[13] W. Chaidee, K. Horapong, and V. Areekul, “Filter design based on spectral dictionary for latent fingerprint pre-enhancement,” in Proc. Int. Conf. Biometrics (ICB), Gold Coast, QLD, Australia, Feb. 2018, pp. 23–30.
[14] Y. Tang, F. Gao, J. Feng, and Y. Liu, “FingerNet: An unified deep network for fingerprint minutiæ extraction,” in Proc. IEEE Int. Joint Conf. Biometrics (IJCB), Denver, CO, USA, Oct. 2017, pp. 108–116.
[15] J. Svoloba, F. Monti, and M. M. Bronstein, “Generative convolutional networks for latent fingerprint reconstruction,” in Proc. IEEE Int. Joint Conf. Biometrics (IJCB), Denver, CO, USA, Oct. 2017, pp. 429–436.
[16] J. Li, J. Feng, and C.-C.-J. Kuo, “Deep convolutional neural network for latent fingerprint enhancement,” Signal Process., Image Commun., vol. 60, pp. 52–63, Feb. 2018.
[17] P. Qian, A. Li, and M. Liu, “Latent fingerprint enhancement based on DenseUNet,” in Proc. Int. Conf. Biometrics (ICB), Crete, Greece, Jun. 2019, pp. 1–6.
[18] M. Liu and P. Qian, “Automatic segmentation and enhancement of latent fingerprints using deep nested UNets,” IEEE Trans. Inf. Forensics Security, vol. 16, pp. 1709–1719, 2021.
[19] A. Dabouei, S. Soleymani, H. Kazemi, S. M. Iranmanesh, J. Dawson, and N. M. Nasrabadi, “ID preserving generative adversarial network for partial fingerprints: A coarse to fine ridge structure dictionary,” in Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Oct. 2018, pp. 1–10.
[20] Y. Liu, Y. Tang, R. Li, and J. Feng, “Cooperative orientation generative adversarial network for latent fingerprint enhancement,” in Proc. Int. Conf. Biometrics (ICB), Crete, Greece, Jun. 2019, pp. 1–10.
[21] Y. Xu, Y. Wang, J. Liang, and Y. Jiang, “Augmentation data synthesis Via gans: Boosting latent fingerprint reconstruction,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Barcelona, Spain, May 2020, pp. 2932–2936.
[22] X. Huang, P. Qian, and M. Liu, “Latent fingerprint image enhancement based on progressive generative adversarial network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Seattle, WA, USA, Jun. 2020, pp. 3481–3489.
[23] P. Sutthiwichaiporn, V. Areekul, and S. Jirachaweng, “Iterative fingerprint enhancement with matched filtering and quality diffusion in spatial-frequency domain,” in Proc. 20th Int. Conf. Pattern Recognit., Istanbul, Turkey, Aug. 2010, pp. 1257–1260.
[24] P. Sutthiwichaiporn and V. Areekul, “Adaptive boosted spectral filtering for progressive fingerprint enhancement,” Pattern Recognit., vol. 46, no. 9, pp. 2465–2486, Sep. 2013.
K. Horapong, K. Srisutheenon, and V. Areekul, “Progressive and Corrective Feedback for Latent Fingerprint Enhancement,” in Proc. Int. Conf. Imag. Crime Detection Prevention, London, U.K., 2019, pp. 62–67.

K. Horapong, K. Srisutheenon, and V. Areekul, “Progressive latent fingerprint enhancement using two-stage spectrum boosting with matched filter and sparse autoencoder,” in Proc. 17th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON), Phuket, Thailand, Jun. 2020, pp. 531–534.

S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, “An augmented Lagrangian method for Total variation video restoration,” IEEE Trans. Image Process., vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

C. R. Maurer, R. Qi, and V. Raghavan, “A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 2, pp. 265–270, Feb. 2003.

VeriFinger Version 10.0 SDK. (2018). Neurotechnology. Accessed: Jul. 3, 2021. [Online]. Available: https://www.neurotechnology.com/verifinger.html

T. F. Chan and L. A. Vese, “Active contours without edges,” IEEE Trans. Image Process., vol. 10, no. 2, pp. 266–277, Feb. 2001.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, and L. Bottou, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” J. Mach. Learn. Res., vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

M. H. Beale, M. T. Hagan, and H. B. Demuth. (2015). Neural Network Toolbox User’s Guide. Accessed: Jul. 3, 2021. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.699.4831&rep=rep1&type=pdf

M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” Neural Netw., vol. 6, no. 4, pp. 525–533, Jan. 1993.

NIST Special Database 4: Fingerprint Database, U.S. Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 1992. [Online]. Available: https://www.nist.gov/srd/nist-special-database-4

NIST Biometric Image Software (NBIS) 5.0.0 SDK. [Online]. Available: https://www.nist.gov/itl/iad/image-group/products-and-services/image-group-open-source-server-nbigs

C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, and K. Ko. User’s Guide to NIST Biometric Image Software (NBIS). Accessed: Jul. 3, 2021. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51097

D. B. Rubin, “Inference and missing data,” Biometrika, vol. 63, no. 3, pp. 581–592, 1976, doi: 10.1093/biomet/63.3.581.

J. Lin, “Divergence measures based on the Shannon entropy,” IEEE Trans. Inf. Theory, vol. 37, no. 1, pp. 145–151, Jan. 1991.

R. Cappelli, M. Ferrara, and D. Maltoni, “Minutia cylinder-code: A new representation and matching technique for fingerprint recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 12, pp. 2128–2141, Dec. 2010.

R. Cappelli, M. Ferrara, and D. Maltoni. (2014). Minutia Cylinder-Code SDK. [Online]. Available: http://biolab.csr.unibo.it/

NIST Special Database 27/27A: Fingerprint Database, U.S. Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2001. [Online]. Available: https://www.nist.gov/itl/iad/image-group/nist-special-database-2727a

A. Sankaran, M. Vatsa, and R. Singh, “Multisensor optical and latent fingerprint database,” IEEE Access, vol. 3, pp. 653–665, Apr. 2015.

KITTINUTH SRISUTHEENON received the B.Eng. and M.Eng. degrees in electrical engineering from Kasetsart University, Bangkok, Thailand, in 2001 and 2005, respectively, where he is currently pursuing the D.Eng. degree in electrical engineering. His research interests include biometric authentication, signal and image processing, machine learning, and pattern recognition.

KITTIPOL HORAPONG received the B.Eng. and M.Eng. degrees in electrical engineering from Kasetsart University, Bangkok, Thailand, in 2017 and 2020, respectively. He then joined Chan Wanich Company Ltd. His research interests include fingerprint recognition, image processing, machine learning, and artificial intelligence.

VUTIPONG AREEKUL (Member, IEEE) received the B.Eng. degree in electrical engineering from the King Mongkut’s Institute of Technology, Bangkok, Thailand, in 1986, the M.Eng. degree in electrical engineering from Kasetsart University, Bangkok, Thailand, in 1990, and the Ph.D. degree in electrical and computer engineering from Washington State University, USA, in 1998. He then joined the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, where he is currently a Professor. He was the General Co-Chair of the International Conference on Biometrics, in Phuket, Thailand, in 2015, and in Gold Coast, Australia, in 2018. In addition, he was the principal investigator of several Thailand e-Government biometric interoperability and standardization projects. These projects were supported by the Ministry of Information and Communication Technology (MICT), the Ministry of Justice (MOJ), and the National Security Council (NSC), Thailand. His research interests include fingerprint recognition, biometrics, pattern recognition, and image processing. He was a recipient of the Siew-Sngiem Karnchanachari Research Leadership Awards 2012.