IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 9, 2019, accepted May 20, 2019, date of publication May 23, 2019, date of current version June 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918503

Finger-Vein Recognition Based on Deep DenseNet Using Composite Image

JONG MIN SONG, WAN KIM, AND KANG RYOUNG PARK
Division of Electronics and Electrical Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Kang Ryoung Park (parkgr@dongguk.edu)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1D1A1B07041921, in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2017R1D1A1B03028417, and in part by the Bio and Medical Technology Development Program of the NRF funded by the Korean Government, MSIT, under Grant NRF-2016M3A9E1915855.

ABSTRACT Finger-vein recognition has the advantages of high immutability, as finger veins are located under the skin, high user convenience, as a non-invasive and contactless capture device, is used, and high readability even when one of the fingers is damaged or not available for recognition. However, there is an issue of recognition performance degradation caused by finger positional variation, misalignment, and shading from uneven illumination. The existing hand-crafted feature-based methods have exhibited varied performance depending on how these issues were handled by pre-processing. To overcome this shortcoming of hand-crafted feature-based methods, convolutional neural network (CNN)-based recognition methods have been researched. The existing systems based on a CNN use two methods: using a difference image as the input to the network and calculating the distance between feature vectors extracted from the CNN. Difference images can be susceptible to noise as they are generated by differences in pixel values. Also, the method for calculating the distance between feature vectors cannot employ all layers of the trained network and has less accuracy than the method employing difference images. To address these issues, this paper examined a method less susceptible to noise and which uses the entire network; a composite image of two finger-vein images was used as the input to a deep, densely-connected convolutional network (DenseNet). Two open databases, namely Shandong University homologous multi-modal traits (SDUMLA-HMT) finger-vein database and The Hong Kong Polytechnic University finger image database (version 1), were used for experiments and the results show that the proposed method has greater performance than the existing methods.

INDEX TERMS Finger-vein recognition, composite image, deep DenseNet.

I. INTRODUCTION

Existing biometric systems involve various technologies such as face, iris, fingerprint, and finger-vein recognition. Among these, finger-vein recognition has the following advantages [1]: (1) as veins are hidden inside the body and are mostly invisible to the human eye, finger-vein identification makes it hard to forge or steal identification; (2) the non-invasive and contactless capture ensures both convenience and cleanliness, making the system more acceptable for the user; (3) with ten fingers, if something unexpected happens in one finger, other fingers can also be authenticated. However, upon image capture, finger-vein identification can suffer from performance degradation caused by illumination variations, finger positional variation, shading, and misalignment. To address these issues, image pre-processing is performed including region of interest (ROI) segmentation, rotational transformation, and image enhancement. The existing hand-crafted feature-based methods such as local binary pattern (LBP) and local directional pattern (LDP) suffer recognition accuracy degradation unless they adequately adjust image noise and misalignment during pre-processing or apply an optimal filter for the enhancement of finger-vein lines. These methods also have the disadvantage that they need to obtain an optimal filter value based on the characteristics of experimental database. Due to these reasons, there have been studies on finger-vein recognition using convolutional neural networks (CNNs), which are less affected by pre-processing.

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

The existing finger-vein recognition systems based on a CNN use a difference image as an input to the network or calculate the distance between feature vectors [2], [3]. As for a difference image, since it is generated by calculating the difference in each pixel value between enrolled images and input images, a single input can yield the genuine or imposter output, and it uses all layers of the trained network in the recognition process. However, it can be susceptible to noise as there can be low changes in the feature values of difference images. In addition, the method for calculating the distance between feature vectors uses an original image as an input image. Therefore, to perform recognition, feature vectors are extracted from a layer prior to the final layer of the network for each image to calculate the distance between feature vectors. So, not all layers of the trained network are used, and it also has less accuracy than a system that uses a difference image as an input image. To resolve these issues, this paper studied the method of using a composite of two images as an input image to a CNN in order to be less susceptible to noise and to use the entire trained network; a deep CNN model of a densely-connected convolutional network (DenseNet) was used. In addition, shift matching was introduced to address performance degradation due to the misalignment between enrolled and input images. Related works are explained in Section II. Section III describes the contributions of this study. Section IV explains in detail the proposed deep DenseNet-based finger-vein recognition method. Sections V and VI each contain study results, analysis, and conclusions.

II. RELATED WORKS

Finger-vein recognition involves image acquisition, pre-processing, feature extraction, and matching. In pre-processing, ROI segmentation, image resizing, image alignment, and image enhancement are performed. There have been studies on the application of a Gabor filter to identify vein patterns in the image enhancement process [4]–[12]. Yang *et al.* proposed a method using 16 types of Gabor filters which take into account 2 scales, 8 channels, and center frequencies to extract features before finger-vein recognition [5]. Peng *et al.* used an 8-way Gabor filter generated by selecting optimal parameters. Among the images to which the Gabor filter is applied, the images with enhanced veins were fused to extract finger-vein patterns and the performance was evaluated through scale-invariant feature transform (SIFT) matching, which is robust to rotation and shift [8]. Shin *et al.* combined a 4-way Gabor filtered image with a Retinex filtered one by a fuzzy-based fusion to enhance image quality [10]. Park *et al.* extracted 8 gray profiles which lie at right angles to an 8-way vein-line and identified the gray profile which corresponded to the vein-line to determine the direction of the Gabor filter [11]. Zhang *et al.* proposed a gray-level grouping (GLG) which enhanced image contrast and a circular Gabor filter (CGF) which enhanced the quality of finger-vein image [12]. Apart from Gabor filter-based methods, Pi *et al.* proposed a quality enhancement method for finger-vein images using edge-preserving and elliptical high-pass filters that retain edges while eliminating blur [13]. Yu *et al.* proposed a fuzzy system-based multi-threshold method which took into account the characteristics of finger-vein patterns and skin areas [14]. Qian *et al.* proposed a finger-vein recognition algorithm based on the fusion of score-level moment invariants by the weighted-average method [15].

Furthermore, in order to extract features in consideration of the local patterns of finger-veins, Lee *et al.* used LBP and LDP descriptors [16]. In this study, a modified Gaussian high-pass filter was used for image enhancement, followed by binary code extraction by LBP and LDP. The dissimilarity between the extracted features and enrolled binary codes was calculated using the Hamming distance. LBP and LDP measure relative changes in brightness between the pixels of each image and the surrounding pixels. Therefore, they have the advantages of robustness against changes in image brightness and fast processing time. However, they repeat unnecessary calculations as most bits have the characteristics of consistency. To consider this issue, Yang *et al.* used a personalized best bit map (PBBM) method which used consistent bits [17]. A PBBM is created by the following process: (1) several samples of finger-vein images are captured for each individual and features are extracted to obtain corresponding LBP codes, and (2) some bits of the LBP codes extracted from the same location of same finger-vein image are consistent, and the values and locations of these bits are stored. Using the PBBM, only the consistent bits are used for recognition. Therefore, by reducing the influence of noise bits, it provides higher performance in less processing time compared to the existing LBP methods. Rosdi *et al.* introduced a local line binary pattern (LLBP) which used a straight-line shape filter unlike the square shape filter used by LBP, LDP, and PBBM. [18] In this study, filters of various sizes were applied and the test result showed that this method was superior in vein-line extraction compared to the method using a square shape filter. However, as LLBP extracts line patterns only in the horizontal and vertical directions, effective data from the image may not be fully utilized. Therefore, Lu *et al.* introduced a generalized local line binary pattern (GLLBP) which can alter the filter direction of LLBP [19]. As GLLBP adjusts the direction and length of filters in consideration of texture characteristics, it can better utilize finger-vein direction data than LLBP. In previous researches [20], [21], a line tracking method for vein-line extraction was introduced. However, these non-training-based finger-vein recognition studies used various types of distance-based matching for finger-vein recognition based on extracted finger-vein features. It is therefore difficult to secure good recognition performance for various finger-vein images obtained by different devices and environments.

To address this issue, the training-based methods of support vector machine (SVM) [22]–[25] and CNN [2], [3], [26], [27] methods have been suggested. The study in [22] is about a presentation attack (spoof attack) detection method which

determines whether recognition is being performed by the finger-vein of a live body or by a printed or fake finger-vein image captured by the camera. In this study, the classification of texture features extracted by steerable pyramids using SVM was performed. Wu et al. suggested a method of performing principal component analysis (PCA) and linear discriminant analysis (LDA) to reduce the dimension of the features of finger-vein images, and used SVM with an adaptive neuro-fuzzy inference system (ANFIS) to perform finger-vein recognition [23]. Qin *et al.* suggested a method in which sub-area matching values of vein shape, orientation, and SIFT features extracted from finger-vein images were fused and classified by SVM [24]. Khellat-kihel *et al.* proposed the enhancement of finger-vein images with a Gabor filter and classification by SVM [25].

Aside from these, studies on finger-vein recognition are underway using a deep learning-based method which provides a high level of recognition performance through learning from a large data set. In studies [26], [27], finger-vein recognition was performed using a reduced-complexity four-layer CNN with a fused convolutional-subsampling architecture. In these studies, the finger-vein images of a non-trained class cannot be used because the images of the same class was used during training and testing (closed world setting). For the recognition of images of non-trained classes (open world setting), features need to be extracted from the layer before the final fully connected layer (FCL) of the CNN obtained from a single image input, and a distance matching method needs to be used for recognition [28]. Or the CNN needs to be designed to use two types of images for authentic matching (matching in cases that input and enrolled images are of the same class) or imposter matching (matching in case that input and enrolled images are of different classes) as input to CNN in order to have two outputs [29]. Hong *et al.* conducted a test using the two methods [2]. Two VGG Net-16 networks designed with original and difference images as input were fine-tuned and features were extracted from the layer before the final FCL, and Euclidian distances between input and enrolled features were calculated for recognition. This method, however, has the disadvantage that not all layers of the trained network are used. The network which uses difference images as the input produces two outputs from a single input. Therefore, it has no need to extract features from the layer before the final FCL and can use all layers of the trained network. In addition, the test results showed that the network using difference images as input had higher accuracy. Kim et al. used a network with more depth than the one used in [2], and suggested a CNN-based multimodal biometric method based on finger-vein and finger shape [3]. In the case of finger-vein recognition, Resnet-50 and Resnet-101 networks, which use difference images as inputs, were fine-tuned. The Shandong University of homologous multi-modal traits (SDUMLA-HMT), an open database [30], was used to conduct a test. The results showed that it had higher accuracies than the method in [2]. However, as these difference images are generated by calculating the difference in pixel values of the two images, they can be more susceptible to noise as the changes in feature values of the images can decrease. In [56], Qin *et al.* proposed the method of CNN-based extraction of finger-vein lines. This research used the CNN and fully convolutional network (FCN) just for extracting finger-vein lines, and finger-vein verification was performed by template matching. Because our research does not extract finger-vein lines but uses the whole region of interest (ROI) of finger as the input to deep DenseNet for finger-vein verification, our research is different from the previous study [56]. As shown in Table 8, the error of finger-vein verification by our method (equal error rate (EER) of 0.33%) is much lower than that by [56] (EER of 3.02%). In [57], Fang *et al.* proposed a lightweight two-channel convolutional networks for finger-vein recognition. The training and testing data were from the same class in their experiments (close world setting). However, this kind of experimental scenario cannot be used because the class data in testing is unknown one, and they cannot be used to train the finger-vein recognition system in real-world application, consequently. Different from that, our method ensured that the data of the same class were not used in the training and testing (open world setting). That is, the classes in training were completely different from those in testing for our experiments. In [58], Xie *et al.* proposed the finger-vein authentication method based on lightweight CNN and supervised discrete hashing. However, the error of finger-vein verification by our method (equal error rate (EER) of 0.33%) is much lower than that by [58] (EER of 8.87%) as shown in Table 8.

Although Kumar *et al.*'s method used shift matching, they used hand-crafted features based on Gabor filter with morphological processing for finger-vein recognition [32]. Therefore, the error of finger-vein verification by our method (equal error rate (EER) of 0.33%) is lower than that by [32] (EER of 0.65%) as shown in Table 8. In [59], Yang *et al.* proposed finger vein recognition method, including an anatomy structure analysis-based vein extraction algorithm and an integration matching strategy. Although they also used shift matching, they used hand-crafted features for finger-vein recognition. Therefore, the error of finger-vein verification by our method (equal error rate (EER) of 0.33%) is lower than that by [59] (EER of 0.38%) as shown in Table 8. Although the recognition accuracy by their method with the Shandong University homologous multi-modal traits (SDUMLA-HMT) finger-vein database is higher than that by our method, the training and testing data were from the same class in their experiments (close world setting). However, this kind of experimental scenario cannot be used because the class data in testing is unknown one, and they cannot be used to train the finger-vein recognition system in real-world application, consequently. Different from that, our method ensured that the data of the same class were not used in the training and testing (open world setting). That is, the classes in training were completely different from those in testing for our experiments. In [60], Kang and Wu proposed contactless

TABLE 1. Summarized comparisons of previous and proposed methods of finger-vein recognition.

Category			Method	Advantage	Disadvantage
Non-training-based	Enhancement of vein direction		Gabor filter [4-12], edge-preserving and elliptical high-pass filters [13]	Image quality and recognition improve	- High quality images are required - Accuracy is affected by misalignment and shading
	Local patterns of finger-vein		LBP with LDP [16], PBBM [17]	Fast processing speed	
	Highlighting the vein-line features		LLBP [18], GLLBP [19]	Directional information of finger vein is highlighted	
Training-based	SVM		Sophisticated classification [22-25]	Accuracy is less affected by misalignment and shading	Require optimal feature extraction and dimension reduction
	CNN	Feature vector distance	Distance matching based on feature vectors from layer [2]	Can recognize the data of non-training classes	Cannot use all the layers of the trained network
		Difference image	Use difference image as input [2,3]	Higher accuracy by using all the layers of the trained network	Susceptible to noise
		Composite image (proposed method)	Use composite image as input and shift matching	The highest accuracy due to robustness to noise and misalignment	Data loss can occur in the composite image

palm vein recognition based on mutual foreground-based local binary pattern. Although they also used shift matching, their method is for contactless palm vein recognition, and it cannot be compared with our research of finger-vein recognition because the recognition modality is different.

To address the limitations of previous researches, this paper proposes a finger-vein recognition method which uses a composite image with robustness against noise as input to a deep DenseNet, and introduces shift matching to resolve performance degradation issues caused by misalignment between enrolled and input images. Table 1 provides an outline of the advantages and disadvantages of the methods proposed in this paper and in previous studies.

III. CONTRIBUTIONS

Our research contributes to the body of knowledge in the following three ways.

- Most of previous researches used AlexNet, visual geometry group (VGG)-Net, residual network (ResNet), and there is no previous study to use deep DenseNet for finger-vein recognition. Therefore, although we use the well-known technique of DenseNet in a known problem of finger-vein recognition, this study has the value as the first finger-vein recognition study using a deep DenseNet.
- The reason why previous researches did not adopt DenseNet for finger-vein recognition is that the numbers of layers and training complexity in this network are larger than those in AlexNet, VGG-Net, and ResNet. In order to overcome the problems of increase of numbers of layers and training complexity in DenseNet in addition to the disadvantages of the CNNs with two inputs, one difference image or the distance matching-based method using the CNN features, this study proposes a new method of combining two images into one composite image of three channels for the input to the DenseNet.
- The trained CNN model and algorithms developed in this study are made available in [31] to other researchers for fair performance evaluation.

IV. PROPOSED METHOD

A. OVERVIEW OF PROPOSED METHOD

Figure 1 shows the flowchart of our finger-vein recognition method. The captured finger image is binarized. Then, the broken boundaries of finger are restored, and in-plane rotation compensation is conducted (step (2) of Figure 1). Subsequently, a 4×20 mask is used to detect the upper and lower boundaries of the finger region with noise reduction, and the collapsed region inside the finger area is restored to obtain the final ROI (step (3) of Figure 1). The detected finger ROI is then transformed to the image of 224×224 by size normalization, and the enrolled and input images are combined into one composite image (step (4) of Figure 1). By an 8-way shift transformation, 8 enrolled images (pre-generated) and input images are combined with one another, and a total of 9 composite images (including one non-shift composite image) are input to CNN to obtain 9 matching scores. Among the 9 matching scores, the minimum score is determined as final matching score (steps (5) and (6) of Figure 1). With this score, finger-vein recognition is performed (step (7) of Figure 1).

B. PREPROCESSING AND IN-PLANE ROTATION COMPENSATION

To remove the background area from the captured image, the image is binarized as shown in Figure 2 (b).

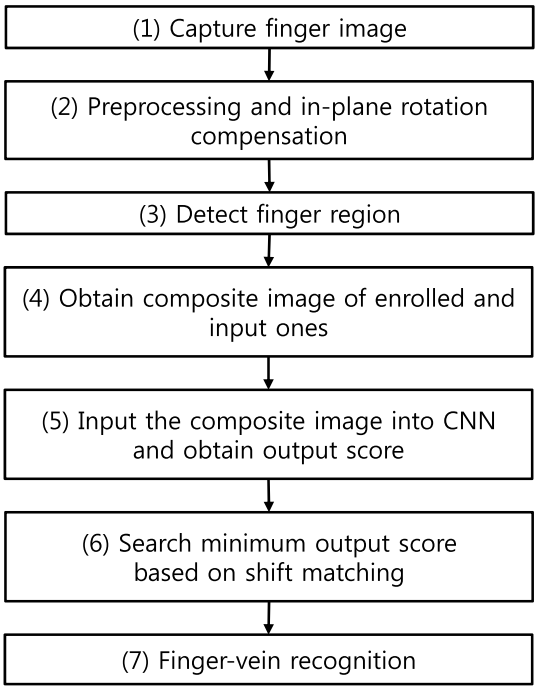

FIGURE 1. Flowchart of the proposed method.

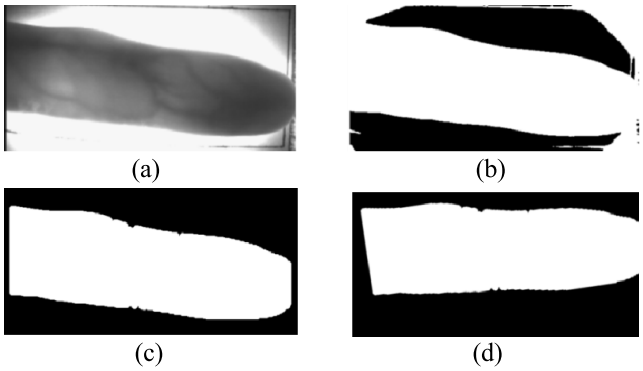

FIGURE 2. Example of input image and in-plane rotation compensation: (a) original image, (b) binarized image, (c) image with background removed, (d) image after in-plane rotation compensation.

However, the background near the finger area is not completely removed. Therefore, an edge map created by Sobel edge detector and the binarized image are used to create a difference image, and the area thresholding method [32] is applied to obtain an image without the background, as in Figure 2 (c). Then, the second order moments of binarized mark R(Figure 2 (c)) are calculated as in Equation (1).

$$\begin{aligned}\alpha_{11} &= \frac{\sum_{(x,y)\in R}(y-c_y)^2 \cdot I(x,y)}{\sum_{(x,y)\in R} I(x,y)},\\ \alpha_{12} &= \frac{\sum_{(x,y)\in R}(x-c_x)\left(y-c_y\right) \cdot I(x,y)}{\sum_{(x,y)\in R} I(x,y)},\\ \alpha_{22} &= \frac{\sum_{(x,y)\in R}(x-c_x)^2 \cdot I(x,y)}{\sum_{(x,y)\in R} I(x,y)}\end{aligned} \tag{1}$$

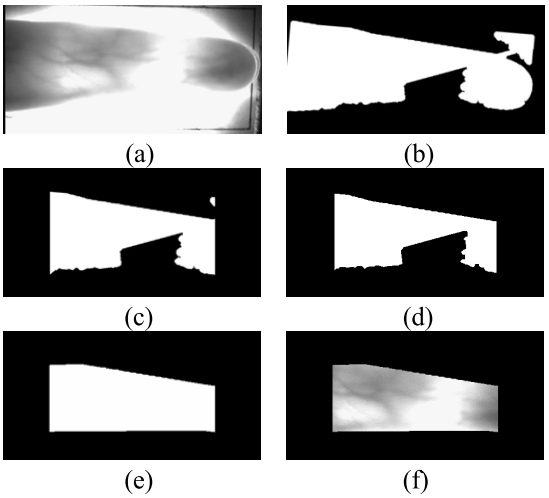

FIGURE 3. Procedure for detecting finger ROI for recognition: (a) original image, (b) the image of a rotated mask, (c) the image with the removal of left and right areas of (b), (d) the result image by component labeling with (c), (e) the ROI mask by filling the black region inside the finger area of (d), and (f) ROI image.

$I(x, y)$ and (c_x, c_y) indicate image pixel values and central coordination, respectively. Based on these, the angle of rotation ω in Equation (2) is calculated to compensate for in-plane rotation [33]. Finally, the compensated image is obtained by image rotation based on ω and bilinear interpolation as shown in Figure 2 (d).

$$\omega \begin{cases} tan^{-1}\left\{\dfrac{\alpha_{11}-\alpha_{22}+\sqrt{(\alpha_{11}-\alpha_{22})^2+4\alpha_{12}^2}}{-2\alpha_{12}}\right\} \\ \qquad\qquad if\ \alpha_{11} > \alpha_{22} \\ tan^{-1}\left\{\dfrac{-2\alpha_{12}}{\alpha_{22}-\alpha_{11}+\sqrt{(\alpha_{22}-\alpha_{11})^2+4\alpha_{12}^2}}\right\} \\ \qquad\qquad if\ \alpha_{11} \le \alpha_{22} \end{cases} \tag{2}$$

C. DETECTION OF FINGER ROI

The left and right end areas are usually the thick part of the finger or fingernail as shown in Figure 3 (a), where the lighting is inadequate. As the vein patterns of these areas are not captured accurately and are not useful for recognition, these areas are removed by predetermined size. Then, noise, as shown in Figure 3 (c), is removed using a component labeling method to obtain an image as in Figure 3 (d). The vein-patterns in the black region inside the finger area cannot not be observed due to bright lighting, and this region is filled with the average value of the surrounding pixels using a 4×20 mask in order to create an ROI mask (Figure 3 (e)). The ROI mask is then used to obtain the ROI image as in Figure 3 (f).

D. IMAGE COMPOSITION FOR THE INPUT TO CNN

The previous studies on CNN-based finger-vein recognition used an original image or a difference image as an

TABLE 2. DenseNet architecture used for our research. Each "conv" layer consists of the sequence of batch normalization (BN), rectified linear unit (ReLU), and convolution layer. "s1" and "s2" represent strides of 1 and 2 pixels, respectively.

Layer		Input Size	Output Size
Image Input layer		$224 \times 224 \times 3$	$224 \times 224 \times 3$
7×7 conv, s2		$224 \times 224 \times 3$	$112 \times 112 \times 96$
2×2 max pooling, s2		$112 \times 112 \times 96$	$57 \times 57 \times 96$
Dense Block (1)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 3 \times 3 \text{ conv, s1} \end{bmatrix} \times 6$	$57 \times 57 \times 96$	From $57 \times 57 \times 96$ to $57 \times 57 \times 384$ by the depth increment of 48
Transition Layer (1)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 2 \times 2 \text{ max pooling, s2} \end{bmatrix}$	$57 \times 57 \times 384$	$29 \times 29 \times 192$
Dense Block (2)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 3 \times 3 \text{ conv, s1} \end{bmatrix} \times 12$	$29 \times 29 \times 192$	From $57 \times 57 \times 192$ to $57 \times 57 \times 768$ by the depth increment of 48
Transition Layer (2)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 2 \times 2 \text{ max pooling, s2} \end{bmatrix}$	$29 \times 29 \times 768$	$15 \times 15 \times 384$
Dense Block (3)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 3 \times 3 \text{ conv, s1} \end{bmatrix} \times 36$	$15 \times 15 \times 384$	From $57 \times 57 \times 384$ to $57 \times 57 \times 2112$ by the depth increment of 48
Transition Layer (3)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 2 \times 2 \text{ max pooling, s2} \end{bmatrix}$	$15 \times 15 \times 2112$	$8 \times 8 \times 1056$
Dense Block (4)	$\begin{bmatrix} 1 \times 1 \text{ conv, s1} \\ 3 \times 3 \text{ conv, s1} \end{bmatrix} \times 24$	$8 \times 8 \times 1056$	From $57 \times 57 \times 1056$ to $57 \times 57 \times 2208$ by the depth increment of 48
Classification Layer	8×8 global average pooling	$8 \times 8 \times 2208$	$1 \times 1 \times 2208$
	2D fully-connected, softmax	$1 \times 1 \times 2208$	2

input [2], [3]. To solve the disadvantage of these studies, a composite image is used in this study. The composite image is generated by compositing enrolled and input images as following process: (1) an ROI image of Figure 3 (f) is stretched by bilinear interpolation and normalized to a size of 224×224 pixels as shown in Figures 4 (a) and (b). (2) The enrolled and input images, as in Figure 4 (a) and (b), are input as the first and second channels of the 3-channel image. For the third channel, the enrolled and input images are resized to 224×112 and they are concatenated vertically to create a composite image as in Figure 4 (d).

E. CNN-BASED FINGER-VEIN RECOGNITION

In this study, the output layer of pre-trained DenseNet-161 [34] is modified and fine-tuned. The data used in the test is the composite image created with the data class comprising two types of authentic matching and imposter matching. Therefore, the output of the classification layer is modified to be a 2D fully-connected layer for DenseNet-161 to have two outputs as shown in Table 2. As the low-level layers of the pre-trained model have generic features and there is no need to train the entire network, only the final fully-connected layer is fine-tuned. Sections IV.*E*.1 and IV.*E*.2 give the detail explanations of DenseNet architecture and shift matching method, respectively.

1) DenseNet ARCHITECTURE

Suppose a single image x_0 passed through the convolutional network of L layers, and the lth layer including $H_l(.)$ as a non-linear transformation. $H_l(.)$ is the function including the operations of convolution, pooling, batch normalization (BN), or a rectified linear unit (ReLU). AlexNet [35] and VGG Net [36] as feed forward networks, connect the output of the $(l-1)$th layer to the input of the lth layer as shown in Equation (3).

$$x_l = H_l(x_{l-1}) \tag{3}$$

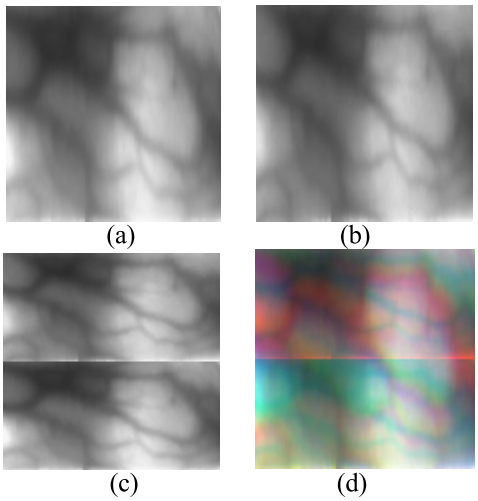

FIGURE 4. Examples of input and enrolled images of the same class with their corresponding composite images: (a) enrolled image (1st channel), (b) input image in same class (2nd channel), (c) concatenated image of (a) and (b) (3rd channel), (d) composite image of three channels.

ResNet [37] introduces skip-connections and uses identity function to add x_{l-1} as shown in Equation (4).

$$x_l = H_l\left(x_{l-1}\right) + x_{l-1} \tag{4}$$

DenseNet uses dense connectivity which has improved the skip connection structure of ResNet. This is the method of concatenating the feature maps of the lth layer and previous layers within the dense block. Therefore, the input of the lth layer consists of the concatenated feature map of the previous layers $(x_0, x_1, \ldots, x_{l-1})$ as shown in Equation (5) [34].

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \tag{5}$$

The dense block of Figure 5 concatenates the features of previous and subsequent layers, and transfer them to the subsequent layer. This prevents signal attenuation caused by the growing depth of a layer and improves performance. However, as the network grows deeper, the number of channels of the concatenated feature map becomes too high and this greatly increases the size of the network. To prevent this, DenseNet adds a bottleneck layer between the layers within the dense block. The bottleneck layer consists of BN, ReLU, a 1×1 convolutional filter (conv), BN, ReLU, and a 3×3 conv as shown in Figure 6. Using the bottleneck structure prevents the increase of feature map size with the reduction of computational cost. Nonetheless, since the output of the dense block concatenates all the layers within the block, the deeper the layer becomes with the more layers in the dense block, the greater the feature map size becomes. To address this issue, a transition layer is added between dense blocks to reduce the feature map size, as shown in Figure 7. The transition layer halves the number of channels of the feature map through 1×1 conv calculations, and also halves the width and height through 2×2 average pooling. Each layer in the dense block outputs a feature map according to the size of

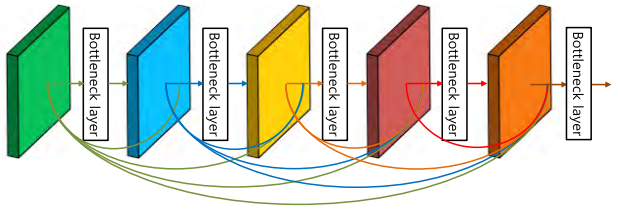

FIGURE 5. Example of a dense block.

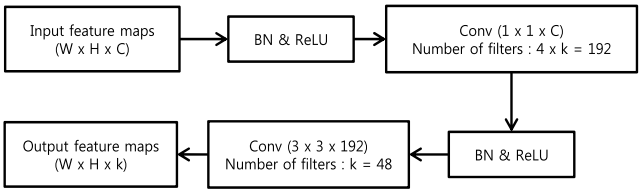

FIGURE 6. Bottleneck layer design: C is the number of channels in the input feature map and k is the growth rate of DenseNet-161.

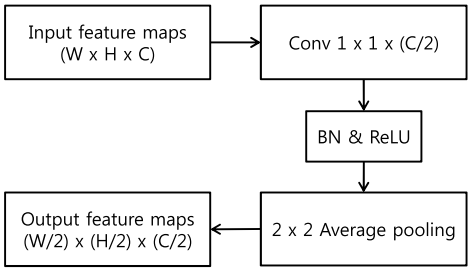

FIGURE 7. Transition layer design: C is the number of channels of the input feature map.

the growth rate, and the growth rate of DenseNet-161 is 48. We show the overall structure of DenseNet-161 used in our study as shown in Table 2.

2) SHIFT MATCHING

A shift matching method is proposed to address performance loss during finger-vein recognition caused by misalignment. The shift matching process is as follows: (1) translation (shifting) transforms of 5 pixels are applied to pre-enrolled images in 8 directions of up, down, right, left, and diagonal as shown in Figure 8; (2) 9 composite images are created with a total of 9 enrolled images that have been transformed and 1 input recognized image; (3) the 9 composite images are inputted to the CNN, and the minimum score among all the obtained ones is determined as the final matching score. Based on this score, it is decided whether it is genuine or imposter matching recognition. Genuine matching refers to matching where the enrolled and input images are in the same class, whereas imposter matching refers to matching where the enrolled and input images are in different classes. When deciding between genuine and imposter matching based on the final matching score, the following method is used: genuine matching occurs when the score is lower than the threshold determined based on the equal error rate (EER) of the genuine and imposter matching distributions obtained from

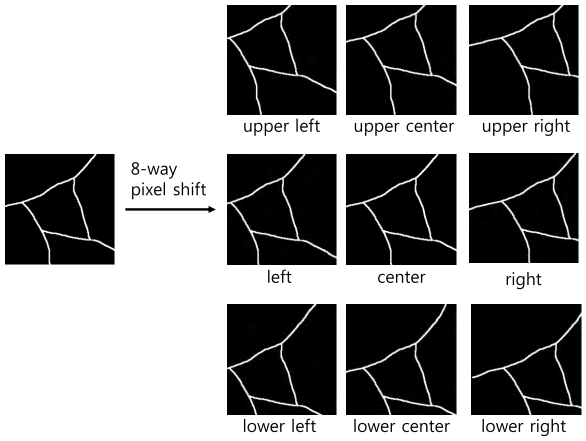

FIGURE 8. Example of 8-way pixel shifting.

the training data; imposter matching occurs in the other case. Here, EER is the error rate at which the false acceptance rate (FAR, the error rate of incorrectly accepting imposter data as genuine) is equivalent to the false rejection rate (FRR, the error rate of incorrectly rejecting genuine data as imposter data).

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL DATA

Two types of finger-vein databases were used for experiments. The first database was the Hong Kong Polytechnic University finger image database (version 1) comprising two sessions [32]. In Session 1, 6 images of index and middle fingers were collected from a total of 156 people, resulting in 1,872 images in total (156 people × 2 fingers × 6 images). In Session 2, 6 images of index and middle fingers were collected from 105 of the 156 people from Session 1, resulting in 1,260 images in total (105 people × 2 fingers × 6 images). In this study, only the images of Session 1 were used. The second database was the Shandong University homologous multi-modal traits (SDUMLA-HMT) finger-vein database. It comprises 6 images of index, middle, and ring fingers of both hands from 106 people, resulting in 3,816 images in total (106 people × 2 hands × 3 fingers × 6 images). In our study, the Hong Kong Polytechnic University finger image database (version 1) is referred to as "PolyU-DB" and the SDUMLA-HMT database as "SDU-DB." In this paper, two-fold cross-validation was conducted in the test. Because PolyU-DB and SDU-DB comprise 312 and 636 classes, respectively, the images of 156 classes of PolyU-DB were used for training and the remaining 156 classes were used for testing in the 1st fold validation. In SDU-DB, the images of 318 classes were used for training and the remaining 318 classes were used for testing. In the 2nd fold validation, the training data and testing data used in the 1st fold validation were exchanged, and training and testing were re-conducted. This method ensured that the data of the same class were not used in the training and testing (open world setting). Also, the average accuracy measured through these two tests was used as the final recognition accuracy.

B. DATA AUGMENTATION

The two databases of PolyU-DB and SDU-DB have a low number of images that are not enough for training the various weights within the deep CNN structure. This can, therefore, cause overfitting problems. To address this issue, prior to creating a composite image as described in Section IV.*D*, a data augmentation method was performed to increase the size of the training data sets. The data augmentation method involves a 3-pixel translation of the image in the vertical direction and a 5-pixel translation of the image in the horizontal direction to generate 4 additional images, increasing the original number of images five-fold. This kinds of translation-based data augmentation has been commonly used in previous research [35]. Table 3 contains the descriptions of augmented images of PolyU-DB and SDU-DB. As described in Section V.A, half of all classes of the database were used for training data and the other half were used for testing data. In other words, PolyU-DB used 156 classes and SDU-DB used 318 classes to generate one training or testing dataset. In the case of the composite images in Table 3, PolyU-DB increased the number of the original images of 156 classes five-fold to generate a total of 4,680 (156 classes × 6 images × 5 times) augmented images.

SDU-DB also used the same method to increase the number of images five-fold to generate a total of 9,540 augmented images. Composite images for authentic matching are made by combining the images in the same class. Training was performed based on the authentic matching and imposter matching data obtained in this way. However, there are too many imposter matches compared to authentic matches in general. Therefore, the same number of imposter matching images was randomly selected to that of authentic matching images as training data. The process for generating the composite image of SDU-DB is the same as above. This data augmentation technique was applied only to the training data, whereas the original data was used as testing data set which was not augmented with the composite and difference images. In this study, training and testing were performed with a desktop computer comprising an Intel® Core™ i7-3770K CPU @ 3.5 GHz (4 cores) with 24 GB RAM and an NVIDIA GeForce GTX 1070 (1920 compute unified device architecture (CUDA) cores) graphics processing unit (GPU) card with graphics memory of 8 GB [38]. Caffe framework [39] and Microsoft Visual Studio 2013 [40] were used to implement the algorithm.

C. TRAINING OF CNN MODEL

Conventional CNN models (VGG Net-16 [36] and ResNet-152 [37]) were compared with DenseNet to evaluate the performance in relation to different input methods. Table 4 provides hyper parameter values used for training with each model. As shown in Table 4, three types of input were used for PolyU-DB and SDU-DB: a composite image, difference image, and an original image. For each type of input, 3 CNN models were fine tuned.

TABLE 3. Descriptions of the two databases with augmented data.

			PolyU-DB	SDU-DB
Original images		# of images	1,872	3,816
		# of people	156	106
		# of hands	1	2
		# of fingers	2 (index and middle fingers)	3 (index, middle, and ring fingers)
		# of classes (# of images per class)	312 (6)	636 (6)
Data augmentation for training	Composite image	# of augmented images	4,680 (156 classes × 6 images × 5 times)	9,540 (318 classes × 6 images × 5 times)
	Difference image	# of augmented images		
	Original image	# of augmented images		

For training these models, a stochastic gradient descent (SGD) method is used, which multiplies the learning rate by gamma values for each step size (epoch) in the unit of a mini-batch to reduce the learning rate and achieve rapid convergence with high training accuracy and low loss [41]. The total number of training datasets divided by the mini-batch size is named as the number of iterations, and the 1 epoch refers to when the training progressed according to the number of iterations. Therefore, the max number of iterations is equivalent to the number of iterations multiplied by the number of the epoch. Figure 10 shows the training loss and accuracy of the DenseNet-161 model which used composite images from PolyU-DB. Training loss is close to 0 and the accuracy is close to 100, suggesting that the CNN training is sufficient. The training time was 19 hours and 49 minutes in total.

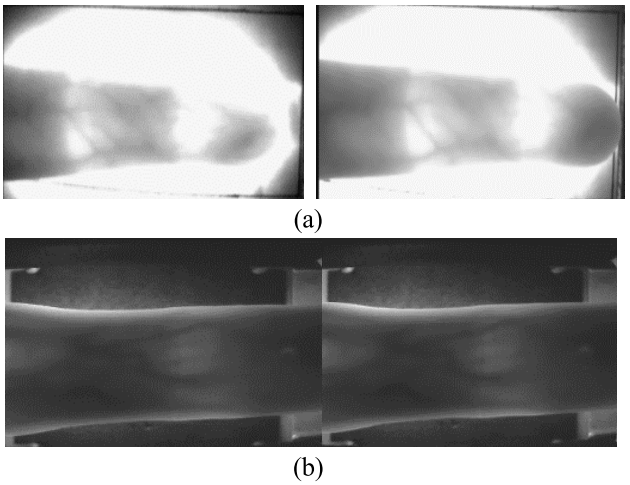

FIGURE 9. Example images of different trials from the same finger of one individual with (a) PolyU-DB and (b) SDU-DB.

D. TESTING OF PROPOSED METHOD ON PolyU-DB

The recognition error rate was measured using FAR and genuine acceptance rate (GAR). Here, GAR is calculated as 100-FRR (%). In this test, the output score value of the CNN or the distance between feature vectors of the fully connected layer were used to measure FAR and FRR. In the first test, the impact of the noise in the original image on the recognition accuracy was examined when a composite image and difference image were used as input to CNN.

Figures 11 (a) and (b) are the resulting images when Gaussian random noise of 0 mean and 0.05 standard deviation was applied to the original image. Figures 11 (c) and (d) are the composite and difference images generated from Figures 11 (a) and (b), respectively. To compare the recognition accuracy, a test was conducted using our DenseNet-161 structure and shift matching was not applied.

As shown in Table 5, the recognition accuracy for the difference image is highly affected by noise compared to composite images. Therefore, composite images have more robustness against noise than difference images.

In a subsequent test, the recognition accuracy by proposed minimum score selection based on shift matching was compared with non-shift matching and the fusion method of weighted average-based matching distances. As shown in Table 6 and the receiver operating characteristic (ROC) curves in Figure 12, shift matching showed better performance results than non-shift matching in most cases.

TABLE 4. Training parameters used for comparison of the performance using different input methods and different CNN models.

Dataset		# of output class	CNN model	# of max epoch	Mini-batch size	Learning rate/step size	Momentum/ gamma
PolyU-DB	Original image	156	VGG Net-16	12	30	0.0001	0.9/0.5
			ResNet-152	12	6		
			DenseNet-161	8	4		
	Difference image	2	VGG Net-16	12	30	0.0001/6	
			ResNet-152	12	6		
			DenseNet-161	12	4		
	Composite image	2	VGG Net-16	14	30	0.0001/9	
			ResNet-152	14	6		
			DenseNet-161	14	4		
SDU-DB	Original image	318	VGG Net-16	8	30	0.0001	
			ResNet-152	4	6		
			DenseNet-161	8	4		
	Difference image	2	VGG Net-16	6	30	0.0001	
			ResNet-152	6	6		
			DenseNet-161	6	4		
	Composite image	2	VGG Net-16	4	30	0.0001	
			ResNet-152	4	6		
			DenseNet-161	3	4		

TABLE 5. EER of finger-vein recognition according to input image with or without noises (unit: %).

Method		EER
Composite image	Without Gaussian random noise	0.44
	With Gaussian random noise	1.22
Difference image	Without Gaussian random noise	1.78
	With Gaussian random noise	37.15

In addition, the MIN rule method (proposed minimum score selection) showed the best performance result among the shift matching methods.

In a following test, accuracy was compared by changing the inputs to various CNN structures for PolyU-DB. As shown in Table 7, the highest performance of 0.33% was observed when the input to the DenseNet-161 structure was a composite image. As mentioned above, the method using original images refers to distance matching using feature vectors extracted from the layer before the FCL obtained from enrolled and input images. Figure 13 provides the results in ROC curves. Like Figure 12, this is an average curve of the two ROC curves obtained from two-fold cross-validation. As can be seen in Figure 13, using the DenseNet-161 with composite images showed the highest performance.

In the next test, non-training based methods were compared with the DenseNet-161 with shift matching. The non-training based methods includes a Gabor filter with morphological processing [32], a multi-scale matched filter with line tracking [42], a personalized best patches map (PBPM) [43], singular value decomposition (SVD)-based minutiae matching (SVDMM) [44], subpixel-based features (SPFs) [45], discriminative binary codes (DBC) [46], and discriminative binary descriptors (DBD) [47]. The training-based method using a ResNet-101 [3] were also compared. As shown in Table 8, the results indicate that our method

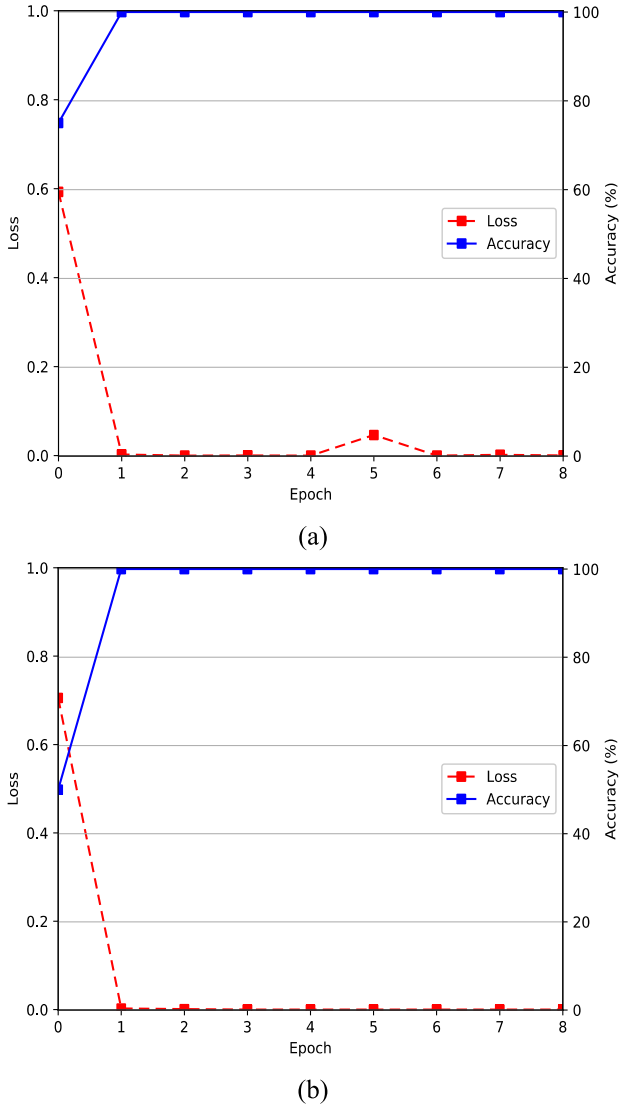

FIGURE 10. Examples of loss and accuracy curves of DenseNet-161 with training data of two-fold cross validation with PolyU-DB. (a) The 1st fold, (b) the 2nd fold.

showed higher accuracy than the methods proposed in previous studies.

Figures 14 (a) and (b) show correct recognition cases by the proposed method. Figure 14 (a) is the case of authentic matching and it shows that the vein pattern of the input image has moved compared to the enrolled image due to misalignment upon image capture. Using the shift matching method, the misalignment was solved to allow accurate recognition. Figure 14 (b) is the case of imposter matching and shows that it is accurately recognized as imposter matching despite that the vein patterns have been captured with low brightness.

In the false rejection case, the vein patterns are not clear as the vein area is dark and there is a large degree of misalignment. In this case, the misalignment could not be solved with the composite image and shift matching. In the false acceptance case, as the vein area of the input image was captured with very low brightness, the vein patterns are not

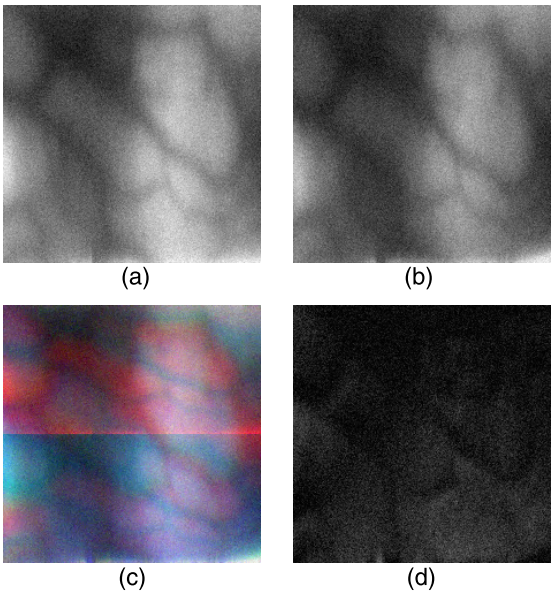

FIGURE 11. Example images with noise. (a), (b) enrolled and recognized images with Gaussian random noise. (c) Composite image generated from (a) and (b). (d) Difference image generated from (a) and (b).

clearly seen. Also, the shading of the enrolled and input images is similar. Therefore, the false acceptance case occurs.

Figures 15 (a) and (b) respectively show false rejection and false acceptance cases by the proposed method.

In the next test, the processing speed of the DenseNet-161-based finger-vein recognition was measured. The measurement was performed with the desktop computer explained in Section V.*B* and the Jetson TX2 embedded system [48] in Figure 16. The reason for also performing measurements on the embedded system is because in most cases of finger-vein recognition for access control, the system is more widely used by an on-board computing (edge computing) as an embedded system for an entrance door than a desktop computer-based server-client computing (cloud computing). The feasibility of on-board computing of the proposed system was therefore evaluated. The Jetson TX2 has a NVIDIA Pascal™-family GPU (256 CUDA cores) with 8 GB of memory shared between the central processing unit (CPU) and GPU, and 59.7 GB/s of memory bandwidth; it uses less than 7.5 Watts of power. Proposed algorithm of finger-vein recognition was implemented by Keras-Tensorflow [49] on the Ubuntu 16.04 operating system [50]. More specifically, we setup python version 3.5, Tensorflow-GPU version 1.12, NVIDIA CUDA® toolkit 9.0, and NVIDIA CUDA® deep neural network library (cuDNN) version 7.3 on the computer. As shown in Table 9, the DenseNet-161-based recognition method using shift matching required 116 ms for the recognition of one image on the desktop computer and 2038 ms on the Jetson TX2 embedded system. Processing speed on Jetson TX2 system was lower as computing resources were limited compared to the desktop computer, but it was confirmed that the proposed method could be applied to the embedded system with limited computing resources.

TABLE 6. EER of finger-vein recognition with or without shift matching and various score fusions (unit: %).

Method			EER
VGG Net-16 [36]	Non-shift matching		0.96
	Shift matching	Weighted sum	0.82
		MIN rule	0.71
ResNet-152 [37]	Non-shift matching		0.94
	Shift matching	Weighted sum	0.76
		MIN rule	0.51
DenseNet-161	Non-shift matching		0.44
	Shift matching	Weighted sum	0.38
		MIN rule (proposed)	0.33

TABLE 7. EER of finger-vein recognition according to different types of input image and CNN models (unit: %).

Method		EER
Original image	VGG Net-16 [36]	4.57
	ResNet-152 [37]	2.29
	DenseNet-161	1.17
Difference image	VGG Net-16 [36]	1.49
	ResNet-152 [37]	1.08
	DenseNet-161	1.01
Composite image	VGG Net-16 [36]	0.71
	ResNet-152 [37]	0.51
	DenseNet-161 (proposed)	0.33

TABLE 8. Comparisons of EER by proposed method with previous algorithms on PolyU-DB (unit: %).

Method		EER
Non-training based method	SVDMM [44]	5.01
	Multi-scale matched filter with line tracking [42]	4.47
	PBPM [43]	3.53
	SPFs [45]	1.47
	DBC [46]	1.44
	DBD [47]	0.69
	Gabor filter with morphological processing [32]	0.65
	Yang et al.'s method [59]	0.38
Training-based method	Xie et al.'s method [58]	8.87
	Qin et al.'s method [56]	3.02
	ResNet-101 [3]	1.0779
	DenseNet-161 using shift matching (proposed)	0.33

E. TESTING WITH SDU-DB

Next, the performance was evaluated with SDU-DB. Table 10 shows the EER results for different inputs to various CNN structures. When composite images are used as input to the DenseNet-161 structure, the highest performance of 2.35% was obtained. Figure 17 shows the ROC curves of the recognition accuracies. These curves are expressed as averages of the two ROC curves obtained from two-fold cross-validation. As shown in Figure 17, the highest accuracy was obtained when the DenseNet-161 network and composite images were used.

In the next test, the accuracies of the proposed and previous methods are compared. The non-training based methods of Gabor + LBP [51], repeated line tracking [20], and maximum curvature [52], and the training-based methods of VGG Net-16 [2], and a ResNet-101 [3] based finger-vein recognition method were compared with our method. As shown in Table 11, our method showed higher accuracy than the methods proposed in previous studies.

Figures 18 (a) and (b) show the correct recognition cases with the proposed method. Figure 18 (a) is the case of authentic matching and shows that the vein pattern of the input image has significantly moved compared to the enrolled image due to misalignment upon image capture. Using the shift matching, the misalignment was solved and recognition was successful. Figure 18 (b) is the case of imposter

TABLE 9. Comparisons of processing speed by proposed method on desktop computer and embedded system (unit: ms).

	DenseNet-161 using shift matching (proposed)
Desktop computer	116
Jetson TX2	2038

TABLE 10. Comparisons of finger vein recognition accuracies according to different types of input image and CNN models (unit: %).

Method		EER
Original image	VGG Net-16 [36]	3.27
	ResNet-152 [37]	3.02
	DenseNet-161	3.47
Difference image	VGG Net-16 [36]	2.94
	ResNet-152 [37]	2.88
	DenseNet-161	2.48
Composite image	VGG Net-16 [36]	2.88
	ResNet-152 [37]	2.52
	DenseNet-161 (proposed)	2.35

TABLE 11. Comparisons of EER by proposed method with previous algorithms on SDU-DB (unit: %).

	Method	EER
Non-training based method	Gabor + LBP [51]	8.096
	Repeated line tracking [20]	5.46
	Maximum curvature [52]	4.54
Training-based method	VGG Net-16 [2]	3.906
	ResNet-101 [3]	3.3653
	DenseNet-161 using shift matching (proposed)	2.35

matching and shows that it was accurately recognized despite the unclear vein patterns of the input image.

Figures 19 (a) and (b) show the false rejection and false acceptance cases with the proposed method. In the case of false rejection, shading and unclear vein patterns of the enrolled and input images are observed. Also, they are not correctly recognized due to different shading patterns

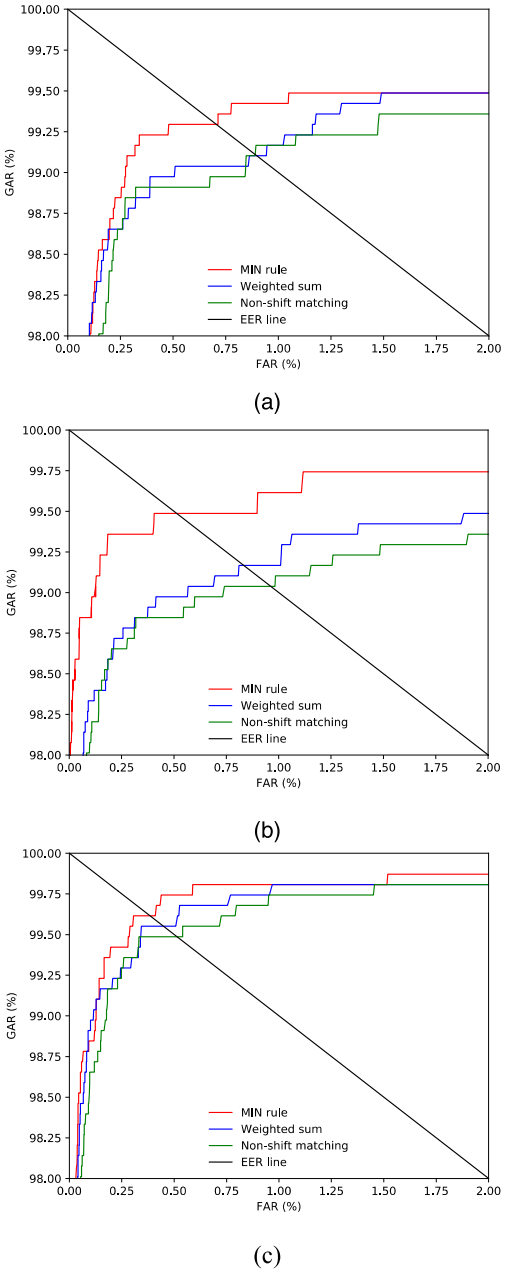

FIGURE 12. ROC curves of finger-vein recognition with or without shift matching and various score fusions: (a) VGG Net-16, (b) ResNet-152, and (c) DenseNet-161.

between the two images. In the false acceptance case, the vein patterns on the left areas of the two images are similar, and those in the right area of the input image were not clearly captured. In other words, it appears that recognition was not successful due to the similarity between local areas.

F. ANALYSIS OF FEATURE MAP AND SHIFT MATCHING

In this section, the feature map of the DenseNet was analyzed according to the depths of layer. The analysis results were presented by the channel of feature map, as the dimension size of feature map was too large. Figure 20 shows the examples of the feature maps extracted from an authentic matching image

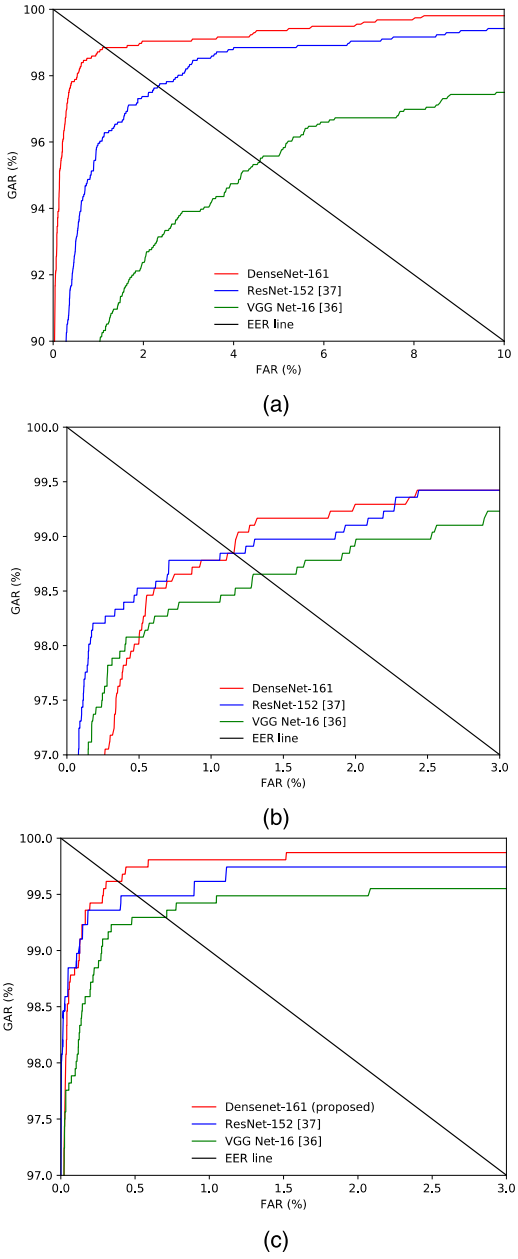

FIGURE 13. EER of finger-vein recognition according to different types of input image and CNN models. Using (a) original image, (b) difference image, and (c) composite image.

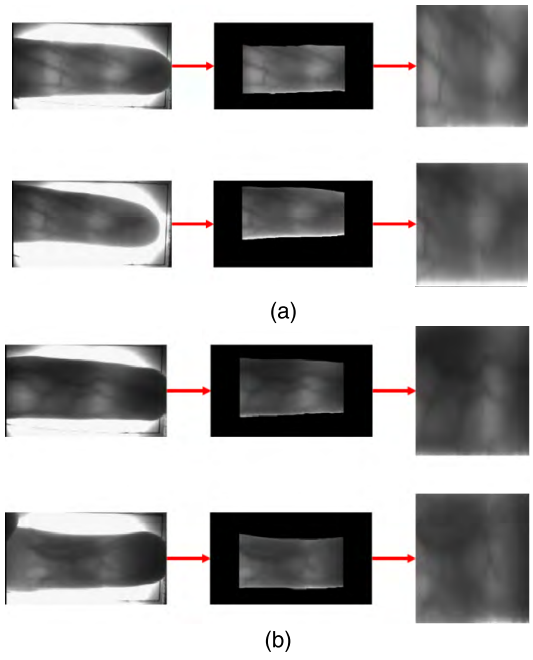

FIGURE 14. Examples of the correct recognition cases. (a) Authentic matching. (b) Imposter matching. Upper and lower images in (a) and (b) show the enrolled and input images, respectively. In (a) and (b), the images are an original image, ROI, and a resized image (224 × 224) from the left to the right.

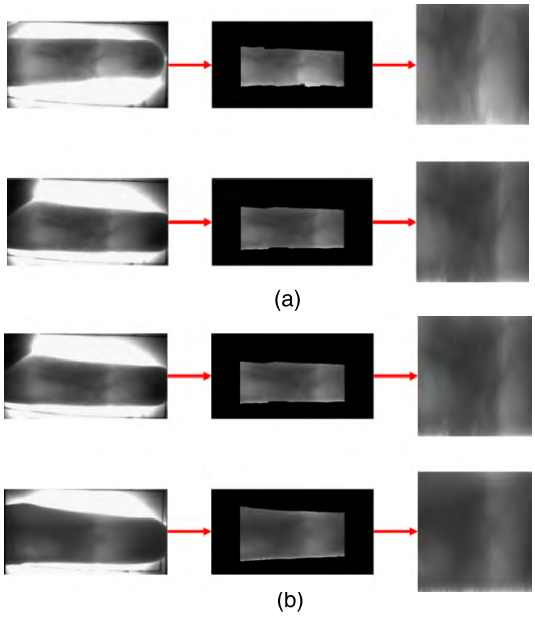

FIGURE 15. Examples of incorrect recognition cases. (a) False rejection case. (b) False acceptance case. Upper and lower images in (a) and (b) show the enrolled and input images, respectively. In (a) and (b), the images are an original image, ROI, and a resized image (224 × 224) from the left to the right.

(Figures 20 (a), (c), (e), (g), (i), and (j)) and an imposter matching image (Figures 20 (b), (d), (f), (h), (k), and (l)) on each layer of the DenseNet. Figures 20 (a) and (b) are the feature maps extracted from the 7 × 7 conv of Table 2. Figures 20 (c) and (d) are the output feature map of the 1 × 1 conv of the transition layer (1) of Table 2. Figures 20 (e) and (f) are the output feature map of the 1 × 1 conv of the transition layer (2) of Table 2. Figures 20 (g) and (h) are the output feature map of the 1 × 1 conv of the transition layer (3) of Table 2. Figures 20 (i) and (k) are the output feature map of the dense block (4) of Table 2. Figures 20 (j) and (l) are the 3D feature map images generated based on the average feature map values of Figures 20 (i) and (k), respectively.

For example, the size of the output feature map extracted from the 7 × 7 conv layer of Table 2 is 112 × 112 × 96, and 96 feature maps of 112 × 112 size are displayed

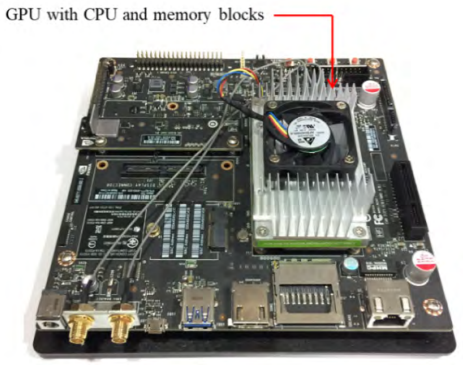

FIGURE 16. Jetson TX2 embedded system.

in Figures 20 (a) and (b) from top left to bottom right. As shown in Figure 20, the deeper the layer is, the more abstract the extracted feature is. For example, Figures 20 (a) and (b) maintained the vein-line and high-frequency edge components of the original image, but Figures 20 (i) and (j) lost the shape of the vein-line and maintained only the abstracted low-frequency features. As in Figures 20 (a)-(i) and (k), there appears to be no significant difference in the feature maps in the case of authentic matching and imposter matching. However, the 3D feature map values obtained from the average feature map values show moderate changes in the case of Figure 20 (j), which is the authentic matching result at the stage immediately before the classification layer in Table 2. On the other hand, Figure 20 (l), which is the imposter matching result, shows relatively more changes in feature map values compared to Figure 20 (j). Based on this, a difference was confirmed in the CNN feature maps of the authentic and imposter matching due to composite images.

As explained in [53], the recognition error in biometric system can be determined by genuine matching distribution (GMD) and imposter matching distribution (IMD). The genuine matching shows the case that the enrolled and input biometric data are from same class whereas the imposter matching represents the case that the enrolled and input biometric data are from different classes. In general, the more overlapped these two distributions are, the larger the recognition error becomes. With shift matching, the decrement of standard deviation of the GMD is comparatively larger than the amount of approaching by the IMD to the GMD, which causes the final reduction of recognition error [53].

To prove this, we experimentally checked the change of the GMD and IMD without and with shift matching algorithm as shown in Figures 21 (a) and (b), respectively. Although the IMD approaches to the GMD with shift matching (the mean of 14.75 in the IMD as in Figure 21 (a) becomes smaller as that of 12.52 as in Figure 21 (b)), the decrement of standard deviation of the GMD is much larger (the standard deviation of 5.43 in the GMD as in Figure 21 (a) becomes smaller as that of 3.68 as in Figure 21 (b)). Consequently, equal error

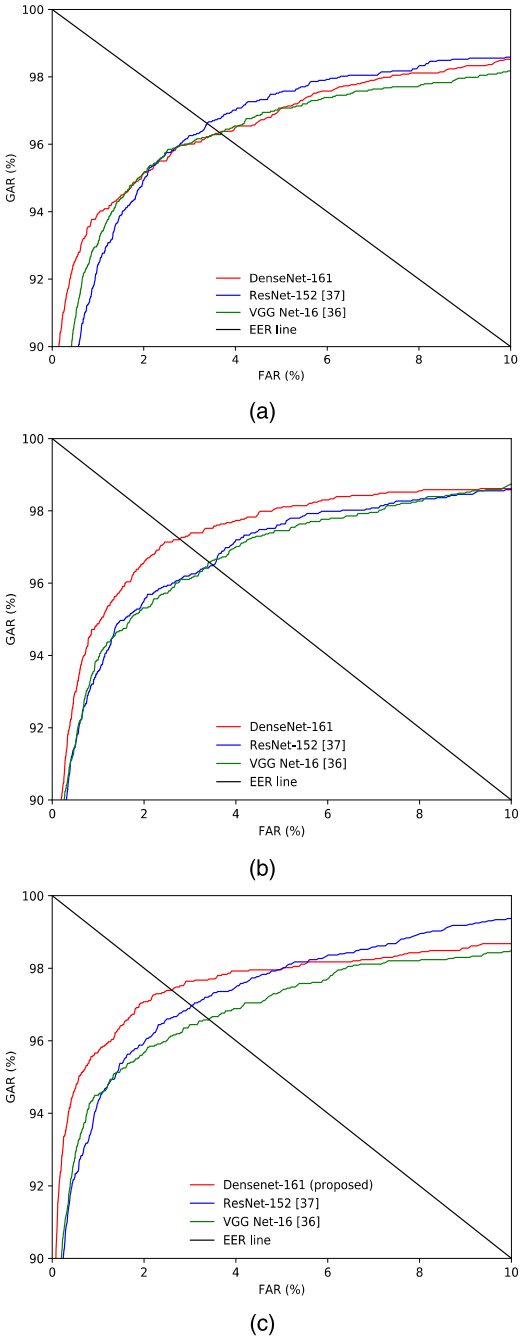

FIGURE 17. EER of finger-vein recognition according to different types of input image and CNN models. Using (a) original images, (b) difference images, and (c) composite images.

rate (EER) of recognition with shift matching (EER of 0.39%) of Figure 21 (b) becomes lower than that without shift matching (EER of 0.64%) of Figure 21 (a).

In conventional biometric system, the evaluation of accuracy has been also conducted by using the decidability value (d' value) of Equation (6) [54].

$$d' = \frac{|\mu_G - \mu_I|}{\sqrt{\frac{1}{2}(\sigma_G^2 + \sigma_I^2)}} \tag{6}$$

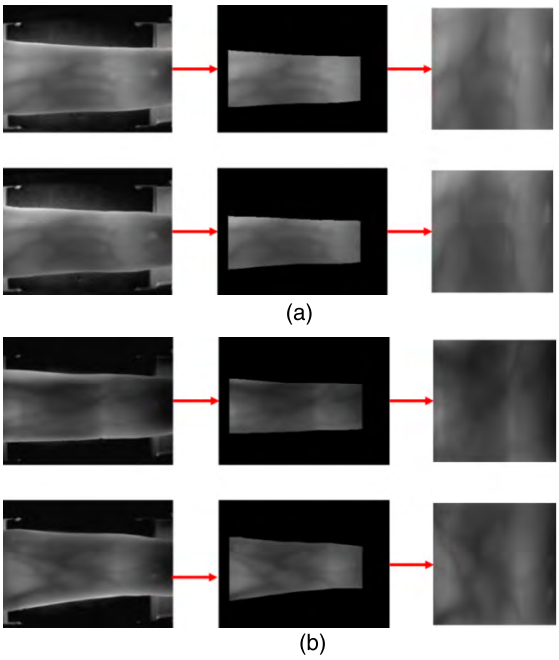

FIGURE 18. Examples of the correct recognition cases. (a) Authentic matching. (b) Imposter matching. Upper and lower images in (a) and (b) show the enrolled and input images, respectively. In (a) and (b), the images are the original image, ROI, and a resized image (224 × 224) from the left to the right.

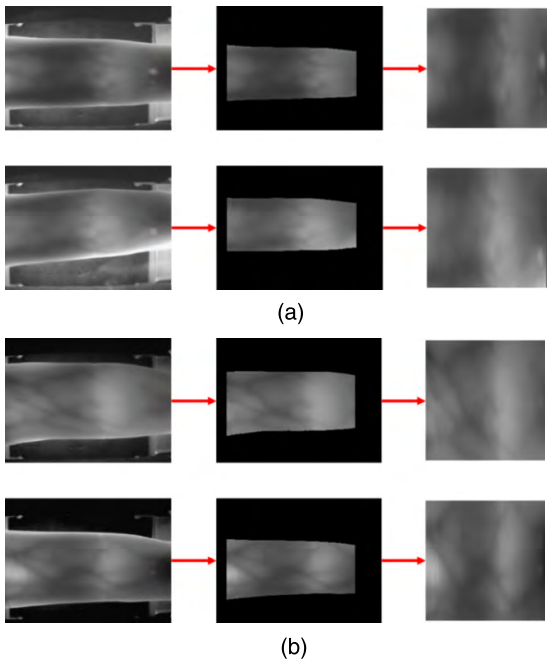

FIGURE 19. Examples of incorrect recognition cases. (a) False rejection case. (b) False acceptance case. Upper and lower images in (a) and (b) show the enrolled and input images, respectively. In (a) and (b), the images are an original image, ROI and a resized image (224 × 224) from the left to the right.

where μ_G and μ_I presents the means of the GMD and IMD, respectively. σ_G and σ_I denote the standard deviations of the GMD and IMD, respectively. In general, the farther the

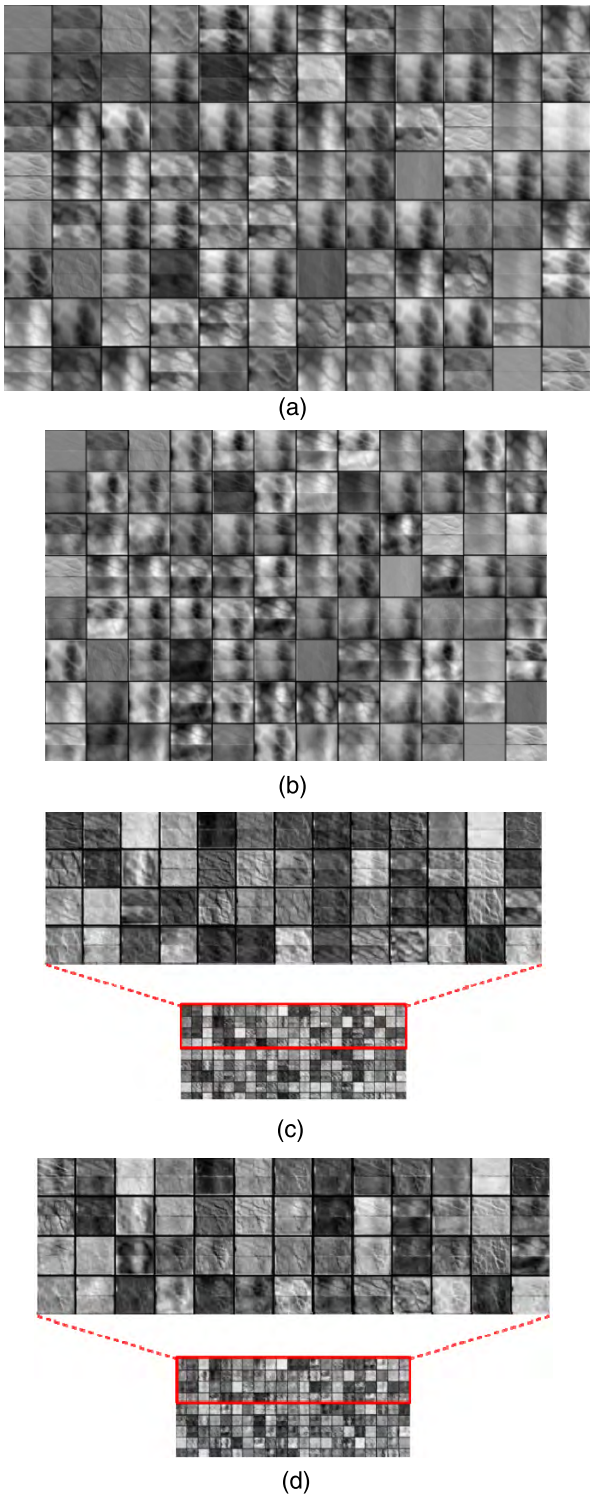

FIGURE 20. Examples of feature maps extracted from an authentic matching image ((a), (c), (e), (g), (i), and (j) and an imposter matching image ((b), (d), (f), (h), (k), and (l) from each layer of the DenseNet. (a, b) Feature map extracted from the 7 × 7 conv of Table 2, (c, d) output feature map of the 1 × 1 conv on the transition layer (1) of Table 2, (e, f) output feature map of the 1 × 1 conv on the transition layer (2) of Table 2, (g, h) output feature map of the 1 × 1 conv on the transition layer (3) of Table 2, (i, k) output feature map of the dense block (4) of Table 2, (j, l) 3D feature map image based on the average feature map values of (i, k), respectively.

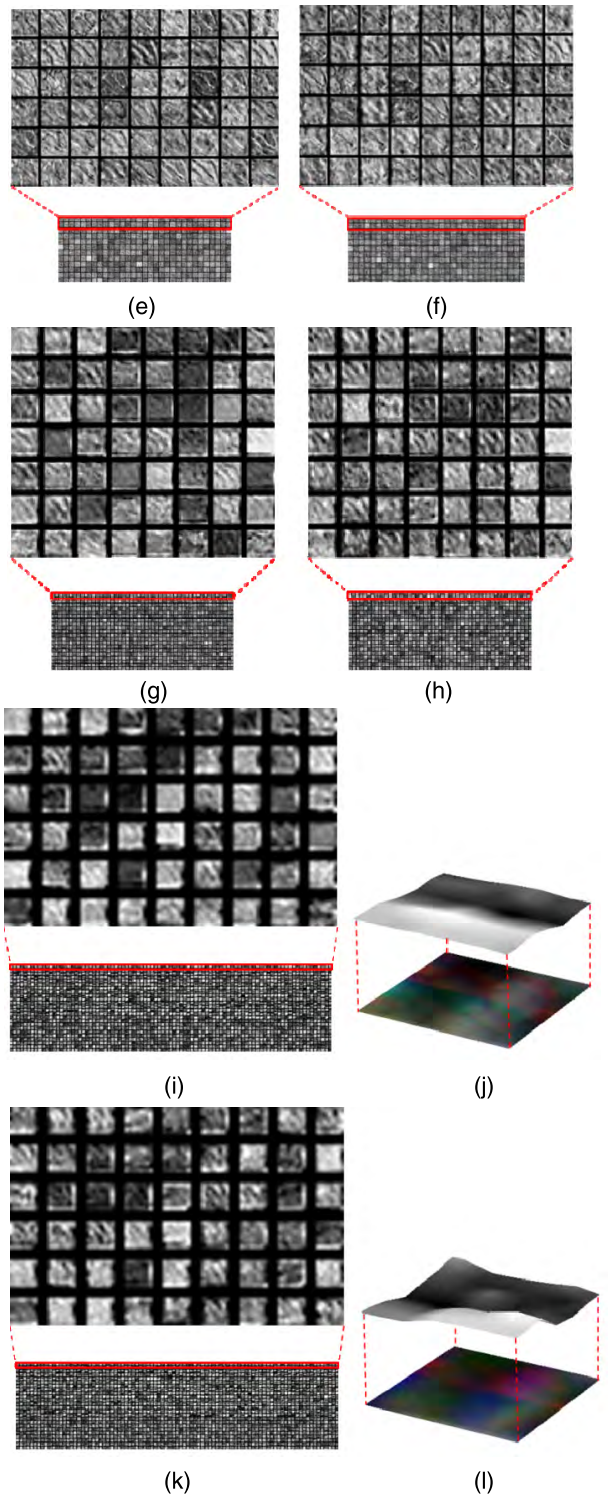

FIGURE 20. *(Continued.)* **Examples of feature maps extracted from an authentic matching image ((a), (c), (e), (g), (i), and (j) and an imposter matching image ((b), (d), (f), (h), (k), and (l) from each layer of the DenseNet. (a, b) Feature map extracted from the 7 × 7 conv of Table 2, (c, d) output feature map of the 1 × 1 conv on the transition layer (1) of Table 2, (e, f) output feature map of the 1 × 1 conv on the transition layer (2) of Table 2, (g, h) output feature map of the 1 × 1 conv on the transition layer (3) of Table 2, (i, k) output feature map of the dense block (4) of Table 2, (j, l) 3D feature map image based on the average feature map values of (i, k), respectively.**

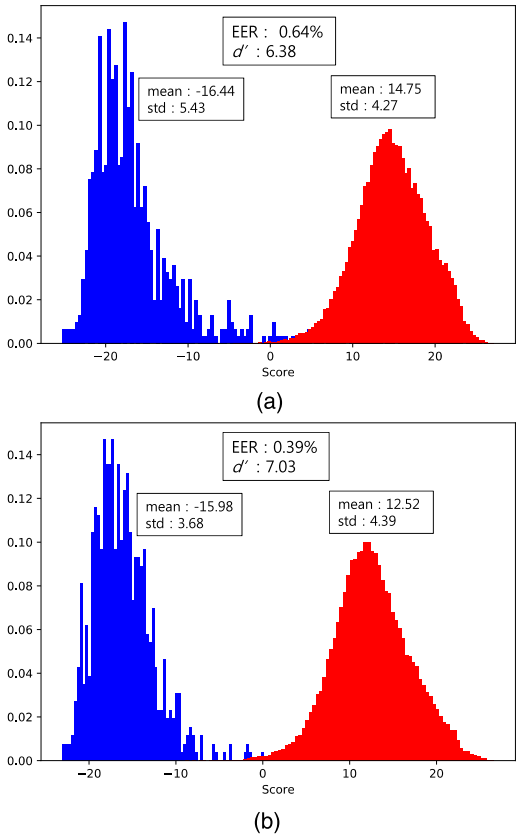

FIGURE 21. **Comparisons on the changes in genuine matching distribution (GMD) and imposter matching distribution (IMD) (a) without shift matching, and (b) with shift matching.**

two distributions are separated, the smaller the recognition error becomes. The d' value of Equation (6) increases when the two distributions separate, whereas it decreases when the overlap caused by the proximity between the two distributions increases. Therefore, a greater d' value is an indicator of a better accuracy of the biometric system subject to evaluation [55].

As shown in Figures 21 (a) and (b), the d' value with shift matching (7.03) of Figure 21 (b) is larger than that without shift matching (6.38) of Figure 21 (a), which means that the GMD and IMD with shift matching are more separable and consequent recognition error is smaller.

In order to obtain the optimal parameter for shift matching, we performed the experiments with training data to measure the change of EER according to the amount of pixels and direction for shift matching. From these experiments, we determined the optimal parameters of 5 pixels and 8 directions for shift matching (as shown in Figure 8) because we obtained the lowest EER of finger-vein recognition with these parameters.

VI. CONCLUSION

In this research, a deep DenseNet based finger-vein recognition method was proposed. A shift matching method was

also used to compensate for performance degradation caused by misalignment of the enrolled and input images, and a 3-channel composite image was used as the input to the CNN in place of a difference image which is susceptible to noise. When the composite images proposed in the study were used as input to the CNN, the recognition accuracy was confirmed to be higher than that with difference images. In addition, when tested with a noisy image, the composite image was observed to have greater robustness against noise compared to the difference image. Also, two open databases were used to test various CNN models, and the highest recognition accuracy was observed when the DenseNet-161 model was fine-tuned and the shift matching method was used for recognition. According to the results of this study, in most cases of false rejection, vein patterns were not clearly captured with shading, and a high degree of misalignment occurred. In false acceptance cases, vein patterns were partially captured and there were issues of similarity in the patterns and shading. The processing speed was measured on a desktop computer and an embedded system, which confirmed the applicability of our method in various environments.

In future studies, a new method would be examined for improving the processing speed while maintaining recognition accuracy by reducing the number of layers and transition layers of the DenseNet. In addition, the deep CNN model and shift matching would be applied to other types of vein images (palm-vein, hand-vein) as well as finger-print and palm-print recognition.

REFERENCES

[1] Z. Liu, Y. Yin, H. Wang, S. Song, and Q. Li, "Finger vein recognition with manifold learning," *J. Netw. Comput. Appl.*, vol. 33, pp. 275–282, May 2010.

[2] H. G. Hong, M. B. Lee, and K. R. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, p. 1297, 2017.

[3] W. Kim, J. M. Song, and K. R. Park, "Multimodal biometric recognition based on convolutional neural network by the fusion of finger-vein and finger shape using near-infrared (NIR) camera sensor," *Sensors*, vol. 18, no. 7, p. 2296, 2018.

[4] J. Yang, J. Yang, and Y. Shi, "Finger-vein segmentation based on multi-channel even-symmetric Gabor filters," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, Shanghai, China, Nov. 2009, pp. 500–503.

[5] J. Yang, Y. Shi, and J. Yang, "Personal identification based on finger-vein features," *Comput. Hum. Behav.*, vol. 27, pp. 1565–1570, Sep. 2011.

[6] W. Kejun, L. Jingyu, P. P. Oluwatoyin, and F. Weixing, "Finger vein identification based on 2-D Gabor filter," in *Proc. 2nd Int. Conf. Ind. Mechatronics Automat.*, Wuhan, China, May 2010, pp. 10–13.

[7] J. Yang, Y. Shi, J. Yang, and L. Jiang, "A novel finger-vein recognition method with feature combination," in *Proc. IEEE Int. Conf. Image Process.*, Cairo, Egypt, Nov. 2009, pp. 2709–2712.

[8] J. Peng, N. Wang, A. A. A. El-Latif, Q. Li, and X. Niu, "Finger-vein verification using Gabor filter and sift feature matching," in *Proc. 8th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Piraeus, Greece, Jul. 2012, pp. 45–48.

[9] J. Yang and J. Yang, "Multi-channel Gabor filter design for finger-vein image enhancement," in *Proc. Int. Conf. Image Graph.*, Xi'an, China, Sep. 2009, pp. 87–91.

[10] K. Y. Shin, Y. H. Park, D. T. Nguyen, and K. R. Park, "Finger-vein image enhancement using a fuzzy-based fusion method with Gabor and retinex filtering," *Sensors*, vol. 14, no. 2, pp. 3095–3129, 2014.

[11] Y. H. Park and K. R. Park, "Image quality enhancement using the direction and thickness of vein lines for finger-vein recognition," *Int. J. Adv. Robot. Syst.*, vol. 9, pp. 1–10, May 2012.

[12] J. Zhang and J. Yang, "Finger-vein image enhancement based on combination of gray-level grouping and circular Gabor filter," in *Proc. Int. Conf. Inf. Eng. Comput. Sci.*, Wuhan, China, Dec. 2009, pp. 1–4.

[13] W. Pi, J. Shin, and D. Park, "An effective quality improvement approach for low quality finger vein image," in *Proc. Int. Conf. Electron. Inf. Eng.*, Kyoto, Japan, Aug. 2010, pp. 424–427.

[14] C.-B. Yu, D.-M. Zhang, H.-B. Li, and F.-F. Zhang, "Finger-vein image enhancement based on muti-threshold fuzzy algorithm," in *Proc. Int. Congr. Image Signal Process.*, Tianjin, China, Oct. 2009, pp. 1–3.

[15] X. Qian, S. Guo, X. Li, F. Zhong, and X. Shao, "Finger-vein recognition based on the score level moment invariants fusion," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Wuhan, China, Dec. 2009, pp. 1–4.

[16] E. C. Lee, H. Jung, and D. Kim, "New finger biometric method using near infrared imaging," *Sensors*, vol. 11, no. 3, pp. 2319–2333, 2011.

[17] G. Yang, X. Xi, and Y. Yin, "Finger vein recognition based on a personalized best bit map," *Sensors*, vol. 12, no. 2, pp. 1738–1757, 2012.

[18] B. A. Rosdi, C. W. Shing, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11357–11371, 2011.

[19] Y. Lu, S. Yoon, S. J. Xie, J. Yang, Z. Wang, and D. S. Park, "Finger vein recognition using generalized local line binary pattern," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 5, pp. 1766–1784, 2014.

[20] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Mach. Vis. Appl.*, vol. 15, pp. 194–203, Oct. 2004.

[21] S. R. Cho, Y. H. Park, G. P. Nam, K. Y. Shin, H. C. Lee, K. R. Park, S. M. Kim, and H. C. Kim, "Enhancement of finger-vein image by vein line tracking and adaptive Gabor filtering for finger-vein recognition," *Appl. Mech. Mater.*, vol. 145, pp. 219–223, Dec. 2012.

[22] R. Raghavendra and C. Busch, "Presentation attack detection algorithms for finger vein biometrics: A comprehensive study," in *Proc. 11th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, Bangkok, Thailand, Nov. 2015, pp. 628–632.

[23] J.-D. Wu and C.-T. Liu, "Finger-vein pattern identification using SVM and neural network technique," *Expert Syst. Appl.*, vol. 38, pp. 14284–14289, Oct. 2011.

[24] H. Qin, L. Qin, L. Xue, X. He, C. Yu, and X. Liang, "Finger-vein verification based on multi-features fusion," *Sensors*, vol. 13, no. 11, pp. 15048–15067, 2013.

[25] S. Khellat-Kihel, R. Abrishambaf, N. Cardoso, J. Monteiro, and M. Benyettou, "Finger vein recognition using Gabor filter and support vector machine," in *Proc. IEEE Int. Image Process., Appl. Syst. Conf.*, Hammamet, Tunisia, Nov. 2014, pp. 1–6.

[26] S. A. Radzi, M. Khalil-Hani, and R. Bakhteri, "Finger-vein biometric identification using convolutional neural network," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1863–1878, 2016.

[27] K. S. Itqan, A. R. Syafeeza, F. G. Gong, N. Mustafa, Y. C. Wong, and M. M. Ibrahim, "User identification system based on finger-vein patterns using convolutional neural network," *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 5, pp. 3316–3319, 2016.

[28] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708.

[29] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Cancun, Mexico, Dec. 2016, pp. 378–383.

[30] *SDUMLA-HMT Finger Vein Database*. Accessed: May 7, 2018. [Online]. Available: http://mla.sdu.edu.cn/info/1006/1195.htm

[31] *Dongguk DenseNet-Based Finger-Vein Recognition Model (DDFRM) With Algorithms*. Accessed: Nov. 8, 2018. [Online]. Available: http://dm.dgu.edu/link.html

[32] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.

[33] A. Kumar and D. Zhang, "Personal recognition using hand shape and texture," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2454–2461, Aug. 2006.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–14.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun./Jul. 2016, pp. 770–778.
[38] *NVIDIA GeForce GTX 1070*. Accessed: Nov. 27, 2018. [Online]. Available: https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1070-ti/
[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 675–678.
[40] *Visual Studio 2013*. Accessed: Jun. 5, 2018. [Online]. Available: https://www.microsoft.com/en-us/search/result.aspx?q=visual+studio+2013
[41] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, Paris, France, Aug. 2010, pp. 177–186.
[42] P. Gupta and P. Gupta, "An accurate finger vein based verification system," *Digit. Signal Process.*, vol. 38, pp. 43–52, Mar. 2015.
[43] L. Dong, G. Yang, Y. Yin, F. Liu, and X. Xi, "Finger vein verification based on a personalized best patches map," in *Proc. IEEE Int. Joint Conf. Biometrics*, Clearwater, FL, USA, Sep./Oct. 2014, pp. 1–8.
[44] F. Liu, G. Yang, Y. Yin, and S. Wang, "Singular value decomposition based minutiae matching method for finger vein recognition," *Neurocomputing*, vol. 145, pp. 75–89, Dec. 2014.
[45] F. Liu, Y. Yin, G. Yang, L. Dong, and X. Xi, "Finger vein recognition with superpixel-based features," in *Proc. IEEE Int. Joint Conf. Biometrics*, Clearwater, FL, USA, Sep./Oct. 2014, pp. 1–8.
[46] X. Xi, L. Yang, and Y. Yin, "Learning discriminative binary codes for finger vein recognition," *Pattern Recognit.*, vol. 66, pp. 26–33, Jun. 2017.
[47] H. Liu, L. Yang, G. Yang, and Y. Yin, "Discriminative binary descriptor for finger vein recognition," *IEEE Access*, vol. 6, pp. 5795–5804, 2018.
[48] *Jetson TX2 Module*. Accessed: Feb. 24, 2019. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/
[49] *Keras: The Python Deep Learning Library*. Accessed: Oct. 1, 2018. [Online]. Available: https://keras.io/
[50] *Ubuntu 16.04*. Accessed: Nov. 19, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Ubuntu_version_history#%20Ubuntu_16.04_LTS_.28Xenial_Xerus.29
[51] T. D. Pham, Y. H. Park, D. T. Nguyen, S. Y. Kwon, and K. R. Park, "Nonintrusive finger-vein recognition system using NIR image sensor and accuracy analyses according to various factors," *Sensors*, vol. 15, no. 7, pp. 16866–16894, 2015.
[52] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," in *Proc. IAPR Conf. Mach. Vis. Appl.*, Tsukuba, Japan, May 2005, pp. 347–350.
[53] J. Daugman, "How iris recognition works," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 21–30, Jan. 2004.
[54] *Noisy Iris Challege Evaluation—Part II*. Accessed: Mar. 11, 2019. [Online]. Available: nice2.di.ubi.pt
[55] C. Tan and A. Kumar, "Efficient and accurate at-a-distance iris recognition using geometric key-based iris encoding," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 9, pp. 1518–1526, Sep. 2014.
[56] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.
[57] Y. Fang, Q. Wu, and W. Kang, "A novel finger vein verification system based on two-stream convolutional network learning," *Neurocomputing*, vol. 290, pp. 100–107, May 2018.
[58] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," *Pattern Recognit. Lett.*, vol. 119, pp. 148–156, Mar. 2019.
[59] L. Yang, G. Yang, Y. Yin, and X. Xi, "Finger vein recognition with anatomy structure analysis," *IEEE Trans. Circuits Syst. Video. Technol.*, vol. 28, no. 8, pp. 1892–1905, Aug. 2018.
[60] W. Kang and Q. Wu, "Contactless palm vein recognition using a mutual foreground-based local binary pattern," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 11, pp. 1974–1985, Nov. 2014.

JONG MIN SONG received the B.S. degree in software engineering from Chungbuk National University, Cheongju, South Korea, in 2017. He is currently pursuing the M.S. degree in electronics and electrical engineering from Dongguk University, Seoul. He designed the finger-vein recognition system based on DenseNet and wrote the draft of original paper. His research interests include biometrics and pattern recognition.

WAN KIM received the B.S. degree in bio medical engineering from Gachon University, Seoul, South Korea, in 2016. He is currently pursuing the M.S degree in electronics and electrical engineering with Dongguk University. He helped to implement the preprocessing algorithm and perform the experiments. His research interests include biometrics and pattern recognition.

KANG RYOUNG PARK received the B.S. and M.S. degrees in electronic engineering from Yonsei University, Seoul, South Korea, in 1994 and 1996, respectively, and the Ph.D. degree in electrical and computer engineering from Yonsei University, in 2000. He has been a Professor with the Division of Electronics and Electrical Engineering, Dongguk University, since 2013. His research interests include image processing and biometrics. He supervised this research and helped the revision of draft of original paper.