Finger-Vein Recognition Based on Densely Connected Convolutional Network Using Score-Level Fusion With Shape and Texture Images

KYOUNG JUN NOH, JIHO CHOI, JIN SEONG HONG, AND KANG RYOUNG PARK, (Member, IEEE)
Division of Electronics and Electrical Engineering, Dongguk University, Seoul 04620, South Korea
Corresponding author: Kang Ryoung Park (parkgr@dongguk.edu)

ABSTRACT Biometrics using finger-veins is a recognition method based on the shape of veins in fingers, and it has the advantage of difficulty to be forged. However, a shade is inevitably produced due to the bones and fingernails, and a change in illumination occurs when acquiring the images of finger-veins. Previous studies have conducted finger-vein recognition using a single-type texture image or finger-vein segmented image (shape image). A texture image provides numerous features, but it is vulnerable to the changes in illumination during recognition and contains noises in regions other than the finger-vein region. A shape image is less affected by noises; however, the recognition accuracy is significantly reduced due to fewer features available and mis-segmented regions caused by shades. In this study, therefore, rough finger-vein regions in an image are detected to reduce the effect of mis-segmented regions, to complement the drawbacks of shape image-based finger-vein recognition. Furthermore, score-level fusion is performed for two output scores of deep convolutional neural network extracted from the texture and shape images, which can reduce the sensitivity to noise, while diverse features provided in the texture image are used efficiently. Two open databases, the Shandong University homologous multi-modal traits finger-vein database and Hong Kong Polytech University finger image database, are used for experiments, and the proposed method shows better recognition performance than the state-of-the-art method.

INDEX TERMS Finger-vein recognition, shape and texture images of finger-vein, deep CNN, score-level fusion.

I. INTRODUCTION
There are various biometric technologies currently available, such as iris, fingerprint, face, voice, and finger-vein. Finger-vein includes deoxygenated hemoglobin which absorbs more amount of near infrared (NIR) light than other areas of tissues and skins. Therefore, finger-vein patterns are observed as darker areas than others in the captured image by NIR illuminator and camera. Different from artery, the vein including finger-vein has the non-uniform shape due to the weak blood pressure, which shows the unique characteristics for human identification [3]. Finger-veins are easy to be acquired, and difficult to be forged because of existence inside skin.

However, the clear observation of finger-vein patterns is affected by shades and changes in illumination, which causes the degradation of recognition accuracy.

The existing handcrafted feature-based methods for finger-vein recognition, such as local binary pattern (LBP) LBP [8], [9], [17], [19], [20], [22], [23] and local directional code (LDC) [16], have reduced recognition accuracy if noise caused by shades and changes in illumination in images are not properly corrected during preprocessing or finger-vein lines are not enhanced by applying optimal filters [4],
extraction is performed by distinctly expressing finger-vein
ods - texture-based finger-vein recognition, in which feature
and feature extraction. Particularly in this section, two meth-
most of the previous studies have focused on preprocessing
resizing are performed. With respect to different processes,
lar enhancement, detection of region-of-interest, and image
which are the same as those adopted by the finger-vein recog-
processing, feature extraction, and matching processes,
Section 5 explains the results of the experiment and analyses,
tions. Section 4 introduces the details of proposed method.
recognition are explained. Section 3 describes our contribu-
As a composite image in various channel images and score-
channel image. Therefore, the current challenges for finger-
vein region of the shape image, which was binary-
image, by restoring the pixel value that existed within the
information loss that occurred when acquiring the shape
performed matching based on the blanket dimension dis-
finger-vein image acquired using the blanket technique, and
minimized
Gaussian high-pass filter. Here, a robust finger-vein recogni-
vein image that was enhanced using a symmetrical modified
Gabor filter. Here, a robust finger-vein recognition system was promoted by providing additional information
during recognition, where the geometric information of
fingers was not removed. Feature extraction was conducted using an LBP, while matching was performed based on the
hamming distance in the final matching stage [8]. However,
LBP uses a square filter, which is not appropriate for a finger-
vein that has linear characteristics. Therefore, Rosdi et al.
proposed a local line binary pattern (LLBP) for efficient
feature extraction. Feature extraction was performed using
LLBP for the shape image extracted using a Gaussian high-
pass filter, and then, matching was performed based on the
hamming distance [9]. Liu et al. extracted lacunarity of a
finger-vein image acquired using the blanket technique, and
performed matching based on the blanket dimension dis-
tance and lacunarity distance [10]. Gupta et al. minimized the
information loss that occurred when acquiring the shape
image, by restoring the pixel value that existed within the
finger-vein region of the shape image, which was binary-
thresholded through the fusion of the finger-vein image that
[6], [11]; moreover, the optimal filters need to be manually
calculated to be customized for each database.
For such reasons, finger-vein recognition using a deep
network such as convolutional neural network (CNN), which
is not greatly affected by preprocessing, has been extensively
investigated. The existing CNN-based finger-vein recogni-
tion systems use a network input image as a difference
image or find the distance between feature vectors [14], [27],
[28]. A difference image is created by finding the difference
in pixel values between the enrolled image and input image,
where a concordant or discordant output can be obtained
from one input and all layers of the trained network are used
during the recognition. However, it can become sensitive
to noise, because of reduction in the change in feature values
of an image [27], [28]. Moreover, calculating the distance
between the feature vectors uses the original image as an
input image. Therefore, not all layers of the trained network
can be used, because feature vectors are extracted from the
layer before the final layer of the network in each image.
However, the accuracy by this scheme is lesser than that of
the system that uses the difference image as the input [27].
In a previous study on finger-vein recognition, a composite
image generated using only a texture image encompassing
all veins and background was used as input in the CNN [2].
A composite image is generated by the 3 channel-image of
enrolled and recognized ones. However, this method has
the drawback of a proportionally increased amount of noise as
the noise in initially generated images is carried onto each
channel image. Therefore, the current challenges for finger-
vein recognition are making the system robust to the noise
caused by shades and changes in illumination, and utilize
more information than either finger shape or finger texture. To
consider these challenges, we propose a CNN-based finger-
vein recognition method using both texture and shape images
as a composite image in various channel images and score-
level fusion.
In Section 2, analyses of existing studies on finger-vein
recognition are explained. Section 3 describes our contribu-
tions. Section 4 introduces the details of proposed method.
Section 5 explains the results of the experiment and analyses,
and finally, Section 6 concludes the paper.

II. RELATED WORK
Typical biometric systems comprise image acquisition, pre-
processing, feature extraction, and matching processes,
which are the same as those adopted by the finger-vein recogni-
tion system. During preprocessing, image alignment, image
enhancement, detection of region-of-interest, and image resizing are performed. With respect to different processes,
most of the previous studies have focused on preprocessing
and feature extraction. Particularly in this section, two meth-
ods - texture-based finger-vein recognition, in which feature
extraction is performed by distinctly expressing finger-vein
patterns during image enhancement, and shape-based finger-
vein recognition, in which feature extraction is performed
using the finger-vein shape image obtained by segmenting the
finger-vein region - are distinguished for analyses.

A. SHAPE-BASED FINGER-VEIN RECOGNITION
During finger-vein image acquisition, some information that
is unnecessary besides finger-vein may be included due to
diffused reflection of light caused by tissues other than finger-
vein or illumination imbalance caused by an NIR camera
sensor. Such information is a crucial factor that generates an
error case during finger-vein recognition. Hence, a shape-
based recognition method has been researched to remove
unnecessary information by segmenting only the finger-vein
region, which absorbs the NIR light by using deoxyhe-
moglobin within the vein, thus having a smaller pixel value
and lesser pixel variation in the vein region as compared to the
adjacent regions. Finger-vein shape images are acquired based
on these characteristics.

Yang et al. performed finger-vein recognition by conduct-
ing finger-vein segmentation and feature extraction using
the Gabor filter bank based on various scales and chan-
els [4]. Guan et al. obtained the shape image by segmenting
the finger-vein region based on local dynamic thresholding.
The existing linear discriminant analysis (LDA) is applied
with unidirectional substances in either horizontal or vertical
direction, thus failing to properly extract finger-vein informa-
tion; therefore, two-dimensional weighted LDA (2DLDA)
was proposed and dimension reduction and feature extraction
were performed accordingly [5]. Peng et al. found an optimal
Gabor filter parameter that extracts features most accurately
and conducted finger-vein recognition using the shape image
produced accordingly [6]. Yang et al. extracted the finger-
vein region based on the vein valley characteristic analy-
sis. Finger-vein recognition was performed using the finally
acquired shape image, based on template matching [7].
Lee et al. acquired a shape image by segmenting the finger-
vein image that was enhanced using a symmetrical modified
Gaussian high-pass filter. Here, a robust finger-vein recogni-
tion system was promoted by providing additional informa-
tion during recognition, where the geometric information of
fingers was not removed. Feature extraction was conducted using an LBP, while matching was performed based on the
hamming distance in the final matching stage [8]. However,
LBP uses a square filter, which is not appropriate for a finger-
vein that has linear characteristics. Therefore, Rosdi et al.
proposed a local line binary pattern (LLBP) for efficient
feature extraction. Feature extraction was performed using
LLBP for the shape image extracted using a Gaussian high-
pass filter, and then, matching was performed based on the
hamming distance [9]. Liu et al. extracted lacunarity of a
finger-vein image acquired using the blanket technique, and
performed matching based on the blanket dimension dis-
tance and lacunarity distance [10]. Gupta et al. minimized the
information loss that occurred when acquiring the shape
image, by restoring the pixel value that existed within the
finger-vein region of the shape image, which was binary-
thresholded through the fusion of the finger-vein image that
was enhanced by a multiscale matched Gaussian filter and the shape image generated by the line tracking algorithm [11]. Hoshyar et al. proposed a finger-vein recognition system that extracts a maximum curvature point for acquiring a shape image, and then performs classification using multilayer perceptron (MLP) [12]. Wang et al. proposed local binary pattern variance (LBPV) considering that LBP does not have rotation invariance. A shape image was acquired by applying the Gaussian matched filter to the acquired image, and then, feature extraction was performed using LBPV. The final classification was performed using a support vector machine (SVM) [13]. Radzi et al. acquired a rough shape image by applying local dynamic thresholding to the previously acquired finger-vein image, and then performed feature extraction and classification through CNN without an additional feature extraction stage [14]. Veluchamy et al. acquired the shape images for finger-vein and finger-knuckle by using a repeated line tracking (RPL) algorithm, and then performed feature-level fusion based on fractional firefly optimization of each shape image. The final classification was performed using a multi-layered k-SVM [15].

B. TEXTURE-BASED FINGER-VEIN RECOGNITION

When finger-vein recognition is performed based on a shape image, the finger-vein region is segmented by various algorithms in which a critical error case may be generated if an incorrectly extracted region exists, depending on the characteristics of the shape image. Moreover, important information, such as pixel variation or hidden directional information, within the finger-vein region is removed by binary thresholding. Due to such drawbacks, a finger-vein recognition that does not perform finger-vein region segmentation on the acquired finger-vein images has been extensively studied. Meng et al. reported that the loss of directional information that occurs in a shape image significantly reduces the recognition accuracy. Therefore, a finger-vein recognition method based on the texture image and a newly suggested LDC is proposed. LDC, which is based on a Weber local descriptor, focuses on the gradient orientation information [16]. Yang et al. proposed a personalized best bit map that matches only the important bits in the existing LBP code [17]. Harsha et al. suggested a matching scheme based on the wavelet transform distance and energy feature distance for which a modified Haar energy feature was extracted using the sequential Haar wavelet [18]. Yang et al. proposed a finger-vein recognition system for which personalized weight maps (PWMs) were added to the matching using an LBP code. A more robust system in which varying weights were assigned to each image using PWMs was suggested [19]. Lu et al. proposed an local line binary pattern (LLBP) with a line shape and then, used a poly-directional filter for accurately extracting the directional information [20]. Subsequently, a system for matching based on the concatenated histogram by calculating competitive Gabor magnitude and competitive Gabor orientation using the histogram of competitive Gabor response (HCGR) was suggested [21]. Furthermore, moving beyond the LLBP, which uses a poly-directional filter, a generalized local line binary pattern with revised filter diameter was suggested. A system for performing score-level fusion for the result of a directional filter with the best matching performance was proposed [22]. Liu et al. proposed a customized local line binary pattern, which is a class-based orientation-selectable poly-directional local line binary pattern [23]. Wu et al. suggested a finger-vein recognition system in which dimension reduction and feature extraction are performed using the principal component analysis (PCA), and then classification is performed using an adaptive neuro-fuzzy inference system (ANFIS) [24]. Subsequently, a finger-vein recognition system that performs dimension reduction and feature extraction using the PCA and LDA simultaneously, and then, final classification using SVM was suggested [25]. Khellat-kihel et al. proposed a finger-vein recognition system with classification method using an SVM in which images are enhanced with a 2D Gabor filter [26]. Hong et al. extracted features from a visual geometry group (VGG)-16 network for which the difference image that shows the pixel value difference between the enrolled and matched images is used as an input, and then performed matching using Euclidian distance [27]. Kim et al. suggested a multimodal biometric recognition system using each of the difference images of the finger-vein and finger-shape [28]. Song et al. used the DenseNet-161 network for feature extraction and classification, and complemented the drawbacks of the difference image, which is vulnerable to noise and misalignment, by using the composite image of the original image instead of the difference image. Furthermore, a finger-vein recognition system having a more robust environment for the misalignment problem was proposed based on a shift-matching technique [2]. However, the existing finger-vein images suffer from poor quality, where error cases still occur when the noise in the original image increases in multiplicative fashion in images of each channel during the production of a composite image; thereby, recognition is proceeded due to unnecessary information instead of the intended finger-vein patterns.

For overcoming the abovementioned disadvantages, this paper proposes a finger-vein recognition system that fully utilizes the advantages of each dataset, where the output scores of two CNNs with composite images as an input based on each image go through score-level fusion for simultaneously using the considerable information of the texture image and the finger-vein pattern-focused information of the shape image.

Table 1 compares the previous and the proposed methods for finger-vein recognition.

III. CONTRIBUTIONS

The contributions of our study are as follows.

- It is the first study that considers both texture and shape images simultaneously for finger-vein recognition.

- To complement the drawbacks of repeated line tracking algorithm, which has been widely used in finger-vein segmentation, we propose a finger-vein segmentation based on
contrast-limited adaptive histogram equalization (CLAHE), morphological operation, and component labeling.

- Existing texture-based finger-vein recognition system has the drawback of being sensitive to noise. To overcome this drawback and simultaneously use the ample information provided by the texture image as well as the compact information by the shape image, we suggest the score-level fusion of two CNNs with the composite images of shape and texture images.

- We made the shape and texture-based finger-vein recognition models trained with our experimental databases available to other researchers through [29] for fair comparisons on the performance.

IV. PROPOSED METHOD
A. OVERVIEW OF PROPOSED METHOD
The overall flowchart of the finger-vein recognition system proposed in this study is shown in Figure 1. First, finger images acquired by an NIR camera and illuminator are preprocessed. The preprocessing includes the steps of binary thresholding, in-plane rotation compensation, and detection of finger region-of-interest (ROI). A $4 \times 20$ mask is used to fill if the finger-vein pattern is not detected due to noise when acquiring an ROI image or a collapsed region is generated in the acquired ROI image due to low quality of original image. Then, any unnecessary region due to noise is removed through component labeling. Then, the final ROI image is stretched to a size of $224 \times 224$ pixels through bilinear interpolation.

In Figure 1, the left-hand side shows the matching process using a texture image, while the right-hand side shows that using a shape image, which is generated through a repeated line tracking algorithm [3] (Step 6). Finally, both texture and shape ROI images are generated as composite images (Steps 3 and 7), to be used as an input for the CNN (Steps 4 and 8), and then, matching is performed using the obtained output matching score.

During the matching, 8-way shift matching is performed considering the misalignment between the enrolled and input images (Steps 5 and 9). Specifically, the minimum matching distance obtained from the enrolled image shifted in each direction is selected as the final matching score. By performing score-level fusion for these scores (Step 10), the final finger-vein recognition is conducted (Step 11).

B. PREPROCESSING
1) FINGER REGION DETECTION AND IN-PLANE ROTATION COMPENSATION
By binarizing the image to remove the background region in the captured image, the image shown in Figure 2(b) is obtained. The background near the finger region is not completely removed; thus, the Sobel edge detector and area threshold method are used to remove the background [30]. First, an edge map is created using the Sobel edge detector and a difference image is created based on the binarized image. Then, the area threshold method is applied to obtain a mask image with the background removed, as shown in Figure 2(c). In general, in-plane rotation causes a serious error in finger-vein recognition, and hence, mismatch caused by rotation is reduced through in-plane rotation compensation in this study [32]. Second-order moments, as shown in Equation (1), are calculated for the binarized mask obtained from Figure 2(c) to perform in-plane rotation compensation.

\[
\gamma_{11} = \frac{\sum_{(x,y) \in M} (y - m_y)^2 I(x, y)}{\sum_{(x,y) \in M} I(x, y)},
\]

\[
\gamma_{12} = \frac{\sum_{(x,y) \in M} (x - m_x)(y - m_y) I(x, y)}{\sum_{(x,y) \in M} I(x, y)}
\]
TABLE 1. Comparisons of the previous and proposed methods for finger-vein recognition.

| Categories                      | Method                                      | Advantage                                                                 | Disadvantage                                      |
|---------------------------------|---------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------|
| Non-training-based              | Gabor filter [4, 6]                         | Higher accuracy of finger-vein recognition                                | Recognition performance can be dependent on misalignment, image quality, and shading |
|                                 | (2D)2 LDA [5]                               |                                                                           |                                                   |
|                                 | Variant of LBP [8, 9]                       | Fast as it can be embedded in real-time devices                           |                                                   |
|                                 | Valley characteristic analysis [7]          |                                                                           |                                                   |
|                                 | Blanket technique [10]                      |                                                                           |                                                   |
|                                 | Gaussian filter [11]                        | Keeps the information of inside the vein region                           |                                                   |
| Training-based                  | MLP [12]                                    | Robust in various environments by training various data, including poor quality | Need additional procedure for feature extraction   |
|                                 | SVM [13, 15]                                |                                                                           |                                                   |
|                                 | Local dynamic thresholding and CNN [14]     | Does not need additional procedure for extracting features               |                                                   |
| Non-training-based              | Sequential Haar wavelet [18]                | Low computational complexity                                              | Recognition performance can be dependent on misalignment, image quality, and shading |
|                                 | LDC [16]                                    |                                                                           |                                                   |
|                                 | Variant of LBP [17, 19, 20, 22, 23]         | Fast processing speed because of using only local region of data         |                                                   |
|                                 | HCGR [21]                                   | Feature dimension is much lower than that in LBP, which makes processing much faster |                                                   |
| Texture-based                   | ANFIS [24], SVM [25, 26]                    | Robust on various environment                                              | Need additional procedure for feature extraction   |
|                                 | CNN with difference image [27, 28]          | Do not need additional procedure for feature extraction                  | Very sensitive to noise and misalignment          |
|                                 | CNN with composite image [2]                | Robust on misalignment problem                                            | Noise can be magnified in multiple channel images of composite image |
| Shape & texture-based           | Score-level fusion of two CNNs with the composite images of shape and texture images (proposed method) | Very robust to noise and shows good performance even though image quality is poor | Long processing time                               |

\[
\gamma_{22} = \frac{\sum_{(x,y) \in M} (x - m_x)^2 I(x, y)}{\sum_{(x,y) \in M} I(x, y)} \tag{1}
\]

\[
I(x, y) \text{ and } (m_x, m_y) \text{ represent the pixel value and center coordinates on the } (x, y) \text{ coordinates of each image. In addition, } M \text{ represents the binary mask area. Based on the values obtained from Equation (1), rotation angle } \theta \text{ is calculated based on Equation (2) to perform in-plane rotation compensation [32]; Figures 2(d) and (e) show the resulting mask and the ROI image obtained thereby.}
\]

\[
\theta = \begin{cases} 
\arctan \left( \gamma_{11} - \gamma_{22} + \sqrt{(\gamma_{11} - \gamma_{22})^2 + 4\gamma_{12}^2} \right) & (\text{if } \gamma_{11} > \gamma_{22}) \\
\arctan \left( \frac{-2\gamma_{12}}{\gamma_{22} - \gamma_{11} + \sqrt{(\gamma_{22} - \gamma_{11})^2 + 4\gamma_{12}^2}} \right) & (\text{else}) 
\end{cases} \tag{2}
\]

2) EXTRACTION OF ROI IMAGE

An NIR camera sensor and illuminator are used to acquire the original finger-vein image, during which NIR light may not be penetrated well due to obstacles such as thick skin tissue or fingernails. Moreover, the acquired images may have background regions that need to be removed. Therefore, the ROI is reset for the ROI image obtained from Figure 2(e), so as to use the central region of the finger image through which NIR light is penetrated well [2]. Here, the cropped mask for which the left- and right-hand sides are cropped by a certain amount is obtained, and then, the finger region that is mis-segmented by the illumination imbalance is removed through component labeling. Finally, the overly eroded region is compensated so as to obtain the optimized ROI mask that expresses the finger-vein pattern. Here, a 4 × 20-type filter is used for compensation because the finger image is longer in the horizontal direction, which is optimal for revising the mask. By filling the pixel value of the original finger image for the generated mask region, the final ROI
image for finger-vein recognition is obtained as shown in Figure 2(f).

C. FINGER-VEIN REGION SEGMENTATION

A finger-vein recognition system that uses the shape image requires a very precise and accurate finger-vein segmentation. The complexity of the algorithm is increased in this case. Therefore, in the proposed method, only finger-vein shape information is used without considering the pixel variation within the finger-vein, and hence, the complexity of the algorithm is lower and a relatively rougher region is detected compared to the existing methods. However, there is a problem with the quality of the image or the unique noise substance generated in the existing finger-vein segmentation algorithm, and thus, a revised algorithm is proposed. The shape image is obtained based on the repeated line tracking algorithm [3]. The downside of the repeated line tracking technique is that it is vulnerable to noise and the accuracy of line tracking is reduced if the variation in the adjacent pixel is not noticeable. In addition, the finger-vein image extracted by the repeated line tracking algorithm has a unique noise substance (gaps & burs), and hence, the performance of the finger-vein recognition system that uses such an image cannot be expected to be high. Generally, the finger-vein image often incurs the abovementioned noise, along with a small pixel variation, and therefore, an additional enhancement process of improving the distinguishability between the finger-vein region and background region through CLAHE and gamma contrast [31] is performed in this study (Figure 3(b)). Each process increases the pixel difference within the image, thus improving the robustness of the repeated line tracking algorithm. Subsequently, after extracting the finger-vein line through the repeated line tracking algorithm (Figure 3(c)), binary thresholding is performed to acquire the binarized finger-vein image (Figure 3(d)), after which the mis-segmented region is removed during the postprocessing. Additional processing is required as the finger-vein recognition performance is reduced if unnecessary components or segmented regions are not smoothed in the shape image. Therefore, the shape of the finger-vein is roughly extracted through component labeling (Figure 3(e)) and a morphological operation (Figure 3(f)) [31]. The final shape image acquired after going through each process is focused on finding the rough regions. After primarily finding the finger-vein regions roughly, unclear regions are filled using a simple morphological operation, and hence, a rough but robust finger-vein extracting algorithm is used in this study.

A repeated line tracking algorithm for acquiring the shape image is conducted as follows [3]. In the acquired finger-vein ROI image, the vein line is drawn by moving the current point as the moving direction is determined with respect to the pixel value within the range of the mask for a random point \((x_c, y_c)\). For images having the same size as that of the original image and all pixel values of 0, the pixel value of each point is increased by 1. After iteration is repeated for a set number of times, an image with the finally extracted vein region is acquired. To obtain the direction to which the current point \((x_c, y_c)\) is moved, the \(p\) point needs to be found, as shown in Figure 4. The \(p\) point is determined according to the following equation.

\[
V = \max_{(x_c, y_c) \in Nc} \{ F(x_c + r \cos \theta - \frac{W}{2} \sin \theta, y_c + r \sin \theta + \frac{W}{2} \cos \theta) + F(x_c + r \cos \theta + \frac{W}{2} \sin \theta, y_c + r \sin \theta - \frac{W}{2} \cos \theta) - 2F(x_c + r \cos \theta, y_c + r \sin \theta) \} \\
\]

In Equation (3), \(F(x, y)\) represents the pixel intensity of \((x, y)\), while \(W\) is the distance between \(s\) and \(t\) in Figure 4. The point at which \(V\) is maximized according to each orientation within the mask region is the shifted point \(p\). When \(V_{max}\) is a positive value, the current point moves to the shifted point \(p\) and a new iteration begins when all \(V\) values are negative.

D. GENERATING COMPOSITE IMAGE

For the acquired texture and shape images, a composite image is generated for each of the selected enrolled image and input image. The enrolled and input images, each of which is resized to \(224 \times 224\) pixels through bilinear interpolation.
Figure 5 shows the DenseNet-161 architecture used for extracting the shape- and texture-based matching score. The first orange block is the input layer, and a composite image of 224 × 224 × 3 format is used as an input, as shown in Figures 6 and 7. A 7 × 7 sized convolution filter is additionally used to primarily extract the large-sized feature information, after which the feature map size is adjusted by max pooling. Each blue block is a dense block, and a bottleneck structure is formed using a 1 × 1-sized and a 3 × 3-sized convolution filter. Based on such structure, the dimension that becomes enlarged compared to the skip connection of the network is adjusted. Furthermore, a green block, which is the transition layer, is arranged between the dense blocks for adjusting the feature map size, in order to efficiently accumulate the dimensions. Each yellow arrow represents the skip connection. The feature map of a previous layer is concatenated to the current feature map, thereby enabling the feature extracted from the top layer to be efficiently brought to the lower layer. After the final dense block, the classification is completed by going through an 8 × 8-sized global average pooling layer, a fully connected layer (FCL), and a softmax layer. In this study, the output layer of the DenseNet-161 network, which used to be 1000 output, is revised so as to allow two outputs to be created for extracting two types of scores: genuine matching score and imposter matching score. Here, genuine matching refers to the case in which the enrolled image and input-recognized image have the same class data, while imposter matching refers to the case in which the enrolled image and input-recognized image have different class data.

Table 2 shows the details of the DenseNet-161 architecture used in this study. We use k as 48, referred to as the growth rate of DenseNet. According to the set growth rate, the number of channels of the input feature map is changed in each dense block. Finally, the number of channels of the output feature map in the dense block is also determined by the growth rate. The number of a base block of the dense block comprising 4k 1 × 1 conv and 1k 3 × 3 conv varies according to the place of the dense block. In the DenseNet-161 architecture used in this study, the 1st dense block has 6, the 2nd dense block has 12, the 3rd dense block has 36, and the 4th dense block has 24 base blocks.

2) SHIFT MATCHING AND SCORE-LEVEL FUSION FOR FINGER-VEIN RECOGNITION
Shift matching is performed to resolve the misalignment issue, which degrades the performance of a finger-vein recognition system [2]. During finger-vein recognition, even if the enrolled image and matched image are a genuine matching case, the misalignment issue inevitably occurs when trials for acquiring each image are repeated. Hence, shift matching is performed to resolve such issue. From the existing shift matching, after which score-level fusion is performed to express as a single score vector, thereby completing the final recognition.

1) DenseNet-161 ARCHITECTURE
Figure 7 shows the DenseNet-161 architecture used for extracting the shape- and texture-based matching score. The first orange block is the input layer, and a composite image of 224 × 224 × 3 format is used as an input, as shown in Figures 6 and 7. A 7 × 7 sized convolution filter is additionally used to primarily extract the large-sized feature information, after which the feature map size is adjusted by max pooling. Each blue block is a dense block, and a bottleneck structure is formed using a 1 × 1-sized and a 3 × 3-sized convolution filter. Based on such structure, the dimension that becomes enlarged compared to the skip connection of the network is adjusted. Furthermore, a green block, which is the transition layer, is arranged between the dense blocks for adjusting the feature map size, in order to efficiently accumulate the dimensions. Each yellow arrow represents the skip connection. The feature map of a previous layer is concatenated to the current feature map, thereby enabling the feature extracted from the top layer to be efficiently brought to the lower layer. After the final dense block, the classification is completed by going through an 8 × 8-sized global average pooling layer, a fully connected layer (FCL), and a softmax layer. In this study, the output layer of the DenseNet-161 network, which used to be 1000 output, is revised so as to allow two outputs to be created for extracting two types of scores: genuine matching score and imposter matching score. Here, genuine matching refers to the case in which the enrolled image and input-recognized image have the same class data, while imposter matching refers to the case in which the enrolled image and input-recognized image have different class data.

Table 2 shows the details of the DenseNet-161 architecture used in this study. We use k as 48, referred to as the growth rate of DenseNet. According to the set growth rate, the number of channels of the input feature map is changed in each dense block. Finally, the number of channels of the output feature map in the dense block is also determined by the growth rate. The number of a base block of the dense block comprising 4k 1 × 1 conv and 1k 3 × 3 conv varies according to the place of the dense block. In the DenseNet-161 architecture used in this study, the 1st dense block has 6, the 2nd dense block has 12, the 3rd dense block has 36, and the 4th dense block has 24 base blocks.
The weighted sum rule is expressed as follows:

\[ S_{ws} = w_1 S_1 + w_2 S_2 \]  

(4)

For the weighted product rule, \((w_1, w_2)\) are transformed into an exponential form to be multiplied by each score, to obtain the final score:

\[ S_{wp} = S_1^{w_1} \times S_2^{w_2} \]  

(5)

As shown in Equation (6), the perceptron rule is a method for finding the optimal fused score using a sigmoid function, by finding the optimal weight combination.

\[ S_p = \frac{1}{1 + \exp[-(w_0 + w_1 S_1 + w_2 S_2)]} \]  

(6)

The score-level fusion method performed based on the Bayesian rule uses each score value without specific weight values, as shown in Equation (7).

\[ S_b = \frac{S_1 \times S_2}{(1 - S_1)(1 - S_2) + S_1 \times S_2} \]  

(7)

Optimal weights in Equations (4) - (6) are found such that the error (equal error rate (EER)) of finger-vein recognition is minimized in the training data. Based on the final matching score obtained from score-level fusion, the following method is used to determine whether it is genuine matching or imposter matching. In other words, if the score is lower than the threshold set based on the EER of genuine matching and imposter matching distribution obtained from the training data, it is determined as genuine matching; if it is higher, it is determined as imposter matching. Here, EER is the error rate where the false acceptance rate (FAR) which is the error rate of incorrectly accepting imposter data as genuine one and the false rejection rate (FRR) which is the error rate of incorrectly rejecting genuine data as imposter one become identical.

V. EXPERIMENTAL RESULTS

A. DATABASE AND EXPERIMENTAL ENVIRONMENTS

The datasets used in this study are the Hong Kong Polytechnic University finger image database (version1, HKPolyU-DB) [30] and the Shandong University homologous multimodal traits finger-vein database (SDUMLA-HMT-DB) [33]. HKPolyU-DB is divided into session 1 and session 2. In this study, only the images of session 1 are used, which include index and middle-finger images of one hand of 156 individuals. The database comprises 1872 images (156 people \(\times\) 2 fingers \(\times\) 6 images), as they are taken for six times per trial. For session 2, 105 individuals from session 1 are selected to form a new session, and it comprises 1260 images (105 people \(\times\) 2 fingers \(\times\) 6 images). On the other hand, SDUMLA-HMT-DB comprises the index, middle-, and ring-finger images of both hands of 106 individuals. The database comprises 3816 images (106 people \(\times\) 2 hands \(\times\) 3 fingers \(\times\) 6 images), as they are taken for six times per trial. Different fingers represent different classes in each database. Table 3 shows that the number of classes in HKPolyU-DB is 312, while that in SDUMLA-HMT-DB is 636. While conducting this experiment, these databases are used as two-fold cross validation. In the experiment conducted using HKPolyU-DB, images of the first 78 individuals among the 156 listed, i.e., 156 classes.
are used for training in the 1st fold validation. Testing is conducted using the remaining 156 classes. For the 2nd fold validation, the training data are used for testing, and vice versa. The experiment using SDUMLA-HMT-DB is conducted in a similar manner. Images of the first 53 individuals among the 106 listed, i.e., 318 classes, are used for training in the 1st fold validation. Testing is conducted using the remaining 318 classes. For the 2nd fold validation, the training data are used for testing, and vice versa. The results of experiments conducted twice for each database are averaged and used as the final performance results. Figure 8 shows the sample image for each database.

**TABLE 3. Specification of experimental databases.**

| DB            | people | hands | Fingers | Classes | images |
|---------------|--------|-------|---------|---------|--------|
| HKPolyU-DB    | 156    | 1     | 2       | 312     | 1,872  |
| SDUMLA-HMT-DB | 106    | 2     | 3       | 636     | 3,816  |

Data augmentation through pixel translation and cropping is performed.

For pixel translation and cropping, 5-pixel vertical translation and 3-pixel horizontal translation are performed so that a total of five images, including the original image, are generated from one image, thus resulting in 5-times data augmentation. Data augmentation is performed only for the training data. The original images for which data augmentation has not been performed are used for testing. As a result, 4,680 images for HKPolyU-DB (5 × 1872) and 9,540 images for SDUMLA-HMT-DB (5 × 3816) are generated for training during the 1st and 2nd fold validation, as shown in Table 4. From these data, the number of authentic matching cases is 140,400, while that of imposter matching cases is 21,762,000 for HKPolyU-DB; these values are 286,200 and 90,725,400 for SDUMLA-HMT-DB, respectively. As there is a large gap between the number of authentic matching cases and imposter matching cases, random selection is performed for the imposter matching case for the same number as that for the authentic matching case, considering the problem of matching imbalance. During random selection, the number of imposter matching cases/authentic matching cases is set as a section, such that only one input datum is to be selected from each section, not to cause a bias in a specific class.

**TABLE 4. Specification of data augmentation.**

| DB            | Method          | # of images |
|---------------|-----------------|-------------|
| HKPolyU-DB    | 4-way shifting  | 4,680       |
|               | (up, down, left, right) |          |
| SDUMLA-HMT-DB | 4-way shifting  | 9,540       |
|               | (up, down, left, right) |          |

**B. DATA AUGMENTATION**

The number of images in the database used for the experiment is not sufficient to train the CNN used in this study. Therefore,
architecture, to compare different networks. The same configuration is applied to each architecture as for training DenseNet-161, but only initial learning rate has changed to 0.0001, and all of the architectures fine-tune the model pretrained by the ImageNet database. As the result of training conducted using the ImageNet database has 1,000 classes, the output layer is revised to perform fine tuning in this study. Furthermore, to compare the performance according to score-level fusion, the input layer is revised for each model for comparison. Channel-wise concatenation is applied to the composite image of the texture and shape images to generate a 6-channel input image, which is then used as an input in each model for which the input layer is revised. Table 5 shows the training parameters used for each model and per data type. The change in training loss is represented in a graph to examine if the training is properly conducted according to the database provided by the proposed method. As shown in Figures 9 and 10, the accuracy reaches nearly 100 as the training loss converges to 0, which indicates that DenseNet-161 is properly trained when both the texture and shape images are used.

### D. TESTING OF PROPOSED METHOD WITH HKPOLYU-DB

1) COMPARATIVE RESULTS OF PROPOSED SHAPE-BASED RECOGNITION WITH STATE-OF-THE-ART METHODS

For evaluating the performance of the proposed finger-vein segmentation algorithm, the generated shape image and the shape image generated using the finger-vein segmentation algorithm [41] are used as an input for DenseNet-161 to create scores for comparing the recognition accuracy. Figure 11 shows the receiver operating characteristics (ROC) curve for the measured shape images that are generated by each algorithm. For the ROC curve, the x-axis represents the FAR value, while the y-axis represents the genuine acceptance rate (100 – FRR (%)).

Table 6 shows the recognition error incurred by the fold based on DenseNet-161, using the shape image extracted from the HKPOLYU-DB database.
using the respective algorithms. Our proposed algorithm exhibits more outstanding performance than the existing method. The shape image generated by the proposed method contains the compressed information with gaps and burrs removed. Regions broken by component labeling and morphological operations [31] are removed and vein regions are roughly extracted, such that the loss of information contained in the finger-vein image is minimized during the segmentation, as shown in Figure 12(b). However, the shape image generated by the existing method [41] has additional noise due to the algorithms used, and particularly, images having numerous gaps and burrs are generated as shown in Figure 12(c). Hence, the appearance of the error case of each method is different. Accordingly, the error rate of each method varies in the 1st and 2nd folds. In particular, the false rejection case is significantly fluctuated due to the misalignment. The proposed method, which grasps the regions roughly, has a few error cases, whereas the existing method [41], which generates few vein-regions but various noises, has an increasing number of false rejection cases due to the misalignment.

2) COMPARATIVE RESULTS OF PROPOSED FUSION METHOD WITH SINGLE MODALITY, IMAGE CONCATENATION METHODS, AND THE STATE-OF-THE-ART METHODS

In this section, the following experiments are conducted for the HKPolyU database: shape image and texture image are used as an input for VGG-16 [44], DenseNet-161 [32], and ResNet-152 [45] to compare the performance; channel-wise concatenation is applied to the shape and texture images to generate an image to be used as an input for performance comparison; and finally, score-level fusion is conducted for CNN output scores of shape and texture images to compare the performance of each architecture.

The composite images generated using the texture and shape images are used as an input for each network for comparison. The EER of both images shown in the first and second columns in Table 7 is the average of the 1st and 2nd fold results, respectively. The results of using the shape image exhibit worse performance, on average, as compared to those of using the texture image. Furthermore, two types of data (shape image and texture image) are concatenated with six channels, to be used as an input for the recognition process. Before conducting the experiment, fine tuning cannot be conducted because the number of input channels required for each of the previously used networks is different from the input data of the experiment. Therefore, channel-wise concatenation is conducted for the weights of the convolutional filter in the 1st convolutional layer to be more appropriate for 6-channel input, in order to carry out transfer learning. As shown in the third column of Table 7, the recognition rate is lower than when only shape and texture images are used. This is because, when recognition is conducted using a concatenated image, the shape and texture image inputs do not complement each other in the same matching case; rather, they interrupt the decision of each other. In the finger-vein recognition system in which a single data type of the texture image or shape image is used, the texture image has the purpose of carrying out recognition by using information on pixel variation inside of finger-vein and the background region that exists in finger-vein image. However, the shape image has the purpose of carrying out recognition by learning only the shape information of finger-veins. As the data characteristics of each type are different, six channel-wise concatenations are applied to simultaneously use the
TABLE 7. Comparisons of EERs by each input type and score-level fusion methods (unit: %).

| Input type & method | VGG-16 [44] | ResNet-152 [45] | DenseNet-161 [32] |
|---------------------|-------------|-----------------|-------------------|
| Texture image       | 0.8294      | 0.7098          | 0.5845            |
| Shape image         | 0.8986      | 0.7663          | 0.7635            |
| Concatenated image  | 1.0752      | 0.4929          | 0.77              |

Score-level fusion

| Method         | EER   |
|----------------|-------|
| Weighted sum   | 0.2543|
| Weighted product| 0.2140|
| Bayesian       | 0.27  |
| Perceptron     | 0.4009|

E. TESTING OF PROPOSED METHOD WITH SDUMLA-HMT-DB

As shown in Section V.D.2, the following experiments are conducted for the SDUMLA-HMT database: shape image and texture image are used as input for VGG-16 [44], DenseNet-161 [32], and ResNet-152 [45] to compare the performance; channel-wise concatenation is applied to the shape image and texture image to generate an image to be used as an input for performance comparison; and finally, score-level fusion is conducted for CNN output scores of the shape image and texture image to compare the performance of each architecture. As shown in Table 7, the best performance of EER at 0.05% is achieved when score-level fusion is performed using the weighted sum rule. Moreover, the recognition rate of DenseNet-161 is higher than that of VGG-16 or ResNet-152 for all cases.

Figure 13 shows the ROC curves for all methods listed in Table 7. The recognition performance is the highest when score-level fusion (weighted sum) is performed based on DenseNet-161, as shown in Table 7.

Table 8 compares the recognition performance of the proposed method and the state-of-the-art methods. The state-of-the-art methods are compared by dividing them into non-training-based and training-based methods. As shown in the table, the proposed method exhibits better recognition performance than the state-of-the-art methods when the HKPolyU database is used.

TABLE 8. Comparison of EERs produced by the proposed method and the state-of-the-art methods using HKPolyU database (unit: %)* EER is referred from [46].

| Method                        | EER   |
|-------------------------------|-------|
| *Huang et al. [42]            | 9.46  |
| *Miura et al. (MCP) [43]      | 6.49  |
| Liu et al. [47]               | 5.01  |
| Gupta et al. [11]             | 4.47  |
| *Miura et al. (RPL) [3]       | 4.45  |
| Dong et al. [48]              | 3.53  |
| Liu et al. [49]               | 1.47  |
| Xi et al. [50]                | 1.44  |
| Joseph et al. [46]            | 1.27  |
| Liu et al. [51]               | 0.69  |
| Kumar et al. [30]             | 0.65  |
| Yang et al. [52]              | 0.38  |
| Xie et al. [53]               | 8.87  |
| Qin et al. [54]               | 3.02  |
| Zhang et al. [55]             | 1.67  |
| Song et al. [2]               | 0.33  |
| Proposed method               | 0.05  |

TABLE 9. Comparisons of EERs by each input type and score-level fusion methods (unit: %).

| Input type & method | VGG-16 [44] | ResNet-152 [45] | DenseNet-161 [32] |
|---------------------|-------------|-----------------|-------------------|
| Texture image       | 2.72        | 2.19            | 2.14              |
| Shape image         | 5.46        | 4.75            | 4.50              |
| Concatenated image  | 5.02        | 2.99            | 2.17              |

Score-level fusion

| Method         | EER   |
|----------------|-------|
| Weighted sum   | 2.15  |
| Weighted product| 2.15  |
| Bayesian       | 2.26  |
| Perceptron     | 2.19  |

VOLUME 8, 2020
FIGURE 13. Comparative ROC curves according to various input types and score-level fusion method by using (a) VGG-16, (b) ResNet-152, and (c) DenseNet-161.
FIGURE 14. Comparative ROC curves according to various input types and score-level fusion method by using (a) VGG-16, (b) ResNet-152, and (c) DenseNet-161.
The analyses for the imposter matching case is conducted in the same manner as the analysis scheme for the activation map of a genuine matching case. As shown in Figure 15, the class activation maps of both shape and texture images have high activation values in the same manner as the analysis scheme for the activation map of a genuine matching case. As shown in Figure 16, the same genuine matching case is compared based on the texture image and shape image. The locations of the layer from which the activation map is extracted are \( 7 \times 7 \) conv layer before the first dense block, \( 1 \times 1 \) conv layer of the 1\(^{\text{st}}\), 2\(^{\text{nd}}\), and 3\(^{\text{rd}}\) transition layers, and finally, output (\( 3 \times 3 \) conv layer of the last dense block) of the last dense block. An activation map is generated based on the number of filters in each layer. It is shown in a grid form, and then, certain parts were enlarged. The left-hand side of Figure 15 shows the activation map generated using the texture image, while the right-hand side shows the activation map generated using the shape image. The activation map shown in Figure 15(b) is a high-level feature extracted from the \( 7 \times 7 \) conv layer near the bottom layer. When the texture image on the left is used, clear information on finger-vein regions is not extracted. On the other hand, when the shape image on the right is used, more evident finger-vein region information is discovered because the shape image of finger-vein regions has been extracted in advance. This shape information is maintained clearly in Figures (c) and (d), which indicates that the shape image utilizes the finger-vein shape information. On the other hand, when the texture image is used, it was evidently shown in the activation map that the background region information is largely used besides the finger-vein region. Finally, the two data types focus on different information, thus resulting in improved performance during score-level fusion. As shown in Figure 15(b), both images have clear representation of the finger-vein region. However, in Figure 15(c), the shape image forms the activation map based on the finger-vein region, while the texture image extracts features using regions other than the finger-vein region, such as the background. However, it was confirmed in the previous section that finger-vein recognition based on the texture image performs better. The finger-vein region may not have been extracted accurately when the shape image was generated, or the background information was not entirely insignificant during recognition. As shown in Figure 15(f), the class activation maps of both texture and shape images have high activation values in the central region of the image.

### Table 10. Comparison of EERs by the proposed method and state-of-the-art methods using SDUMLA-HMT database (unit: %).

| Method                  | EER  |
|-------------------------|------|
| Non-training based      |      |
| Pham et al. [56]        | 8.09 |
| Miura et. al. [3]       | 5.46 |
| Miura et. al. [43]      | 4.54 |
| Yang et al. [57]        | 3.46 |
| VGG-16 [44]             | 3.90 |
| Training based          |      |
| ResNet-101 [45]         | 3.36 |
| Song et al. [2]         | 2.35 |
| Proposed method         | 1.65 |

This table, the proposed method exhibits better recognition performance than the state-of-the-art methods in the case where the SDUMLA-HMT database is used.

All methods in Table 10 are texture image-based, except for the proposed method. Nevertheless, the proposed method exhibits better recognition performance because ample information regarding texture data and compact data generated by finger-vein extraction in the shape image are used simultaneously. The experimental results obtained using only the shape image exhibits the worst performance for all cases, as shown in Table 9. However, the performance certainly improves through score-level fusion. In the experiments using the SDUMLA-HMT database, features extracted by the CNN and the trained information are evidently different when the shape image is used and when the texture image is used, which is similar to the experimental results obtained using the HKPolyU database. Likewise, as the two types have an anisotropic relation, score-level fusion is effective, which results in higher recognition performance than the existing methods.

### F. ANALYSES OF CLASS ACTIVATION MAP FROM SHAPE AND TEXTURE IMAGES

In this section, the analyses of the class activation map obtained from the shape and texture images using the proposed method are shown. During finger-vein recognition, which area generates high activation and which features are extracted by each layer are analyzed to verify if the matched cases are classified based on the significant information obtained regarding each datum. We will also analyze which layer focuses on which feature for each level feature. For analyzing the activation map, the output of certain layers in DenseNet-161 used in the proposed method is extracted to be analyzed by the input type. Furthermore, the Grad-CAM [58] method is used for analyzing the class activation map, and the gradient of the score generated as the final output on the feature map of the examined layer is calculated and multiplied, which shows the grounds on which the current layer produces the score for each class.

#### 1) ANALYSES OF CLASS ACTIVATION MAP OF GENUINE MATCHING CASE

In this section, the respective class activation map of the same genuine matching case is compared based on the texture image and shape image. The locations of the layer from which the activation map is extracted are \( 7 \times 7 \) conv layer before the first dense block, \( 1 \times 1 \) conv layer of the 1\(^{\text{st}}\), 2\(^{\text{nd}}\), and 3\(^{\text{rd}}\) transition layers, and finally, output (\( 3 \times 3 \) conv layer of the last dense block) of the last dense block. An activation map is generated based on the number of filters in each layer. It is shown in a grid form, and then, certain parts were enlarged. The left-hand side of Figure 15 shows the activation map generated using the texture image, while the right-hand side shows the activation map generated using the shape image. The activation map shown in Figure 15(b) is a high-level feature extracted from the \( 7 \times 7 \) conv layer near the bottom layer. When the texture image on the left is used, clear information on finger-vein regions is not extracted. On the other hand, when the shape image on the right is used, more evident finger-vein region information is discovered because the shape image of finger-vein regions has been extracted in advance. This shape information is maintained clearly in Figures (c) and (d), which indicates that the shape image utilizes the finger-vein shape information. On the other hand, when the texture image is used, it was evidently shown in the activation map that the background region information is largely used besides the finger-vein region. Finally, the two data types focus on different information, thus resulting in improved performance during score-level fusion. As shown in Figure 15(b), both images have clear representation of the finger-vein region. However, in Figure 15(c), the shape image forms the activation map based on the finger-vein region, while the texture image extracts features using regions other than the finger-vein region, such as the background. However, it was confirmed in the previous section that finger-vein recognition based on the texture image performs better. The finger-vein region may not have been extracted accurately when the shape image was generated, or the background information was not entirely insignificant during recognition. As shown in Figure 15(f), the class activation maps of both texture and shape images have high activation values in the central region of the image.

#### 2) ANALYSES CLASS ACTIVATION MAP OF IMPOSTER MATCHING CASE

The analyses for the imposter matching case is conducted in the same manner as the analysis scheme for the activation map of a genuine matching case. As shown in Figure 16, a high activation occurs only in minor regions, unlike the
FIGURE 15. Examples of class activation maps from particularly intermediate layers in case of authentic matching case: (a) Input of authentic matching case of each input type (upper and lower images are enrolled and input images, respectively), and others are class activation map from (b) the 1st $7 \times 7$ conv layer, (c) $1 \times 1$ conv layer of the 1st transition layer, (d) $1 \times 1$ conv layer of the 2nd transition layer, (e) $1 \times 1$ conv layer of the 3rd transition layer, and (f) $3 \times 3$ conv layer of the 4th dense block. The left and right images of (a)–(f) show the class activation maps for the texture and shape images, respectively.

FIGURE 16. Examples of class activation maps from particularly intermediate layers in imposter matching case: (a) Input of authentic matching case of each input type (upper and lower images are enrolled and input images, respectively), and others are class activation map from (b) the 1st $7 \times 7$ conv layer, (c) $1 \times 1$ conv layer of the 1st transition layer, (d) $1 \times 1$ conv layer of the 2nd transition layer, (e) $1 \times 1$ conv layer of the 3rd transition layer, and (f) $3 \times 3$ conv layer of the 4th dense block. Left and right images of (a)–(f) show the class activation maps obtained from the texture and shape images, respectively.
genuine matching case. The location of high activation is the region where the enrolled image and matched image overlap.

For the imposter matching case, the same aspect as that of the activation map of the genuine matching case is shown. A composite image generated using the shape image is set to be classified based on the difference in the size of regions where genuine matching and imposter matching cases overlap. Considering the activation map extracted from each layer, the regions where the channels overlap are extracted properly. For the imposter matching case, features are extracted based on the finger-vein region more than when the texture image is used as an input. Ultimately, the size of high activation generated in the imposter matching case in Figure 16 is smaller than that in the genuine matching case in Figure 15; high activation values are obtained in outer regions of the image rather than the central region. There are many regions where channels overlap for the genuine matching case where images having the same class are in each channel of the composite image, which shows that the high activation region is larger than that in imposter matching case. For both imposter matching case and genuine matching case, high activation tends to occur as the combined region where the enrolled and matched image, which are input in each channel of the composite image, overlap is larger in the convolutional layer near the input layer of the network. The same tendency is observed for the final output layer, where high activation is generated only in the region with the largest combined region value, while activation decreases in the remaining regions. Figures 15 and 16 show that genuine matching and imposter matching generate different class activation maps. These results show that DenseNet-161 with texture and shape images used in this study is sufficiently trained for finger-vein recognition.

G. PROCESSING SPEED
The processing speed is measured in the desktop computer environment described in Section V.A and Jetson TX2 embedded system [59], which is shown in Figure 17, in order to verify if the proposed algorithm can be applied in an access-controlled environment or a mobile device in the future. The Jetson TX2 board is an embedded system having NVIDIA Pascal™GPU architecture with 256 NVIDIA CUDA cores, 8GB 128-bit LPDDR4 memory, and Dual-Core NVIDIA Denver 2 64-Bit CPU. The amount of electricity consumed is less than 7.5 W. Finger-vein recognition using the proposed method is ported by Keras [36] and Tensorflow [60] of Ubuntu 16.04 OS. The versions of the installed framework and library are Python 3.5 and Tensorflow 1.12, respectively; the NVIDIA CUDA® toolkit [38] and NVIDIA CUDA® deep neural network library (CUDNN) [39] use versions 9.0 and 7.3, respectively.

Table 11 shows a comparison of the processing time in a desktop computer environment and an embedded system (Jetson TX2). As shown in Table 11, the algorithm proposed in this study is applicable in a desktop computer environment as well as the embedded system with limited resources.

VI. CONCLUSION
This paper proposed a deep CNN-based finger-vein recognition system with improved recognition performance, where score-level fusion is performed by simultaneously using the texture image for which finger-vein extraction is not applied and the shape image for which finger-vein extraction is applied. By analyzing the activation maps of the shape image and texture image, it was found that each datum extracts different features, thereby effectively complementing each other. Various score-level fusion methods and six channel-wise concatenated images were compared, and the weighted sum rule achieved the best recognition performance. Moreover, we proposed an algorithm with better performance than the existing algorithm for finger-vein extraction. When the EER was compared by testing on the same network for the segmented image, it was verified that the proposed algorithm exhibited better performance. In the finger-vein recognition system where only one type of shape image or texture image was used, various error cases were generated due to the limitations of each data type. Considering such drawbacks, a finger-vein recognition system with improved performance through score-level fusion was proposed in this paper.

In the future, studies on lighter model for solving the issue of long processing time, caused using two input images, as well as that on the measures to shorten the processing time of the preprocessing algorithm, will be conducted. Furthermore, the possibility of applying the proposed algorithm to other types of biometric data (face, iris, and fingerprint) besides finger-vein will be investigated.
H. Liu, L. Yang, G. Yang, and Y. Yin, “Discriminative binary descriptor for finger vein recognition,” *IEEE Access*, vol. 6, pp. 5795–5804, Dec. 2018.

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.

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.

H. Qin and M. A. El-Yacoubi, “Deep representation-based feature extraction and recovering for finger-vein verification,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.

Y. Zhang, W. Li, L. Zhang, X. Ning, L. Sun, and Y. Lu, “Adaptive learning Gabor filter for finger-vein recognition,” *IEEE Access*, vol. 7, pp. 159821–159830, Oct. 2019.

T. Pham, Y. Park, D. Nguyen, S. Kwon, and K. Park, “Noninvasive finger-vein recognition system using NIR image sensor and accuracy analyses according to various factors,” *Sensors*, vol. 15, no. 7, pp. 16866–16894, Jul. 2015.

L. Yang, G. Yang, X. Xi, X. Meng, C. Zhang, and Y. Yin, “Tri-branch vein structure assisted finger vein recognition,” *IEEE Access*, vol. 5, pp. 21020–21028, Jul. 2017.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE Int. Conf. Comput. Vis.* (ICCV), Venice, Italy, Oct. 2017, pp. 618–626.

Jetson TX2 Module. Accessed: Jul. 12, 2019. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

Tensorflow: The Python Deep Learning Library. Accessed: Jul. 19, 2019. [Online]. Available: https://www.tensorflow.org/

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.

**Kyoung Jun Noh** received the B.S. degree in electronics and electrical engineering from Dongguk University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S. degree in electronics and electrical engineering. He designed finger-vein recognition system based on score-level fusion method, conducted and analyzed experiments, and wrote the original article. His research interests include biometrics and pattern recognition.

**Jiho Choi** received the B.S. degree in business administration from Dongguk University, Seoul, South Korea, in 2016, where he is currently pursuing the combined M.S. and Ph.D. degrees in electronics and electrical engineering. He helped to implement the preprocessing method and experiments. His research interests include biometrics and pattern recognition.

**Jin Seong Hong** received the B.S. degree in international business administration from Pukyong National University, Busan, South Korea, in 2014. He is currently pursuing the combined M.S. and Ph.D. degrees in electronics and electrical engineering with Dongguk University, Seoul, South Korea. He helped to implement the preprocessing and experiments. His research interests include biometrics and pattern recognition.

**Kang Ryoung Park** (Member, IEEE) 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 March 2013. His research interests include image processing and biometrics. He supervised this research and helped the revision of original article.