IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 11, 2022, accepted January 25, 2022, date of publication January 31, 2022, date of current version February 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147807

ECG-Based Biometrics Using a Deep Network Based on Independent Component Analysis

JAE-NEUNG LEE[1] AND KEUN-CHANG KWAK[2], (Member, IEEE)

[1]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[2]Department of Electronics Engineering IT-Bio Convergence System, Chosun University, Gwangju 501759, South Korea

Corresponding author: Keun-Chang Kwak (kwak@chosun.ac.kr)

This study was supported by research fund from Chosun University (2021).

ABSTRACT In this study, we propose a systematic deep network based on independent component analysis (ICA) called ICANet1, which is subsequently modified into an improved model called ICANet2. The existing principal component analysis network (PCANet) has a smaller computational complexity and is faster than deep learning. However, any modifications in the data lower the performance, which is a limitation. The ICANet algorithm has been proposed to solve this problem. Although ICANet2 does not use eigenvectors, it uses PCA feature vectors. To eliminate the correlation between the PCA feature vectors, the ICA algorithm is used to determine those that are statistically independent. The feature values obtained from the final histogram are vectorized and used as input to the classifier. On using ICANet2, the performance achieved is lower than that of deep learning; however, it is expected to be superior with respect to the recognition speed, especially for mobile devices. The classifier performance is demonstrated using extreme learning machine (ELM), artificial neural network (ANN), k-nearest Neighbor (KNN), and support vector machines (SVM). We use the MIT-ECG, Chosun University-electrocardiogram (CU-ECG), and noise ECG databases to verify the performance of the proposed method. Both ICANet1 and ICANet2 demonstrate better experimental performance than PCANet because the ECG noise data are affected by time. In addition, when noise ECG data are used, the ICANets achieved a better performance than PCANet, further proving its validity.

INDEX TERMS Chosun University-electrocardiogram (CU-ECG), independent component analysis network (ICANet), personal identification.

I. INTRODUCTION

Deep learning is defined as a set of machine-learning algorithms that depend on neural networks derived via machine learning, and it attempts a high level of abstraction (a task that summarizes key content or functions in large amounts of data or complex data) by combining several nonlinear transformation techniques. It is also considered a field of machine learning that teaches computers the mechanism by which people think general thought process of people. Data are always expressed in a form that the computer understands (for example, in the case of images, a column vector represents the pixel information.) As a result of these efforts, various deep-learning techniques, such as deep neural networks, convolutional neural networks (CNNs), and deep belief networks are used for realizing computer vision, speech recognition, and natural language processing. Such techniques have been applied in the field of voice/signal processing and have yielded cutting-edge results [1]–[3].

However, there are certain limitations in deep learning, such as the time and cost of collecting big data, low efficiency (which can be resolved by repeating the procedure several times), poor theoretical verification of algorithms (except slope descent), and the black box which is the lack of transparency in the functioning of the algorithm [4]–[6].

A CNN is a basic deep learning architecture, which has been applied to text, image, and speech recognition. Another advantage of CNN is its ability to automatically extract and learn hidden representations of data across a number of blocks consisting of a convolutional layer, activation function layer, and max-pooling layer. However, choosing the method for selecting the parameters and configurations including the

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

filter sizes, number of layers, and pooling function, remains a significant challenge. Consequently, the use of network structures such as AlexNet, ResNet, and GoogLeNet are expanding. Despite the great success of deep learning, researchers have determined the best feature learning mechanism and optimal network configuration yet. To this end, Chen studied the principal component analysis network (PCANet), which is a shallow and intensive deep learning network. The simplicity of training basic deep learning networks and their ability to adapt to different databases and tasks are beneficial. Consequently, such a basic network could serve as a suitable empirical reference for researchers to justify the use of more advanced processing components or sophisticated architectures for their deep learning networks. To organize the structure of the ICANet algorithm, we examined the trends across various research papers.

First, we list the trends concerning principal component analysis (PCA), which is the origin of the PCANet algorithm. Jolliffe proposed PCA in 2002 [7]. Chungnam National University Bui *et al.* [8] studied in-video human detection and proposed estimation using motion data based on the robust PCA (RPCA) algorithm. Chosun University [9] performed personal identification using electrocardiogram (ECG) signal data based on a multilinear PCA (MPCA) algorithm. The existing one-dimensional (1D) ECG signal was transformed into a three-dimensional (3D) shape to improve the recognition rate. Seoul National University Park and Oh [10] classified images using kernel PCA (KPCA) with an image database class. At Suwon University, Kim *et al.* [11] used a numerical incremental PCA based on handwritten numerical data.

Studies conducted overseas that employed RPCA are listed as follows. Jin *et al.* [12] detected contrast-filled vessels using X-ray data based on RPCA. Lu *et al.* [13] classified facial images and object data based on RPCA. Li *et al.* [14] used RPCA on image data to detect moving objects. Vaswani *et al.* [15] performed subspace tracking and restoration using image data based on RPCA. Liu *et al.* [16] performed tumor classification using RPCA on MRI data. Sun and Du [17] performed geospatial classification using RPCA on hyperspectral data. Zhong *et al.* [18] classified faces and numbers using a robust linear discriminant analysis (LDA) on facial and numerical data. Yan *et al.* [19] used face and numerical data based on RPCA.

The following studies used KPCA. Kuang *et al.* [20] performed intrusion detection using KPCA on communication data. Liu *et al.* [21] performed mental fatigue estimation using the KPCA-hidden Markov model based on EEG data. Vinay *et al.* [22] and Wang [23] performed facial recognition using KPCA on facial data. Romero *et al.* [24] classified images using KPCA on satellite image data. Xia *et al.* [25] performed image classification using KPCA on hyperspectral image data. Yuan *et al.* [26] studied smoke detection using KPCA on image data. Hotta [27] performed scene classification using KPCA on image data. Xiao *et al.* [28] performed novelty detection using KPCA on image data comprising square, spiral, and banana motifs.

The following paragraph lists the studies concerning the independent component analysis (ICA) algorithm and algorithms derived from ICA. First, Hong and Cho [29] performed facial-expression recognition using facial data based on ICA in Korea. Hwang and Kim [30] studied the mode separation of buildings using ICA based on sound data. Baek and Hong [31] classified images using ICA on image data. Kwon and Lim [32] used ICA on image data to classify images of red peppers. Kang *et al.* [33] studied the translational warping mode of dry matter using ICA on sound data. Kim [34] analyzed the vibration-source contribution of plate structures using ICA on vibration signal data. Kim *et al.* [35] studied the identification of vibration source signals of structures using ICA on sound data. Quan and Bae [36] studied frequency binarization using frequency-domain ICA on sound data. Park [37] studied blind signal separation using ICA on voice data. Hwang [38] studied a new mode separation technique that employed the acceleration response of structures using sound data based on ICA. Lee and Lee [39] designed a dynamic noise-reduction filter using ICA. Symposium [40] designed a partial discharge pattern classifier using ICA based on high-dimensional data.

Wu *et al.* [41] performed gender recognition using multiscale ICA on image data. Sun *et al.* [42] performed tumor classification using microarray data on ICA. Martis *et al.* [43] performed ECG beat classification using ECG data based on the ICA. Wang and Zhang [44] performed video event classification using ICA on video data. Yu *et al.* [45] performed image classification based on image data and ICA. Falco *et al.* [46] performed hyper-spectral image classification using ICA algorithms with different levels of effectiveness. Stewart *et al.* [47] performed a single-trial classification using ICA based on EEG data. Xiao *et al.* [48] studied sparse representations using an image-reconstruction ICA.

An initial motivation of our study is to apply a simple deep learning network using ECG. There is a lot of approaches to make a model simple such as mobilenet to apply mobile environment. Similarly, ICANet is applied to propose a lightweight model that can be used in a mobile environment. PCANet is an algorithm that is mainly used in face recognition and facial data is mainly composed of two dimensions.

The contributions of this study are as follows. First, we generalize the ICANet algorithm. The theoretical and structural background of the proposed ICANet1 algorithm is explained. In addition, this study aims to extend the algorithms from ICANet1 to ICANet2 to explore the concept further. Second, we performed a detailed analysis and visualized the effects of the parameters such as number of filter and block size. Finally, the ECG data were used to supplement the personal authentication security problem.

The remainder of this paper is organized as follows. Section 1 describes the motivation for developing the ICANet algorithm and lists the algorithms derived from the PCANet algorithm that overcome the shortcomings of deep learning. This indicates the emergence of PCANet and ICANet. In Section 2, we present the research studies involving ICA, deep learning, PCANet, and local binary pattern (LBP) algorithms. In addition, we list the methods utilized in the PCANet and ICANet algorithms. Section 3 presents ICANet, which is the focus of this study. We prove the validity of the proposed algorithm using an ECG database and compare the effects of each parameter, which is described in Section 4. Finally, Section 5 concludes the paper.

II. PCANET AND ITS VARIANCE WITH ICA

A. ICA

In this section, we present the ICA algorithm used to address the recognition problem and describe the procedure for developing it for different architectures.

First, we address the motivation behind the ICA. Let us assume that two persons in a room speak simultaneously. The room has two microphones that are placed at different locations to obtain two voice signals, which are denoted by $\mathrm{x}_1(t)$ and $\mathrm{x}_2(t)$, where x_1 and x_2 represent the amplitudes, and t represents the time index. The signal recorded using the microphone is the weighted sum of the speech signals emitted by the two speakers, which we denote as $\mathrm{s}_1(t)$ and $\mathrm{s}_2(t)$. This can be expressed by the following linear equation:

$$x_1\,(t) = a_{11}s_1 + a_{12}s_2 \tag{1}$$

$$x_2\,(t) = a_{21}s_1 + a_{22}s_2 \tag{2}$$

where a_{11}, a_{12}, a_{21}, and a_{22} are parameters that depend on the distance between the speakers and microphones. Thereafter, we estimated the two original speech signals s_1(t) and $s_2(t)$ using only the respective recorded signals $x_1(t)$ and $x_2(t)$. In summary, the motivation for developing ICA is similar to that of CPP, and therefore, it is robust against noise. Second, we defined ICA using an algorithm. To define ICA, we used a statistical "latent variable" model. Let us assume that we observe n linear mixtures, $x_1,\ldots, x_n$ of n independent components.

$$x_j = a_{j1}s_1 + aj_2s_2 + \cdots a_{jn}, s_n, \quad for\ all\ j \tag{3}$$

We originally defined the time index t in the ICA algorithm. Now, let us assume that each mixture, x_j, as well as each independent component s_k, is a random variable instead of a suitable time signal. The observed data $x_j(t)$, which are the microphone signals in the CPP, can then be considered as a sample of this random variable. Without loss of generality, we assume that both the mixture and the independent components have a zero mean. If this is not true, then the observable variables x_i can always be centered by subtracting the sample mean, which makes the model zero-mean.

We use the vector-matrix notation instead of summation notation herein. Let x be a random vector with $x_i, \cdots, x_n$ elements and s be a random vector with $s_1, \cdots, s_n$ elements. Let A be a matrix with elements a_{ij}. Generally, bold lowercase letters indicate vectors, and bold uppercase letters denote matrices. All vectors are assumed as column vectors; thus, x^T, or the transpose of x, is a row vector. Using this vector matrix notation, the abovementioned mixing model from equation (1) can be expressed as follows.

$$x = As \tag{4}$$

However, the objective of the first architecture model is to identify a set of statistically independent basis images from the eigenvectors obtained using the PCA method. Fig. 1 depicts the ECG representation of the ICANet.

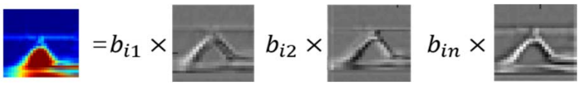


FIGURE 1. ECG representation of ICANet.

B. PCANet

In this section, we first review the PCANet. Its architecture is depicted in Fig. 2, and Fig. 3 depicts the ICANet workflow for extracting features from the training dataset.

Stage 1 of PCANet performs patch sliding operation followed by the mean removal step. The equation is as follows.

$$\overline{Y} = patc\,hY - patc\,h\overline{Y_i} \in R^{k_s^2\times(m-k_S+1)(n-k_s+1)} \tag{5}$$

Thereafter, convolution is performed using PCA.

$$v_i = \overline{Y}^T v_i', \quad s.t.\ v_i' = \overline{Y}^T v_i \tag{6}$$

The final output obtained can be expressed as:

$$I^s i, l_i = v_i X_{padding} \tag{7}$$

The second stage is similar to the first, and the binary hashing of the output layer can be calculated as:

$$P_{i,l,\iota} = H\left(I^{II}_{i,l},_{\iota}\right),\ l = 1, 2, \ldots, L_1; \\ \iota = 1, 2, \ldots, L_2; \quad i = 1, 2, \ldots, N \tag{8}$$

Finally, the histogram step can be performed as follows.

$$f_i = [Hist\left(z_{i,1,1}\right)\cdots Hist\left(z_{i,1,B}\right)\cdots Hist\left(z_{i,L_1,1}\right) \\ \cdots Hist\left(z_{i,L_1,B}\right)] \in R^{(2^{L_2})L_1B} \tag{9}$$

The final output is as follows.

$$Output: f_i = (f_1\cdots f_N) \in R^{(2^{L_2})L_1BN} \tag{10}$$

A detailed description of the workflow is provided below.

[Step 1] A mean removal step was performed to obtain the center image. The central image was obtained using an average value.

[Step 2] The PCA filter bank extracts PCANet filters.

[Step 3] The PCA filter bank outputs eigenvectors.

[Step 4] Original images are convoluted with the output of the PCANet filters for the next step.

[Step 5] Eigenvectors is convoluted with the output of the PCANet filters for the next step.

[Step 6] The output image is binarized, and block-wise histograms are calculated. Here, we create a weight map and proceed with binary quantization and weighted combination of the elements in the input data.

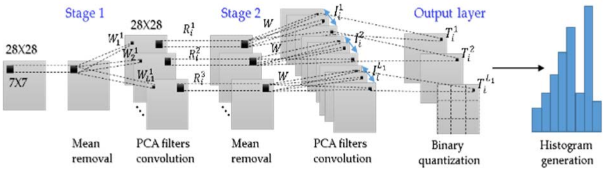


FIGURE 2. Diagram of PCANet.

III. PROPOSED ICANet

A. ICANet1

ICANet1 (statistically independent basic image Net) is a modified ICA algorithm, which is an algorithm that extracts independent components through the ICA filter using eigenvectors obtained from PCA. The first stage is one of simple deep learning, wherein the PCA and ICA algorithms form a cascade network. In the second step, three algorithms, PCA, ICA, and LBP, were implemented using image patch data. In the hashing code step, which is similar to that of the LBP, the feature using the histogram is vectorized. Finally, the classification was completed using SVM. Fig. 3 depicts the structure of ICANet1.

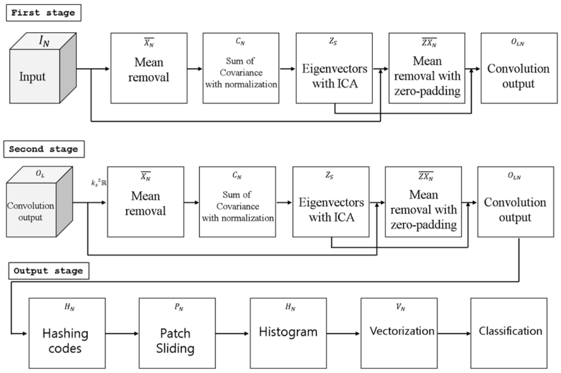


FIGURE 3. Structure of ICANet1.

To identify a statistically independent basic image set according to a set of ECGs, the independent components of the ECG image are separated based on the image synthesis model shown in Fig. 4. Because existing eigenvectors are not statistically independent, the goal of ICANet1 is to identify eigenvectors that are statistically independent using ICA. The ECG image of X is assumed to be a linear mixture of an unknown set of statistically independent source images S, and A is an unknown mixing matrix, as depicted in Fig. 4. The source is reconstructed using a W-learning filter matrix, which produces a statistically independent output. When there is an ECG image called S, the noise signals A and the ECG data S are combined to generate noise data called X. Here, the noise component is separated, and the output U can be represented. The coefficients of the linear combination of the independent basic images of U that comprise each ECG image are shown in Fig. 4. In this model, the coefficient matrix B is obtained by mixing matrix A $\triangleq \mathbf{W_I^{-1}}$. Each row of A contains a coefficient b for one image x.

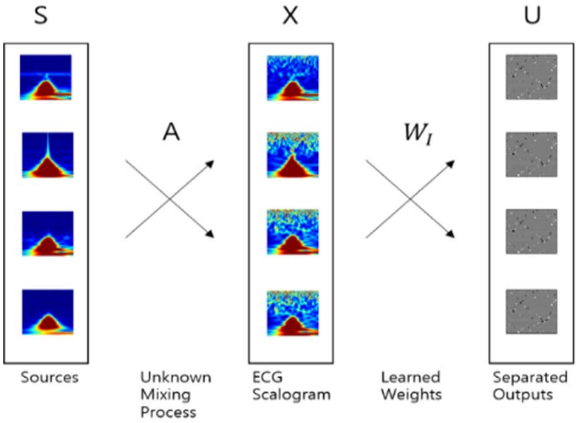


FIGURE 4. An image synthesis model.

The following describes the steps in ICANet1 from the first to the last stages.

[Step 1] First patch sliding process.

Before sliding, the images are padded to $I_i' \in R^{(m+k_1-1)\times(n+k_2-1)}$, and the out-of-range input pixels are assumed to be zero. This ensures that all weights in the filters reach the entire area of the images. We used a patch of size $k_1 \times k_2$ to slide each pixel of the i^{th} image, $I_i' \in R^{(m+k_1-1)\times(n+k_2-1)}$, and subsequently reshaped each $k_1 \times k_2$ matrix into a column vector.

$$X_i = [x_{i,1}x_{i,2} \cdots x_{i,mn}] \in R^{k_1k_2\times mn} \tag{11}$$

where $x_{i,j}$ denotes the j^{th} vectorized patch in I_i. Thus, for all input training images $I_i, i = 1, 2, \ldots, N)$, we obtain the following matrix:

$$X = [X_1, X_2 \cdots X_N] \in R^{k_1k_2\times Nmn} \tag{12}$$

$$X = [X_1, X_2 \cdots X_N] \in R^{k_s^2\times(m-k_s+1)(n-k_s+1)N} \tag{13}$$

[Step 2] First mean removal process.

In this step, we subtracted the patch mean from each patch and obtained the following:

$$patc\,h\overline{X}_i = [\overline{X}_{i,1}\overline{X}_{i,2} \cdots \overline{X}_{i,mn}] \in R^{k_s^2\times(m-k_s+1)(n-k_s+1)N} \tag{14}$$

By subtracting each patch image from (14), we obtain the following expression:

$$\overline{X} = patc\,hX - patc\,h\overline{X}_i \in R^{k_s^2\times(m-k_s+1)(n-k_s+1)} \tag{15}$$

[Step 3] First PCA process with ICA

In this step, the eigenvalues and eigenvectors were determined via (17) using the PCA algorithm. The covariance was obtained by repeating (17) for all patches and images.

$$\overline{X_{Cov}} = \sum_{N=1}^{NL} (\overline{X} \times \overline{X}^T) \tag{16}$$

In the above equation, N is the number of images and L is the number of filters. The eigenvector can be obtained using the following equation:

$$\overline{X_{Cov}} \times \overline{X_{Cov}}^T v_i = \lambda_i v_i \tag{17}$$

$$\overline{X_{Cov}}^T \times \overline{X_{Cov}} (\overline{X_{Cov}}^T v_i) = \lambda_i (\overline{X_{Cov}}^T v_i) \tag{18}$$

$$v_i \times \overline{X}^T v_i', \quad s.t\ v_i' = \overline{X}^T v_i \tag{19}$$

$$U = W_{ICA} v_{i(PCA)} \tag{20}$$

Let us assume that the eigenvector obtained from (19) is L. To use the ICA algorithm to estimate the independent component, we applied eigenvectors to the ICA. First, the centering and whitening operations of ICA were applied, and thereafter, data normalization was performed. Subsequently, the initial weight was set randomly, and the initial weight and the data obtained from the whitening were linearly combined. Kurtosis and negentropy are two ICA methods. Therefore, the expression is as follows.

$$kurt\ (y) = E\{y^4 - 3 \left(E\{y^2\}\right)^2 \tag{21}$$

The values obtained using (21) and the whitening data were linearly combined and normalized. Singular value decomposition was performed to remove any correlation. Thereafter, the delta value was obtained as the difference between the dot product of the weight obtained from the singular value decomposition, the initial weight obtained randomly, and one. If the delta value exceeds 0.000001, the data obtained from the final weight and whitening are linearized, and these values are called independent components. The obtained independent components U and the linear image of the center image of the zero-padded original image were combined. A linear combination of the obtained independent components U and the center image of the zero-padded original image was obtained.

$$I^s i, l = U_i X_{padding} \tag{22}$$

s is the stage, and there are two stage. *l* refers to the independent elements. By subtracting each patch image from Equation (14), we obtained the following expression: For each input image $I_i \in R^{m \times n}$, we obtained L_1 output images $I_{i,l}^s = 1, 2, \ldots L_1, I_{i,l}^s \in R^{m \times n}$ after the first stage of ICANet1. We denoted I^s as,

$$I^s = [I_{1,1}^s \cdots I_{1,L_1}^s \cdots I_{N,1}^s \cdots I_{N,L_1}^s] \in R^{m \times NL_1 n} \tag{23}$$

Fig. 5 depicts the first stage of ICANet1, wherein the PCA and ICA are cascaded and a two-dimensional (2D) convolutional output is obtained. Fig. 6 depicts the feature CU-ECG database in the first stage of ICANet1. Fig. 7 depicts the second and output stages of ICANet1, and Fig. 8 depicts the feature CU-ECG database in the second and output stages of ICANet1.

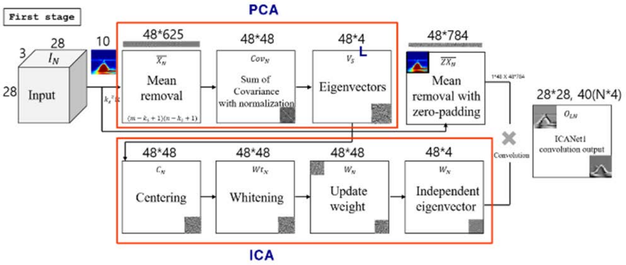


FIGURE 5. First stage of ICANet1.

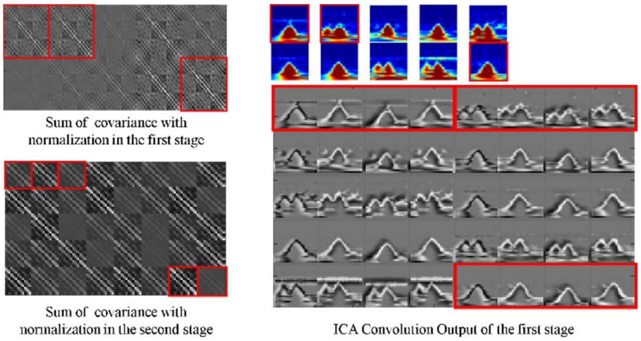


FIGURE 6. Feature CU-ECG database in first stage of ICANet1.

The detailed description of the workflow is given below.

(1) The center image was obtained in the PCA phase, and the eigenvector was obtained by normalizing the covariance.
(2) The centering and whitening processes in the ICA stage were performed, and we obtained an independent eigenvector using the updated weights.
(3) We obtained the center image from the basic data, and convolution was performed with the zero-padded image and the independent eigenvector obtained via ICANet1 in step (2), after which the ICA convolution output value was extracted.
(4) In the LBP step, the feature value was extracted using the hashing code, patch sliding, and histogram steps.
(5) Finally, the feature obtained in step (4) was vectorized, and classification was performed using the feature value.

B. ICANet2

ICANet2 is a modified version of the ICANet1 algorithm, and it extracts independent components by applying feature vectors obtained from PCA to ICA filters. We used the ICA algorithm to obtain a statistically independent basic image set

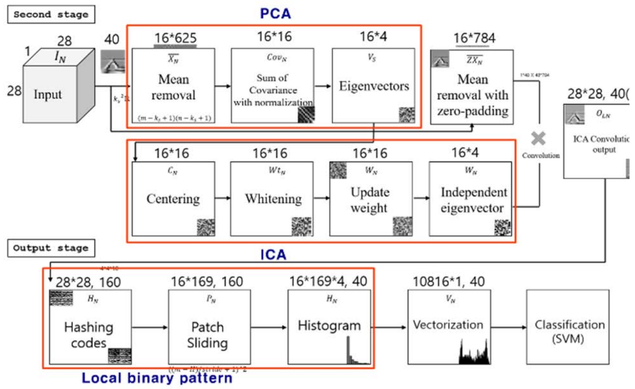


FIGURE 7. Second and output stages of ICANet1.

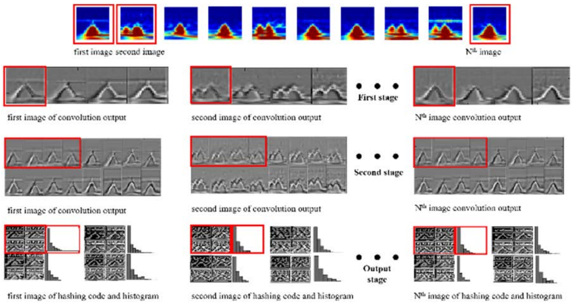


FIGURE 8. Feature CU-ECG database in second and output stages of ICANet1.

according to a set of ECGs. Because existing feature vectors are not statistically independent, it was the goal of ICANet2 to identify statistically independent feature vectors using ICA.

[Step 1] First patch sliding process.

Before sliding, the images were padded to $I_i' \in R^{(m+k_1-1)\times(n+k_2-1)}$, and the out-of-range input pixels were assumed to be zero. This ensured that all weights in the filters reach the entire area of the images. We used a patch of size $k_1 \times k_2$ to slide each pixel of the i^{th} image $I_i' \in R^{(m+k_1-1)\times(n+k_2-1)}$, and subsequently reshaped each $k_1 \times k_2$ matrix into a column vector, and then concatenated these vectors to obtain the following matrix:

$$X_i = [x_{i,1} x_{i,2} \cdots x_{i,l,mn}] \in R^{k_1 k_2 \times mn} \tag{24}$$

where $x_{i,j}$ denotes the j^{th} vectorized patch in I_i. Thus, for all input training images $I_i, i = 1, 2, \ldots, N$), we obtain the following matrix:

$$X_{i,l} = [X_{1,l}, X_{2,l} \cdots X_{N,l}] \in R^{k_1 k_2 \times Nmn} \tag{25}$$

$$X = [X_1, X_2 \cdots X_N] \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)N} \tag{26}$$

[Step 2] First mean removal process.

In this step, we subtracted the patch mean from each patch and obtained the following:

$$patc\,h\overline{X}_i = [\overline{X}_{i,1} \overline{X}_{i,2} \cdots \overline{X}_{i,mn}] \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)N} \tag{27}$$

Algorithm 1 ICANet1

```
1. Reading of image or signal data
2. Initialization of variables (patch size, number of filters,
   block size)
3. Stage 1 of ICANet1
   - Patch mean-removal:
       $\overline{X} = patc\,hX - patc\,h\overline{X}_i \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)}$
   - Compute Eigenvector using PCA:
       $v_i \times \overline{X}^T v_i', s.tv_i' = \overline{X}^T v_i$
   - Compute convolution using ICA:
       $U = W_{ICA} v_{i(PCA)}$
   - Output: $I^s i, l = U_i X_{padding}$
4. Stage 2 of ICANet1
   - Patch mean-removal:
       $\overline{Y} = patchY - patc\,h\overline{Y}_i \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)}$
   - Compute Eigenvector using PCA:
       $v_i \times \overline{Y}^T v_i', \quad s.t\ v_i' = \overline{Y}^T v_i$
   - Compute convolution using ICA:
       $U = W_{ICA} v_{i(PCA)}$
   - Output: $I^s i, l = U_i X_{padding}$
5. Output layer
   - Binary hashing: compute the decimal-valued image:
       $P_{i,l,l} = H\left(I^{II}_{i,l,\ell}\right),$
       $l = 1, 2, \ldots, L1; \ell = 1, 2, \ldots, L2; i = 1, 2, \ldots, N,$
   - Histogram:
       $f_1 = \begin{bmatrix} Hist\left(Z_{i,1,1}\right) \cdots Hist\left(Z_{i,1,B}\right) \cdots \\ \cdot Hist\left(Z_{i,L_1,1}\right) \cdots Hist\left(Z_{i,L_1,B}\right) \end{bmatrix}^T \in \mathbb{R}^{(2^{L_2})L_1}$
   - Output: $f = [f_1 \cdots f_N] \in \mathbb{R}^{(2^{L_2})L_1 BN}$
   -
```

By subtracting each patch image from (27), we obtained the following expression:

$$\overline{X} = patc\,hX - patc\,h\overline{X}_i \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)} \tag{28}$$

[Step 3] First PCA process with ICA

In this step, the covariance was obtained by repeating (28) for all patches and images.

$$\overline{X_{Cov}} = \sum\nolimits_{N=1}^{NL} (\overline{X} \times \overline{X}^T) \tag{29}$$

In the above equation, N is the number of images and L is the number of filters. The eigenvector can be obtained using the following equation:

$$\overline{X_{Cov}} \times \overline{X_{Cov}}^T v_i = \lambda_i v_i \tag{30}$$

$$\overline{X_{Cov}}^T \times \overline{X_{Cov}}(\overline{X_{Cov}}^T v_i) = \lambda_i(\overline{X_{Cov}}^T v_i) \tag{31}$$

$$v_i \times \overline{X}^T v_i', \quad s.t\ v_i' = \overline{X}^T v_i \tag{32}$$

$$V_i = \frac{\overline{X_{cov}}}{N} v_i \tag{33}$$

$$V_i = normalized(V_i) \tag{34}$$

$$F_i = \frac{\overline{X_{cov}}}{N} V_i \tag{35}$$

$$U = W_{ICA} FI_{i(PCA)} \tag{36}$$

$FI_i(pca)$ refers to the whitened process and centered feature vector. Let us assume that the eigenvector obtained using (36) is L. To apply the ICA algorithm for estimating the independent component, we applied it to the ICA based on the feature vector. First, ICA centering and whitening algorithms were applied, and data normalization was performed thereafter. Then, the initial weight was set randomly, and the initial weight and the data obtained from the whitening were linearly combined. The ICA was performed using Kurtosis and Negentropy, and we obtained the following expression:

$$kurt\ (y) = E\{y\}^4 - 3\left(E\{y^2\}\right)^2 \tag{37}$$

The values obtained from (37) and the whitening data were linearly combined and normalized. Singular value decomposition was performed to remove any correlation. Thereafter, the delta value was obtained as the difference between the dot product of the weight obtained from the singular value decomposition and the initial weight obtained randomly and one. If the delta value is greater than 0.000001, the data obtained from the final weight and whitening are linearized, and these values are called independent components. The obtained independent components U and the linear image of the center image of the zero-padded original image are combined. A linear combination of the obtained independent components U and the center image of the zero-padded original image is obtained as follows:

$$I^s i, l = U_i X_{padding} \tag{38}$$

s indicates the stage, and the first stage is one. l refers to independent elements one to l. By subtracting each patch image from (38), we obtain the following expression: For each input image $I_i \in R^{m\times n}$, we obtained L_1 output images $I^s_{i,l} = 1, 2, \ldots L_1, I^s_{i,l} \in R^{m\times n}$ after the first stage of ICANet2. Let us denote I^s as

$$I^s = [I^s_{1,1} \cdots I^s_{1,L_1} \cdots I^s_{N,1} \cdots I^s_{N,L_1}] \in R^{m\times NL_1 n} \tag{39}$$

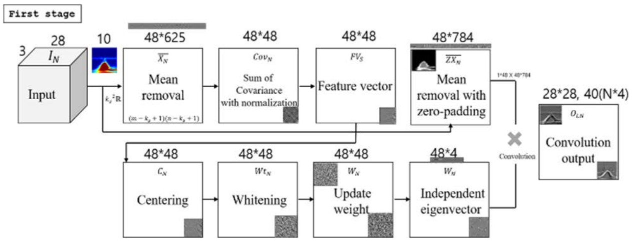


FIGURE 9. First stage of ICANet2.

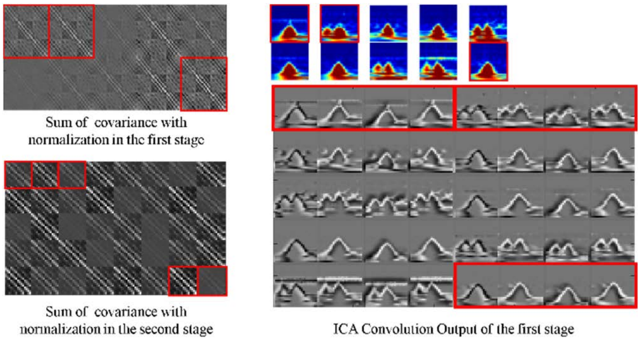


FIGURE 10. Feature CU-ECG database in first stage of ICANet2.

Fig. 9 shows the first stage of ICANet2, and the PCA and ICA are cascaded. In the first stage, a 2D convolutional output is obtained. Fig. 10 shows the feature CU-ECG database in the first stage of ICANet2. One covariance was generated for each image, and the second stage was affected by the number of filters to generate 40 covariance values ($L_1 L_2 = 4$), further resulting in 40 convolution outputs in the first stage.

[Step 4] Second patch sliding process.

By almost repeating the same process as in the first stage, the second stage of ICANet2 also includes three steps: similar to step 1, we used a patch of size $k_1 \times k_2$ to slide each pixel of the ith image $I'_{i,l} \in R^{k_1 k_2 \times mn}$, $l = 1, 2, \ldots, L_1$ and obtain a matrix as follows:

$$Y_i = [y_{i,l,1} y_{i,l,2} \cdots y_{i,l,mn}] \in R^{k_1 k_2 \times mn}$$
$$l = 1, 2, \ldots, L_1, i = 1, 2, \ldots, \tag{40}$$

where $y_{i,l,j}$ denote the jth vectorized patch in $I^l_{i,l}$. Thus, for all input training images $I_{i,l}, i = 1, 2, \ldots, N$), we obtain the following matrix:

$$Y = [X_{1,l}, Y_{2,l} \cdots Y_{N,l}] \in R^{k_1 k_2 \times Nmn} \tag{41}$$

We concatenated the matrices of all the filters and obtained a matrix,

$$Y = [Y_1, Y_2 \cdots Y_{L_1}] \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)N} \tag{42}$$

[Step 5] Second mean removal process.

In this step, we subtracted the patch mean from each patch and obtained the following:

$$patc\,h\overline{Y_i} = [\overline{Y}_{i,1} \overline{Y}_{i,2} \cdots \overline{Y}_{i,mn}] \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)N} \tag{43}$$

By subtracting each patch image from (43), we obtain the following expression:

$$\overline{X} = patc\,hY - patc\,h\overline{Y}_i \in R^{k_s^2\times(m-k_s+1)(n-k_s+1)} \tag{44}$$

[Step 6] Second PCA process with ICA

In this step, the covariance was obtained by repeating (44) for all the patches and images.

$$\overline{X_{Cov}} = \sum\nolimits_{N=1}^{NL}(\overline{X}\times\overline{X}^T) \tag{45}$$

where N is the number of images and L is the number of filters. The eigenvector can be obtained using the following equation:

$$\overline{X_{Cov}}\times\overline{X_{Cov}}^T v_i = \lambda_i v_i \tag{46}$$

$$\overline{X_{Cov}}^T \times\overline{X_{Cov}}(\overline{X_{Cov}}^T v_i) = \lambda_i(\overline{X_{Cov}}^T v_i) \tag{47}$$

$$v_i\times\overline{X}^T v_i', \quad s.t\ v_i' = \overline{X}^T v_i \tag{48}$$

$$V_i = \frac{\overline{X_{cov}}}{N} v_i \tag{49}$$

$$V_i = normalized(V_i) \tag{50}$$

$$F_i = \frac{\overline{X_{cov}}}{N} V_i \tag{51}$$

$$U = W_{ICA}FI_{i(PCA)} \tag{52}$$

A linear combination of the obtained independent components U and the center image of the zero-padded original image is obtained as follows:

$$I^s i, l = U_i X_{padding} \tag{53}$$

where s represents the stage, and the first stage is one. l refers to independent elements one to l. By subtracting each patch image from (53), we obtained the following expression: For each input image $I_i^s \in R^{m\times n}$, we obtained L_2 output images $I_{i,l,\iota}^{ss}, \iota = 1, 2, \ldots, L_2, I_{i,l,\iota}^{ss} \in R^{m\times n}$ after the second stage of ICANet2. Thus, we obtained NL_1L_2 images $I_{i,l,\iota}^{ss}, l = 1, 2, \ldots L_2, \iota = 1, 2, \ldots, L_2, i = 1, 2, \ldots, N, I_{i,l,\iota}^{ss} \in R^{m\times n}$ after the first and second stages. Let us denote I^s as

$$I^{ss} = [I_{1,1,1}^{ss}\cdots I_{1,1,L_2}^{ss}\cdots I_{1,L_1,1}^{ss}\cdots I_{1,L_1,L_2}^{ss} \\ \cdots I_{N,1,1}^{ss}\cdots I_{N,1,L_2}^{ss}\cdots I_{N,L_1,1}^{ss}\cdots I_{N,L_1,L_2}^{ss}] \tag{54}$$

The second stage processed the data for each image by updating the eigenvectors and independent eigenvectors accordingly. Therefore, the feature values of the output stage exhibited different feature values. Fig. 11 shows the CU-ECG database in the second stage of ICANet2, and Fig. 12 shows the feature CU-ECG database in the second and output stages of ICANet2. The number of eigenvectors increased based on the number of filters (L$_2$ = 4), and four feature values were extracted from one image.

The detailed description of the workflow is as given below:

(1) Perform the PCA step in the same manner as in ICANet1.

(2) Patch sliding of all images was performed, and the average image and eigenvector obtained from the PCA were convoluted.

(3) The eigenvectors are updated through the normalized eigenvectors, and then the convolution is performed using the average image determined using all the images and updated eigenvectors. This is called a feature vector.

(4) The center image is obtained from the basic data. The zero-padded image and the independent eigenfeature vector obtained in step (2) are convoluted, and the ICA convolution output value is extracted. Steps (5) and (6) are the same as those in ICANet1.

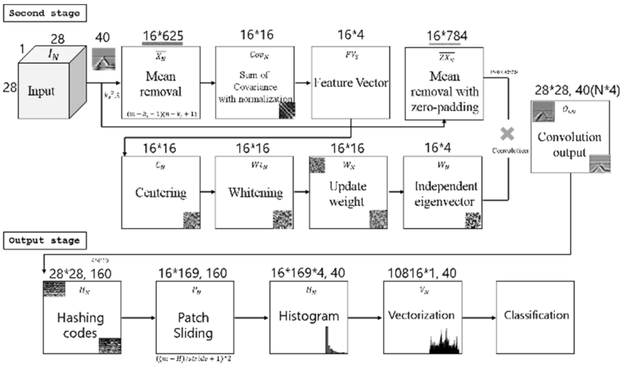
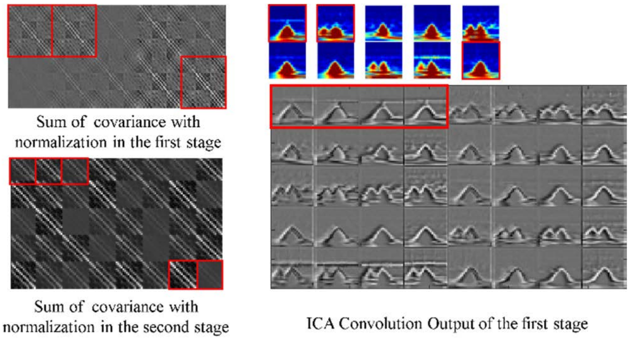


FIGURE 11. Second and output stages of ICANet2.


Sum of covariance with normalization in the first stage

Sum of covariance with normalization in the second stage

ICA Convolution Output of the first stage

FIGURE 12. Feature CU-ECG database in second and output stages of ICANet2.

[Step 7] Binary quantization.

In this step, we binarized the outputs of the second stage of PCANet to obtain:

$$P_{i,l,l} = H\left(I_{i,l,l}^{II}\right), l = 1, 2, \ldots, L1; \\ l = 1, 2, \ldots, L2; \quad i = 1, 2, \ldots, N, \tag{55}$$

where $H(\cdot)$ is a Heaviside step function whose value equals one for positive entries and zero in all other cases. Let us denote P as,

$$P = P_{1,1,1}\cdots P_{1,1,L_2}\cdots P_{1,L_1,1}\cdots P_{1,L_1,L_2} \\ \cdots P_{N,1,1}\cdots P_{N,1,L_2}\cdots P_{N,L_1,1}\cdots P_{N,L_1,L_2} \tag{56}$$

[Step 8] Weight and sum.

Around each pixel, we can see the vector of L_2binary bits as a decimal number. This converts the binary images $P_{i,l,l}$ back into integer-valued images as follows.

$$T_{i,l} = \sum_{l=1}^{L_2} 2^{l-1} \tag{57}$$

Let us denote T as follows.

$$T = [T_{1,1} \cdots T_{1,L_1} \cdots T_{N,1} \cdots T_{N,L_1}] \in \mathbb{R}^{m\times NLL_1 n} \tag{58}$$

[Step 9] Block sliding.

We used a block of size $h_1 \times h_2$ to slide each of the L_1 images $T_{i,l}, l = 1, \ldots, L_1$, with overlap ratio R, and subsequently reshaped each $h_1 \times h_2$ matrix into a column vector, which was then concatenated to obtain a matrix as follows:

$$Z_{i,l} = [z_{i,l,1} z_{i,l,2} \cdots z_{i,l,B}] \in \mathbb{R}^{h_1 h_2 \times B},\ i = 1, 2, \ldots, N \tag{59}$$

Let us denote T as follows:

$$T = [T_{1,1} \cdots T_{1,L_1} \cdots T_{N,1} \cdots T_{N,L_1}] \in \mathbb{R}^{m\times NLL_1 n} \tag{60}$$

where $z_{i,l,j}$ denotes the jth vectorized patch in $T_{i,l}, l = 1, \ldots, .B$ represents the number of blocks when using a block of size $h_1 \times h_2$ to slide each $T_{i,l}, l = 1, \ldots, L_1$, with overlap ratio R, and can be expressed as

$$B = [1 + (m + k_1 - 1 - h_1)/stride1] \; [1 + (n + k_2 - 1 - h_2)/stride2] \tag{61}$$

where strides 1 and 2 are the vertical and horizontal steps, respectively, and *round* refers to the rounding off function.

$$\begin{aligned} stride1 &= \partial\left((1-R) \times h_1\right), \\ stride2 &= \partial((1-R) \times h_2) \end{aligned} \tag{62}$$

$\partial(.)$ means to round off. As shown in (62), the number of blocks B increases as the overlap ratio R increases. For L_1 images, we concatenated $Z_{i,l}$ to obtain the following matrix:

$$Z_i = \left[Z_{i,1} Z_{i,2} \cdots Z_{i,B}\right] \in \mathbb{R}^{h_1 h_2 \times L_1 B},\ i = 1, 2, \ldots, N \tag{63}$$

Let us denote Z as follows.

$$Z = [Z_{1,1} \cdots Z_{1,B} \cdots Z_{N,1} \cdots Z_{N,B}] \in \mathbb{R}^{h_1 h_2 \times L_1 BN} \tag{64}$$

[Step 10] Histogram.

We computed the histogram (with 2^{L_2} bins) of the decimal values in each column of Z_i and concatenated all the histograms into one vector to obtain

$$f_1 = \begin{bmatrix} Hist\left(Z_{i,1,1}\right) \cdots Hist\left(Z_{i,1,B}\right) \cdot\cdot \\ \cdot Hist\left(Z_{i,L_1,1}\right) \cdots Hist\left(Z_{i,L_1,B}\right) \end{bmatrix}^T \in \mathbb{R}^{(2^{L_2})L_1} \tag{65}$$

which is the "feature" of input image I_1, and *Hist* (.) denotes the histogram operation. Let us denote f as,

$$f = [f_1 \cdots f_N] \in \mathbb{R}^{(2^{L_2})L_1 BN} \tag{66}$$

The feature vector is then sent to a classifier, such as an SVM. Fig. 13 shows the hashing codes in the output stage of the ICANet2.

$$H(x) = \begin{cases} 1, & x > 0 \\ \frac{1}{2}, & x = 0 \\ 0, & x < 0 \end{cases}$$

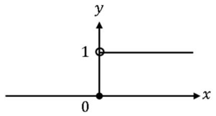


FIGURE 13. Hashing codes in the output stage of ICANet2.

The hashing code is similar to that of the LBPs. It is a binary algorithm that codes zero if the feature value is less than zero and one if the feature value is greater than zero, as shown in Fig. 13. Therefore, the independence component obtained from ICA is simplified to zero and one.

≪Advantage of ICANet≫

I. The network structure is simple and computationally efficient.

II. The ICA filter is trained with an unsupervised algorithm using unlabeled samples, which is practical.

III. Compared to deep learning models, each layer parameter in ICANet can be easily trained.

IV. EXPERIMENT AND RESULTS

This section describes the data acquisition process and environment, evaluates the data. In addition, we presented the performance assessment and effectiveness of ICANet.

A. CU-ECG DATABASE

We obtained the CU-ECG database from Chosun University, Korea. We recruited many volunteers to acquire and collate the data. The dataset contained 95 participants. The duration of each measurement was 10 s, and there were 60 measurements for each participant over a three-day period. The participants were allowed to sit on a chair in a relaxed state, and the measurements were obtained with a sampling rate of 500 Hz. All acquired ECGs were of the Lead1 type that were obtained using wet electrodes. We used a processor, amplifier, bandpass filter, and low-pass filter as the primary board and sensor. An Atmega8 processor was used, and the analog-to-digital converter had a 10-bit resolution. We used a USB-to-serial cable to establish a communication link. The gain of the amplifier was 1000, and the signal was measured using a +5 V power source. To obtain the ECG data, the patient was allowed to sit on a chair in a relaxed state. The patch depicted in Figure 14 was attached to the arm, and the data were acquired. The acquired data were stored on a PC using universal synchronous/asynchronous receiver/transmitter (USART) communication. Fig. 14 depicts the data acquisition environment for ECG biometrics, and Fig. 15 shows the preprocessing using the CU-ECG database. Fig. 16 shows

Algorithm 2 ICANet2

```
1. Reading of image or signal data
2. Initialization of variables (patch size, number of filters,
   block size)
3. Stage 1 of ICANet2
   - Patch mean-removal:
       $\overline{X} = patc\,hX - patc\,h\overline{X}_i \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)}$
   - Compute PCA feature vector using PCA:
       $V_i = \frac{\overline{X_{cov}}}{N} v_i, \quad V_i = normalized\,(V_i)\,, F_i = \frac{\overline{X_{cov}}}{N} V_i$
   - Compute convolution using ICA:
       $U = W_{ICA} FI_i(pca)$
   - Output: $I^s i, l = U_i X_{padding}$
4. Stage 2 of ICANet2
   - Patch mean-removal:
       $\overline{Y} = patc\,hY - patc\,h\overline{Y}_i \in R^{k_s^2 \times (m-k_s+1)(n-k_s+1)}$
   - Compute PCA feature vector using PCA:
       $V_i = \frac{\overline{X_{cov}}}{N} v_i, \quad V_i = normalized\,(V_i)\,, F_i = \frac{\overline{X_{cov}}}{N} V_i$
   - Compute convolution using ICA:
       $U = W_{ICA} v_{i(PCA)}$
   - Output: $I^s i, l = U_i X_{padding}$
5. Output layer
   - Binary hashing: compute the decimal-valued image:
       $P_{i,l,l} = H\left(I^{II}_{i,l,\ell}\right),$
       $l = 1, 2, \ldots, L1;\ \ell = 1, 2, \ldots, L2;\ i = 1, 2, \ldots, N,$
   - Histogram:
       $f_1 = \begin{bmatrix} Hist\left(Z_{i,1,1}\right) \cdots Hist\left(Z_{i,1,B}\right) \cdot\cdot \\ \cdot Hist\left(Z_{i,L_1,1}\right) \cdots Hist\left(Z_{i,L_1,B}\right) \end{bmatrix}^T \in \mathbb{R}^{(2^{L_2})L_1}$
   - Output: $f = [f_1 \cdots f_N] \in \mathbb{R}^{(2^{L_2})L_1 BN}$
   -
```

the samples of the CU-ECG database with a noise CU-ECG, whereas Fig. 17 shows samples of the CU-ECG database with a size scale. Noise (white Gaussian) was used to determine the robustness of ICA. In addition, additive white Gaussian noise was a basic noise model used in information theory to mimic the effects of several random processes occurring in nature. The database was divided into 90% and 10% for training and testing, respectively.

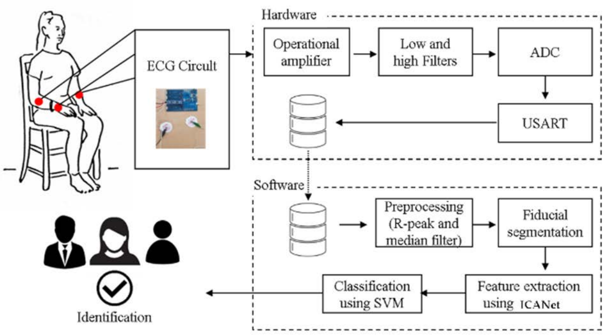
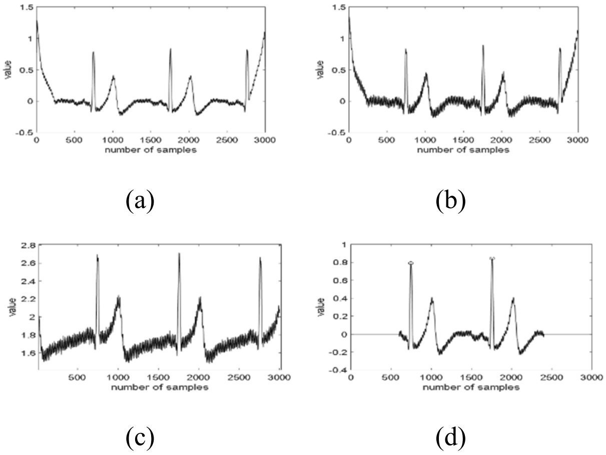


FIGURE 14. Data acquisition environment for ECG biometrics.



(a)

(b)

(c)

(d)

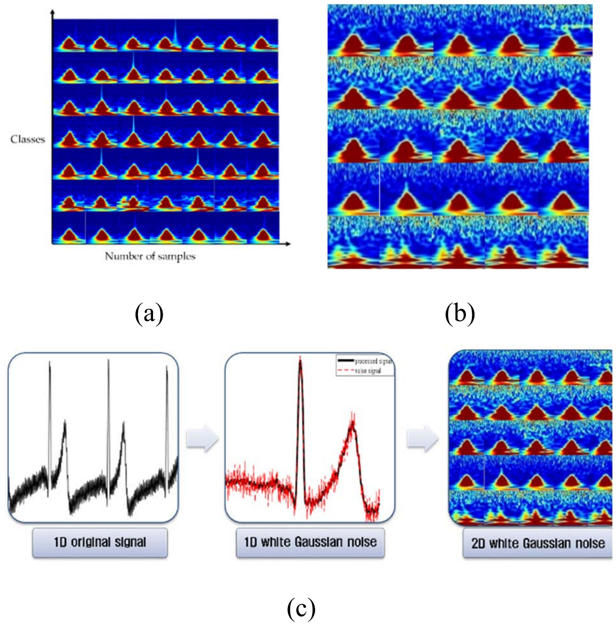
FIGURE 15. Preprocessing using CU-ECG database: (a) raw signal; (b) mean variance; (c) spike removal; (d) R peak detection.

B. MIT-BIH ECG DATABASE

The MIT-BIH ECG database includes 48 parts that contain two-channel ECG recordings. These parts were recorded between 1975 and 1979 at Boston's Beth Israel Hospital (now the Beth Israel Deaconess Medical Center). The MIT-BIH ECG database is consist of 47 participants. This database was acquired in approximately 30 minutes. The participants comprised 25 men and 22 women, with age-ranges of 32 to 89 years and 23 to 89 years, respectively. The sampling rate was 360 samples per second, and the resolution for digitization was 11-bit over a 10-mV range. Twenty sets of data were obtained for each class. The size of the training data was 940 × 1600, and both lead types I and II were used. Fig. 18 shows an example of the MIT ECG data with a scalogram.

C. PERFORMANCE EVALUATION AND SETTING PARAMETERS WITH CLASSIFIER

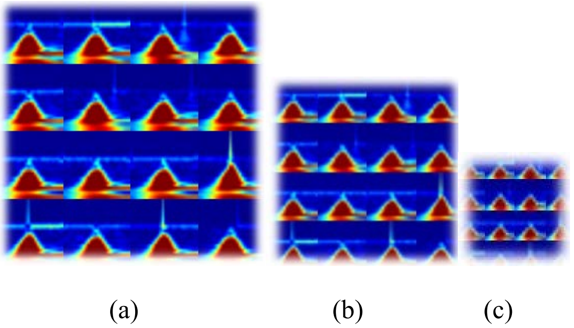
True positives (TPs) and true negatives (TNs)are the number of correct predictions made for the positive and negative



(a) (b)

(c)

FIGURE 16. Samples of CU-ECG database with noise CU-ECG: (a) ECG scalogram; (b) salt noise with ECG scalogram; (c) noise signals.



(a) (b) (c)

FIGURE 17. Samples of CU-ECG database with size scale: (a) 28 × 28; (b) 20 × 20; (c) 12 × 12.

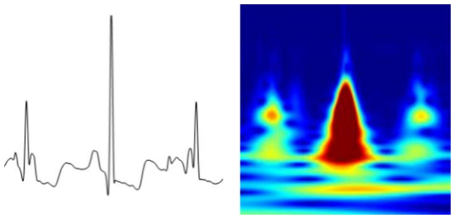
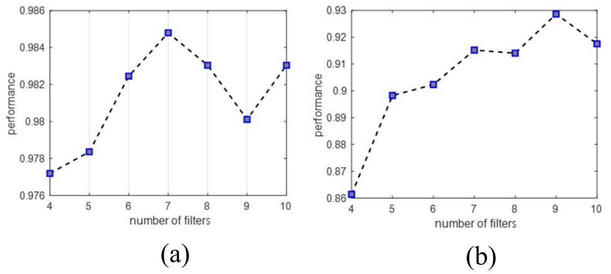


FIGURE 18. Example of MIT ECG data with scalogram.

samples, respectively, and false negatives (FNs) and false positives (FPs) are the number of incorrect predictions made for the positive and negative samples, respectively.

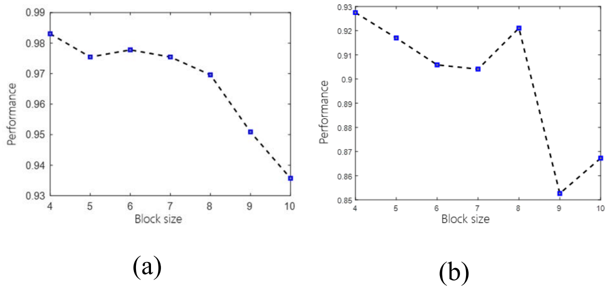
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{67}$$

D. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS



(a) (b)

FIGURE 19. Effect on the number of filters in ICANet. (a) CU-ECG database. (b) noise CU-ECG database.



(a) (b)

FIGURE 20. Effect on the number of filters in ICANet. (a) CU-ECG database. (b) noise CU-ECG database.

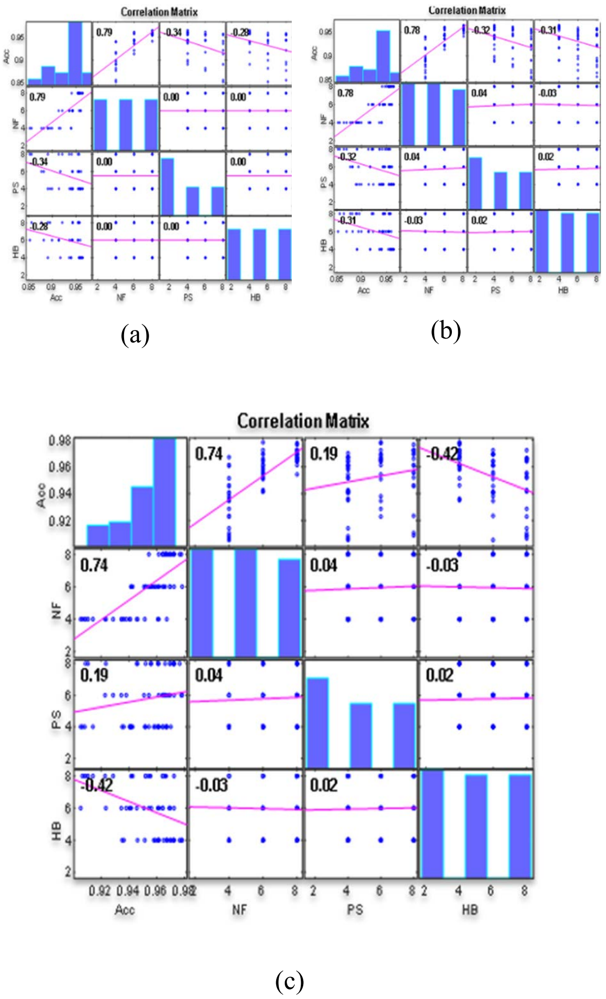
To demonstrate the performance of the proposed algorithm, we used various classifiers. The feature extractors were PCANet, ICANet1, and ICANet2. The performance obtained using CU-ECG data showed a performance similar to that of PCANet without noise data. However, as the data changed shape, they showed a higher performance in ICANet2 than in PCANet. If the training data and testing data are similar, the PCANet can have a good effect, but if non-similar real-life data are used, the ICANet2 algorithm becomes robust. In particular, when noise ECG data and scale data are used, the robustness of the ICANet2 algorithm can be verified. In addition, the classifier confirmed that the SVM algorithm worked well, and the verification time was important for personal ECG authentication. The verification time of PCANet was approximately 0.2 s, and that of ICANet2 was 0.15 s, which is approximately equal to that of PCANet. Therefore, the effectiveness of the ICANet algorithm for personal authentication with respect to the classification time was demonstrated.

1) EFFECT ON THE NUMBER OF FILTERS

The number of filters is an important factor, which determines the recognition performance in the convolution layer on ICANet because the amount of information starts increasing when the number of filters in the first stage gradually increases. Fig. 19 shows the effect of the number of filters on ICANet performance.

2) EFFECT ON BLOCK SIZE

Unlike in the case of an increase in the number of filters, the performance tends to decrease as the block size increases. As the block size increases, the spacing between the strides



(a) (b)

(c)

FIGURE 21. Correlation of PCANet, ICANet1 (a) PCANet. (b) ICANet1. (c) ICANet2.

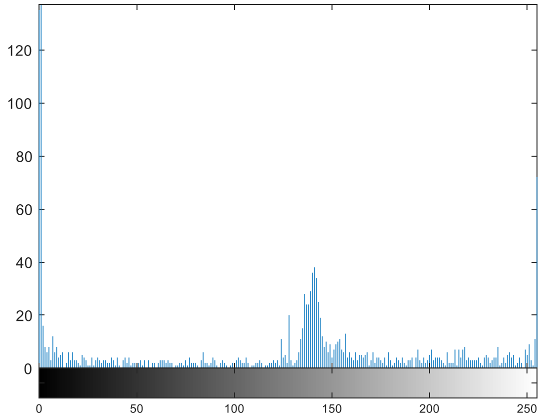


FIGURE 22. Histogram of sample CU-ECG.

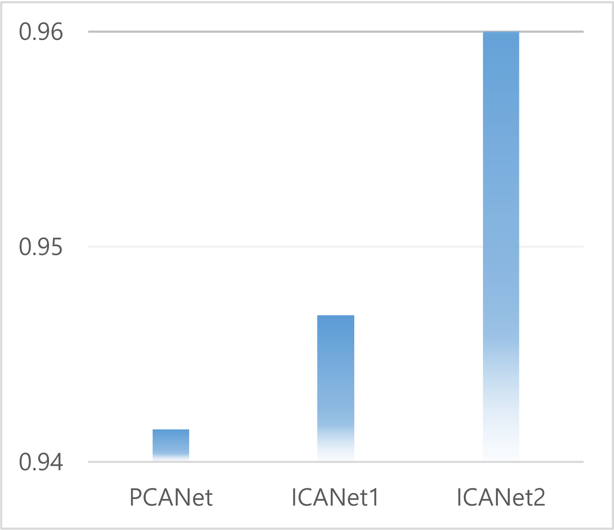


FIGURE 23. Performance of MIT BIH-ECG.

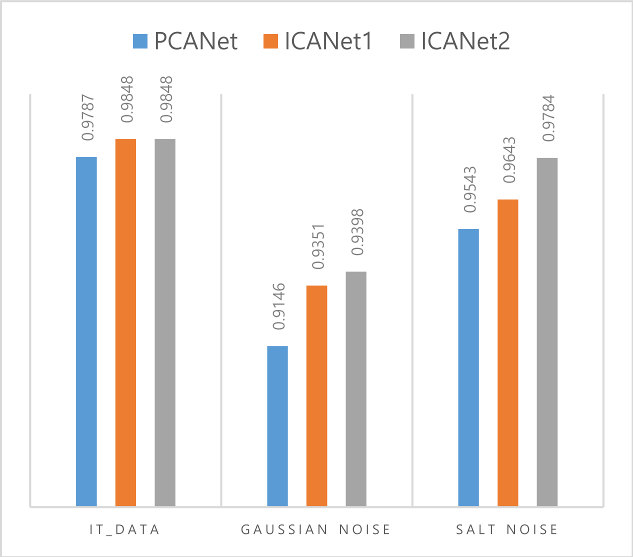


FIGURE 24. Performance of CU-ECG with noise by SVM.

TABLE 1. Performance of CU-ECG in terms of classifier.

Algorithms	ACC
Decision Tree	0.8320
ELM	0.8989
AE	0.9324
ANN	0.9820
KNN	0.9650
Naive Bayes	0.9630
SVM	**0.9848**

becomes wider, resulting in a loss of data and degradation of performance. Fig. 20 shows the effect of the ICANet block size on performance, and Fig. 21 shows the correlation between PCANet and ICANet. Fig. 22 shows the histogram of sample CU-ECG; Fig. 23 indicates the performance of MIT BIH-ECG. Fig. 24 shows the performance of the CU-ECG with noise. Fig. 25 shows the performance of the scale CU-ECG, and Table 1 lists the performance of the CU-ECG in terms of the classifier. Fig. 25 Confusion Matrix of

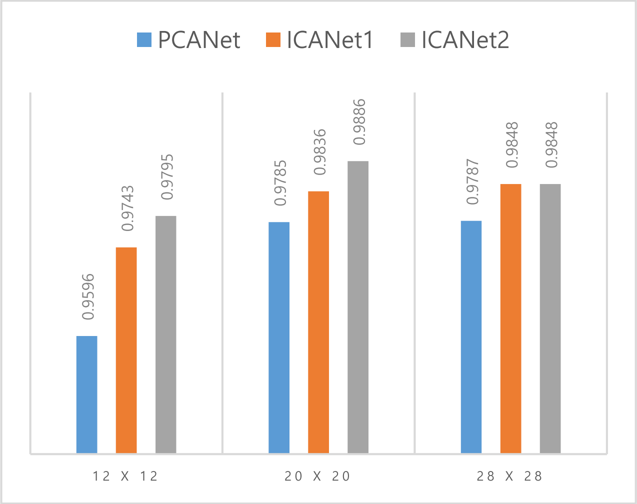


FIGURE 25. Performance of scale CU-ECG.

TABLE 2. Performance of CU-ECG (28*28) with SVM classifier.

Methods	SVM
F-measure	0.9848
Cohen's kappa	0.9846
kappa error	0.0030
kappa C.I.	0.9788 - 0.9905
Precision	0.9848
sensitivity	0.9848
specificity	0.9998

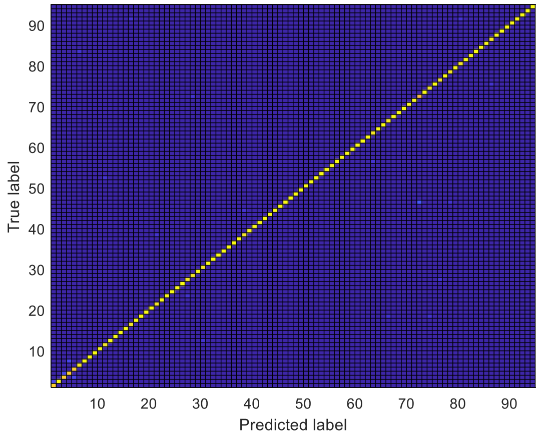


FIGURE 26. Confusion matrix of CU-ECG with SVM.

CU-ECG with SVM and Table 2 lists performance of CU-ECG (28*28) with SVM classifier.

V. CONCLUSION

In this study, we developed the ICANet2 algorithm to overcome the disadvantages of PCANet. The experimental results demonstrated that the PCANet algorithm had a lower performance when using noisy data and small feature data. Generally, when processed data are used, the performance is similar to that of PCANet. However, the PCANet that used eigenvectors was weak against noise, and the feature vector that used statistical independence was robust to noise. In addition, ICANet2 exhibited robustness compared to PCANet, even with the loss of data. Furthermore, the performance was better than that of the basic deep learning of auto-encoders, and the classification time was approximately 0.05 s faster than that of PCANet.

REFERENCES

[1] H. Nguyen, L.-M. Kieu, T. Wen, and C. Cai, "Deep learning methods in transportation domain: A review," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 998–1004, Nov. 2018, doi: 10.1049/iet-its.2018.0064.

[2] R. Mu, "A survey of recommender systems based on deep learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018, doi: 10.1109/ACCESS.2018.2880197.

[3] F. Celesti, A. Celesti, J. Wan, and M. Villari, "Why deep learning is changing the way to approach NGS data processing: A review," *IEEE Rev. Biomed. Eng.*, vol. 11, pp. 68–76, 2018, doi: 10.1109/RBME.2018.2825987.

[4] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014, doi: 10.1109/ACCESS.2014.2325029.

[5] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, Mar. 2015, doi: 10.1186/s40537-014-0007-7.

[6] P. Angelov and A. Sperduti, "Challenges in deep learning," in *Proc. Eur. Symp. Artif. Neural Netw.*, no. 1, 2016, pp. 489–496.

[7] I. T. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics), vol. 98, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.

[8] N. B. Ngoc, "Human detection in video using poselet with articulated pose estimation and edge-based RPCA foreground extraction," *J. Korean Inst. Inf. Technol.*, vol. 12, no. 3, pp. 51–60, Mar. 2014, doi: 10.14801/KIITR.2014.12.3.51.

[9] Y. Byeon, W. Lim, J. Lee, H. Jeong, H. Han, and C. Kwak, "Individual identification on electrocardiogram using third-order tensor-based MPCA," *J. Adv. Eng. Technol.*, vol. 11, no. 2, pp. 151–157, Jun. 2018.

[10] M. S. Park and S. Oh, "Nonlinear feature extraction using class-augmented kernel PCA," *J. Inst. Electron. Eng. Korea*, vol. 48, no. 5, pp. 7–12, Sep. 2011.

[11] B.-Y. Kim, S.-K. Oh, and J.-Y. Kim, "Design of digit recognition system realized with the aid of fuzzy RBFNNs and incremental-PCA," *J. Korean Inst. Intell. Syst.*, vol. 26, no. 1, pp. 56–63, Feb. 2016.

[12] M. Jin, R. Li, J. Jiang, and B. Qin, "Extracting contrast-filled vessels in X-ray angiography by graduated RPCA with motion coherency constraint," *Pattern Recognit.*, vol. 63, pp. 653–666, Mar. 2017, doi: 10.1016/j.patcog.2016.09.042.

[13] Y. Lu, Z. Lai, Z. Fan, J. Cui, and Q. Zhu, "Manifold discriminant regression learning for image classification," *Neurocomputing*, vol. 166, pp. 475–486, Oct. 2015, doi: 10.1016/j.neucom.2015.03.031.

[14] Y. Li, G. Liu, Q. Liu, Y. Sun, and S. Chen, "Moving object detection via segmentation and saliency constrained RPCA," *Neurocomputing*, vol. 323, pp. 352–362, Jan. 2019, doi: 10.1016/j.neucom.2018.10.012.

[15] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018, doi: 10.1109/MSP.2018.2826566.

[16] J. X. Liu, Y. Xu, C. H. Zheng, H. Kong, and Z. H. Lai, "RPCA-based tumor classification using gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 4, pp. 964–970, Jul. 2015, doi: 10.1109/TCBB.2014.2383375.

[17] W. Sun and Q. Du, "Graph-regularized fast and robust principal component analysis for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3185–3195, Jun. 2018, doi: 10.1109/TGRS.2018.2794443.

[18] F. Zhong, L. Liu, and J. Hu, "Robust 2DLDA based on correntropy," *Neurocomputing*, vol. 316, pp. 399–404, Nov. 2018, doi: 10.1016/j.neucom.2018.08.026.

[19] M. Yin, D. Zeng, J. Gao, Z. Wu, and S. Xie, "Robust multinomial logistic regression based on RPCA," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1144–1154, Dec. 2018, doi: 10.1109/JSTSP.2018.2872460.

[20] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput.*, vol. 18, pp. 178–184, May 2014, doi: 10.1016/j.asoc.2014.01.028.

[21] J. Liu, C. Zhang, and C. Zheng, "EEG-based estimation of mental fatigue by using KPCA–HMM and complexity parameters," *Biomed. Signal Process. Control*, vol. 5, no. 2, pp. 124–130, Apr. 2010, doi: 10.1016/j.bspc.2010.01.001.

[22] A. Vinay, V. S. Shekhar, K. N. B. Murthy, and S. Natarajan, "Face recognition using Gabor wavelet features with PCA and KPCA—A comparative study," *Proc. Comput. Sci.*, vol. 57, pp. 650–659, 2015, doi: 10.1016/j.procs.2015.07.434.

[23] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," in *Proc. Comput. Vis. Pattern Recognit.*, Aug. 2012, vol. 21, no. 10, pp. 1–9.

[24] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016, doi: 10.1109/TGRS.2015.2478379.

[25] J. Xia, N. Falco, J. A. Benediktsson, P. Du, and J. Chanussot, "Hyperspectral image classification with rotation random forest via KPCA," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1601–1609, Apr. 2017, doi: 10.1109/JSTARS.2016.2636877.

[26] F. Yuan, X. Xia, J. Shi, H. Li, and G. Li, "Non-linear dimensionality reduction and Gaussian process based classification method for smoke detection," *IEEE Access*, vol. 5, pp. 6833–6841, 2017, doi: 10.1109/ACCESS.2017.2697408.

[27] K. Hotta, "Local co-occurrence features in subspace obtained by KPCA of local blob visual words for scene classification," *Pattern Recognit.*, vol. 45, no. 10, pp. 3687–3694, Oct. 2012.

[28] Y. Xiao, H. Wang, W. Xu, and J. Zhou, "L1 norm based KPCA for novelty detection," *Pattern Recognit.*, vol. 46, no. 1, pp. 389–396, Jan. 2013.

[29] S.-J. Hong and Y.-H. Cho, "Image recognition by using hybrid coefficient measure of correlation and distance," *J. Korean Inst. Intell. Syst.*, vol. 20, no. 3, pp. 343–347, Jun. 2010, doi: 10.5391/JKIIS.2010.20.3.343.

[30] J. S. Hwang and J. H. Kim, "A study on the estimation of modal parameters using independent component analysis method," *J. Architectural Inst. Korea Struct. Construct.*, vol. 27, pp. 27–35, May 2011.

[31] S. C. Baek and J. S. Hong, "Image classification based on multi-resolution characteristic and independent component analysis," in *Proc. KIIT Conf.*, Nov. 2012, pp. 455–460.

[32] K.-H. Kwon and J.-D. Lim, "Dried pepper sorting using independent component analysis on RGB images," *J. Korea Soc. Comput. Inf.*, vol. 17, no. 4, pp. 59–65, Apr. 2012.

[33] K. S. Kang, H.-J. Kim, and J.-S. Hwang, "Mode decomposition and output-only system identification of lateral-torsionally coupled structures using independent component analysis method," *J. Architectural Inst. Korea Struct. Construct.*, vol. 28, no. 2, pp. 37–45, Feb. 2012.

[34] K.-H. Kim, "Vibration source contribution analysis of plate structure using independent component analysis," *J. Ocean Eng. Technol.*, vol. 26, no. 4, pp. 70–76, Aug. 2012.

[35] K. Kim, H.-M. Kwon, D.-S. Cho, J.-H. Kim, and J.-J. Jun, "Vibration source signal identification of structures using ICA," *J. Soc. Nav. Architects Korea*, vol. 49, no. 6, pp. 498–503, Dec. 2012.

[36] X. Quan and K. Bae, "Frequency bin alignment using covariance of power ratio of separated signals in multi-channel FD-ICA," *Phonetics Speech Sci.*, vol. 6, no. 3, pp. 149–153, Sep. 2014.

[37] C. G. Park, "Blind source separation using ICA and masking operation in time-frequency domain," *Korea Inst. Inf. Technol. Rev.*, vol. 11, no. 10, pp. 35–43, Oct. 2013.

[38] J.-S. Hwang, "A novel mode decomposition method for non-classical damping structure using acceleration responses," *J. Architectural Inst. Korea Struct. Construct.*, vol. 31, no. 1, pp. 19–26, Jan. 2015.

[39] J.-W. Lee and B.-R. Lee, "Design of filter to remove motionartifacts of photoplethysmography based on indepenent components analysis and filter banks," *J. Korea Inst. Inf. Commun. Eng.*, vol. 20, no. 8, pp. 1431–1437, Aug. 2016.

[40] C. Symposium, "Designed of partial discharge pattern classifier of independent component analysis based fuzzy neural networks," *Korean Inst. Elect. Eng.*, vol. 2017, no. 10, pp. 142–143, Oct. 2017.

[41] M. Wu, J. Zhou, and J. Sun, "Multi-scale ICA texture pattern for gender recognition," *Electron. Lett.*, vol. 48, no. 11, p. 629, 2012, doi: 10.1049/el.2012.0834.

[42] Z. L. Sun, C. H. Zheng, Q. W. Gao, J. Zhang, and D. X. Zhang, "Tumor classification using eigengene-based classifier committee learning algorithm," *IEEE Signal Process. Lett.*, vol. 19, no. 8, pp. 455–458, Aug. 2012, doi: 10.1109/LSP.2012.2202317.

[43] R. J. Martis, U. R. Acharya, and L. C. Min, "ECG beat classification using PCA, LDA, ICA and discrete wavelet transform," *Biomed. Signal Process. Control*, vol. 8, no. 5, pp. 437–448, 2013.

[44] X. Wang and X.-P. Zhang, "An ICA mixture hidden conditional random field model for video event classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 46–59, Jan. 2013, doi: 10.1109/TCSVT.2012.2203195.

[45] C. Yu, Q. Qiu, Y. Zhao, and X. Chen, "Satellite image classification using morphological component analysis of texture and cartoon layers," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1109–1113, Sep. 2013, doi: 10.1109/LGRS.2012.2230612.

[46] N. Falco, J. A. Benediktsson, and L. Bruzzone, "A study on the effectiveness of different independent component analysis algorithms for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2183–2199, Jun. 2014, doi: 10.1109/JSTARS.2014.2329792.

[47] A. X. Stewart, A. Nuthmann, and G. Sanguinetti, "Single-trial classification of EEG in a visual object task using ICA and machine learning," *J. Neurosci. Methods*, vol. 228, no. 1, pp. 1–14, May 2014, doi: 10.1016/j.jneumeth.2014.02.014.

[48] Y. Xiao, Z. Zhu, Y. Zhao, Y. Wei, and S. Wei, "Kernel reconstruction ICA for sparse representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1222–1232, Jun. 2015, doi: 10.1109/TNNLS.2014.2334711.

JAE-NEUNG LEE received the B.S. and M.S. degrees from the Department of Control and Instrumentation Engineering, Chosun University, Gwangju, South Korea, in 2013 and 2015, respectively. He is currently a Postdoctoral Fellow with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, China.

KEUN-CHANG KWAK (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Electronics Engineering, Chungbuk National University, Cheongju, South Korea, in 1996, 1998, and 2002, respectively. From 2003 to 2005, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He was a Senior Researcher with the Human–Robot Interaction Team, Intelligent Robot Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, from 2005 to 2007. He was also a Visiting Professor with the Department of Computer Science, California State University, Fullerton, CA, USA, from 2014 to 2015. He is currently a Professor with the Department of Electronics Engineering, Chosun University, Gwangju, South Korea. His current research interests include computational intelligence, human–robot interaction, and biometrics.