



# Convolutional Neural Networks for Electrocardiogram Classification

Mohamad M. Al Rahhal<sup>1</sup> · Yakoub Bazi<sup>2</sup> · Mansour Al Zuair<sup>2</sup> · Esam Othman<sup>2</sup> · Bilel BenJdira<sup>3,4</sup>

Received: 27 February 2017 / Accepted: 12 February 2018 / Published online: 30 March 2018  
© Taiwanese Society of Biomedical Engineering 2018

## Abstract

In this paper, we propose a transfer learning approach for Arrhythmia Detection and Classification in Cross ECG Databases. This approach relies on a deep convolutional neural network (CNN) pretrained on an auxiliary domain (called ImageNet) with very large labelled images coupled with an additional network composed of fully connected layers. As the pretrained CNN accepts only RGB images as the input, we apply continuous wavelet transform (CWT) to the ECG signals under analysis to generate an over-complete time–frequency representation. Then, we feed the resulting image-like representations as inputs into the pretrained CNN to generate the CNN features. Next, we train the additional fully connected network on the ECG labeled data represented by the CNN features in a supervised way by minimizing cross-entropy error with dropout regularization. The experiments reported in the MIT-BIH arrhythmia, the INCART and the SVDB databases show that the proposed method can achieve better results for the detection of ventricular ectopic beats (VEB) and supraventricular ectopic beats (SVEB) compared to state-of-the-art methods.

**Keywords** ECG classification · Convolutional neural networks (CNNs)

**JEL Classification** C89

---

✉ Mohamad M. Al Rahhal  
mmalrahhal@ksu.edu.sa

Yakoub Bazi  
ybazi@ksu.edu.sa

Mansour Al Zuair  
Zuair@ksu.edu.sa

Esam Othman  
eothman@ksu.edu.sa

Bilel BenJdira  
bilel.benjdira@gmail.com

<sup>1</sup> College of Applied Computer Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia

<sup>2</sup> College of Computer and Information Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia

<sup>3</sup> Raytheon Chair for Systems Engineering, Advanced Manufacturing Institute, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia

<sup>4</sup> Research Unit Signals and Mechatronic Systems SMS, National Engineering School of Carthage, Carthage University, UR13ES49, Tunis, Tunisia

## 1 Introduction

In recent years, deep learning has emerged as a powerful computer-based method for solving various recognition problems. Deep learning was first introduced by Hinton [1] and focuses on automatically learning a good feature representation from input data [1–4]. Typical deep learning architectures include deep belief networks (DBNs) [1], the stacked autoencoder (SAE) [5], and convolutional neural networks (CNNs) [6].

Deep learning methods have provided outstanding results for several benchmark classification problems, such as image classification and segmentation [7–12], landmark detection [13], object recognition [14], face detection and recognition [15, 16], and speech recognition [17, 18]. Compared to shallow methods based on handcrafted features, deep learning methods allow learning of powerful representations in a hierarchical way thanks to their sophisticated structure.

In the biomedical engineering field, several authors have recently used these powerful methods to solve various problems, such as breast cancer diagnosis and mass classification [19, 20], abdominal adipose tissues extraction [21], detection and classification of brain tumors in MR images

[22–25], skeletal bone age assessment in X-ray Images [26], EEG classification of motor imagery [27], and arrhythmia detection and analysis using ECG signals. [28–30].

Analysis of ECG signals provides valuable information to cardiologists about the rhythm and function of the heart. Therefore, its analysis represents an efficient way to detect and treat different types of cardiac diseases [29, 31–37]. In [28], the authors proposed an approach that learns a suitable feature representation from raw ECG data in an unsupervised way using a de-noising auto-encoder (DAE). Then, they build a deep neural network (DNN) by adding a softmax regression layer on top of the resulting hidden representation layer. During the interaction phase, they allow the expert to label the most relevant ECG beats in the test record and use them to update the weights of the network. In [29], the authors proposed a 1-D convolutional neural network for ECG classification. In [30], the authors used the de-noising auto-encoder (DAE) to enhance the quality of ECG signals by removing different types of residual noise.

Although promising results have been obtained in these studies, further research is required to boost the classification accuracy. In addition, the above works mainly rely on a large initial ECG training set for learning a suitable feature representation of the ECG data to reduce the computationally demanding data-shift problem. In addition, it is well known that collection and labeling of training data are costly and time-consuming. In this paper, we propose to achieve the above goal using a small training set. To this end, we introduce a transfer learning approach based on pre-trained convolutional neural networks (CNNs). We recall that CNNs were recently introduced for the analysis of ECG signals where the authors tailored them to address one dimensional signals [29].

While CNNs work well for problems with large labeled data, they are likely subjected to over-fitting problems when dealing with datasets consisting of small labeled data. A common strategy that was recently introduced in the computer vision literature is to exploit CNNs pre-trained on very large labeled dataset and transfer the knowledge to another classification task with limited training data [38]. Examples of pre-trained CNNs include AlexNet [39], VGGNet [40], and GoogleNet [41] trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset. Typical transfer methods include fine-tuning the pretrained network using new target data or using the CNN as a feature extractor and train an external classifier (such as the support vector machine (SVM) classifier) on the new feature representation of the target data.

In this work, we propose using these models to generate a robust feature representation from the raw ECG training data available at hand. Since the pre-trained CNN accepts RGB images, whereas the ECG signals are raw vectors, we propose first converting them into images using the

Continue Wavelet transform (CWT). The choice of CWT is motivated by its success at analyzing ECG signals [42–49]. This method of representing the signal can be seen as the over-complete representation used by a special type of networks called auto-encoders where the dimensionality of the output is higher than the dimensionality of the input [50]. Unlike feature reduction, the over-complete representation allows for the discovery of more robust and sparse feature representations from the data. Then, we feed the resulting image-like data into the pre-trained CNN to generate their corresponding CNN features. During the learning phase, we train an extra network (placed on top of CNN) on the available labeled data. Then, we iteratively fine-tune this extra network by allowing the expert to interact with the system to label the most uncertain ECG beats from the records under analysis [28]. In the experiments, we validate the method in a cross-database setting using the MIT-BIH arrhythmia, INCART, and SVDB databases. The obtained results show that the proposed approach provides better accuracy improvements compared to the recent solutions. This paper conveys the following main contributions: (1) it uses the CWT and a pretrained CNN to learn a robust ECG representation, unlike the method proposed in [28] that is based on a simple DNN initialized with a DAE; (2) it uses a reduced training set (100 ECG beat per class) compared to [28] that relied on a larger training set (50,933 ECG beat); and (3) it is able to achieve better results with fewer ECG beats labeled by the expert.

This paper is organized as follows. A detailed description of ECG data processing is presented in Sect. 2. The proposed approach is presented in Sect. 3, while experimental results are reported in Sect. 4. Finally, conclusions and future directions are reported in Sect. 5.

## 2 ECG Data Processing

In the experiments, we use three different ECG databases to evaluate the proposed method, as shown in Table 1. We follow the recommendations of the Association for the Advancement of Medical Instrumentation (AAMI) for class labeling. The AAMI standard defines five classes of interest: normal (N), ventricular (V), supraventricular (S), fusion of normal and ventricular (F) and unknown beats (Q). The Q beats are removed from the analysis as they are marginally represented in these three databases obtained under different acquisition conditions and for different patients.

The most common database that has been used is the MIT-BIH Arrhythmia Database (MIT-BIH). This Database consists of 48 records (taken from 47 patients, 25 men aged 32–89 years, and 22 women aged 23–89 years). Each record is slightly over 30 min long (approximately a half-hour long) and sampled at 360 Hz. The first 23 records have been

**Table 1** ECG databases used in the experiments

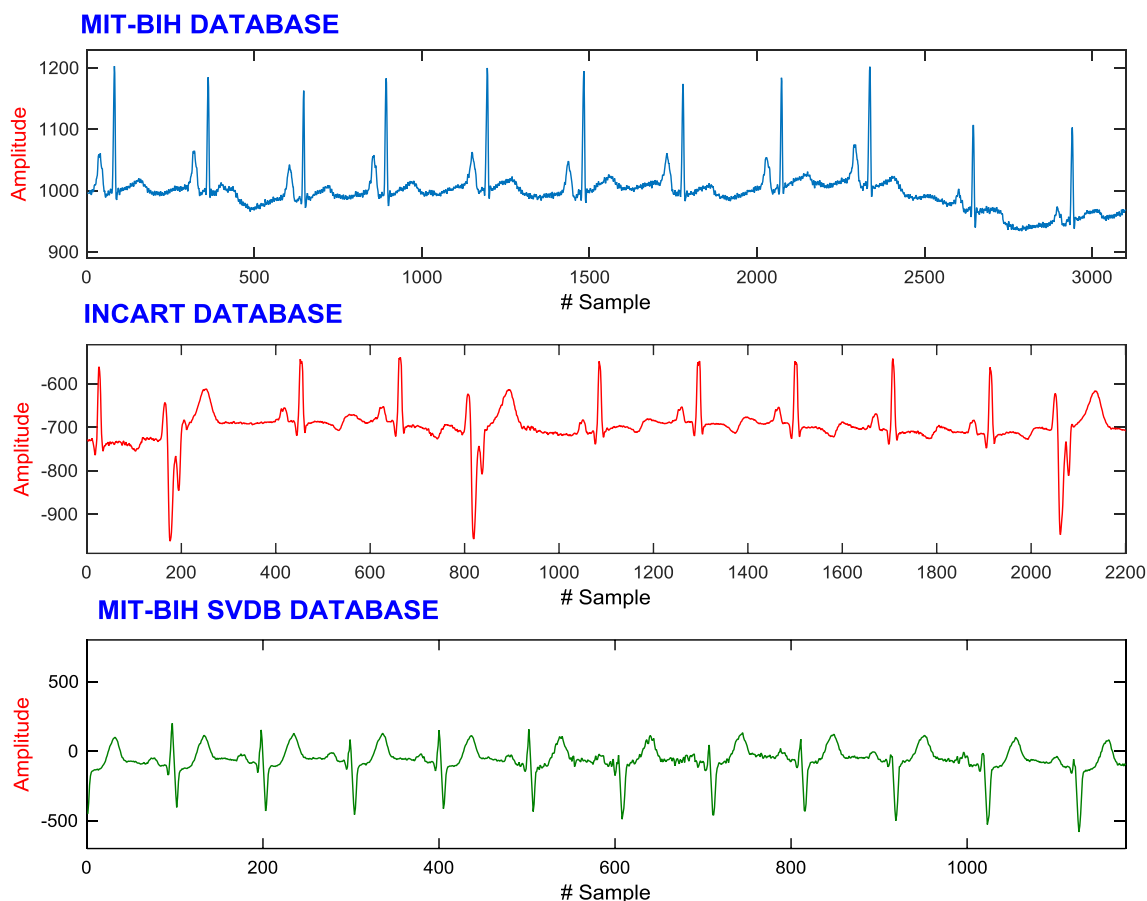
	Database	N	S	V	F	#REC	Sample rate	# Leads
ECG training beats	MIT-BIH (DS1)	45777	973	3769	414	22	360 Hz	2 leads
ECG test beats	MIT-BIH (DS2)	44011	2049	3216	388	22	360 Hz	2 leads
	INCART	153545	1958	2000	219	75	257 Hz	12 leads
	SVDB	145436	10733	8281	23	78	128 Hz	2 leads

chosen at random from this set that is numbered from 100 to 124 inclusive with some numbers missing. The remaining 25 records have been selected from the set to include the rarer variety.

To further assess the proposed approach, we use other databases, including the St.-Petersburg Institute of Cardio-logical Technics (INCART) and the MITBIH Supraventricular Arrhythmia Database (SVDB). The INCART database consists of 75 annotated recordings extracted from 32 Holter records. Each record is 30 min long and contains 12 standard leads, each sampled at 257 Hz. The reference annotation files contain over 175,000 beat annotations in all. The original records were collected from patients undergoing tests for coronary artery disease (17

men and 15 women, aged 18–80; mean age: 58). None of the patients had pacemakers, and most had ventricular ectopic beats. In selecting records to be included in the database, preference was given to subjects with ECGs consistent with ischemia, coronary artery disease, conduction abnormalities, and arrhythmias.

The MIT-BIH Supraventricular Arrhythmia Database (SVDB) consists of 78 two-lead recordings of approximately 30 min that are sampled at 128 Hz. The beat type annotations of the recordings were first automatically performed by the Marquette Electronics 8000 Holter scanner and later reviewed and corrected by a medical student, as shown in the Table 1. Figure 1 shows the raw ECG signals for three different datasets (MIT-BIH, INCART, and SVDB).

**Fig. 1** ECG row data for three different datasets

We preprocess the ECG signals to reduce noise similar to [31, 34]. To do this, we apply a 200-ms width median filter to remove the P wave and the QRS complex. Then, we use a 600-ms width median filter to remove the T wave. The resulting signals are subtracted from the original signals, which yield the baseline-corrected ECG signals. Then, we apply a 12-order low-pass filter with a 35 Hz cut-off frequency to remove the power-line and high-frequency noise. To extract the ECG waveform, we perform the QRS detection and the ECG wave boundary by means of the ecgpu-wave software [51]. Then, we resample all segmented ECG signals using a different sampling rate to the same periodic length equal to 50 uniformly distributed samples using a different sampling rate, as shown in Fig. 2. Finally, we applied the CWT with three different mother wavelets, as shown in Fig. 3.

Unlike previous studies, as training we use 400 ECG training beats (100 from each class) from the following 22 records of the MIT-BIH database termed as DS1 = {101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230} [1]. To extract the representative ECG signals from DS1, we independently

apply the *k-means* clustering algorithm to each AAMI class in DS1 to generate 100 clusters. Then, we take the centers of the resulting clusters and form the initial training set. This represents approximately 0.78% of the global training set DS1. The remaining records of this database are grouped into DS2 = {100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234}, and the other databases are used as tests for evaluating the capability of the method. Details about the number of training and test ECG samples are reported in Table 1.

### 3 Proposed Methodology

Let  $Tr = \{x_i, y_i\}_{i=1}^n$  be a training set where  $x_i \in \mathcal{R}^d$  is an ECG beat signal,  $y_i \in \{1, 2, \dots, K\}$  is its corresponding class label where  $K$  is the number of classes and  $n$  is the number of training samples. Let us also consider  $f^{CNN}$  to be a CNN model pretrained on the ImageNet dataset. Our aim is to develop a classification system that allows classifying the test ECG record  $Ts = \{x_j\}_{j=n+1}^{n+m}$  based on the available train-

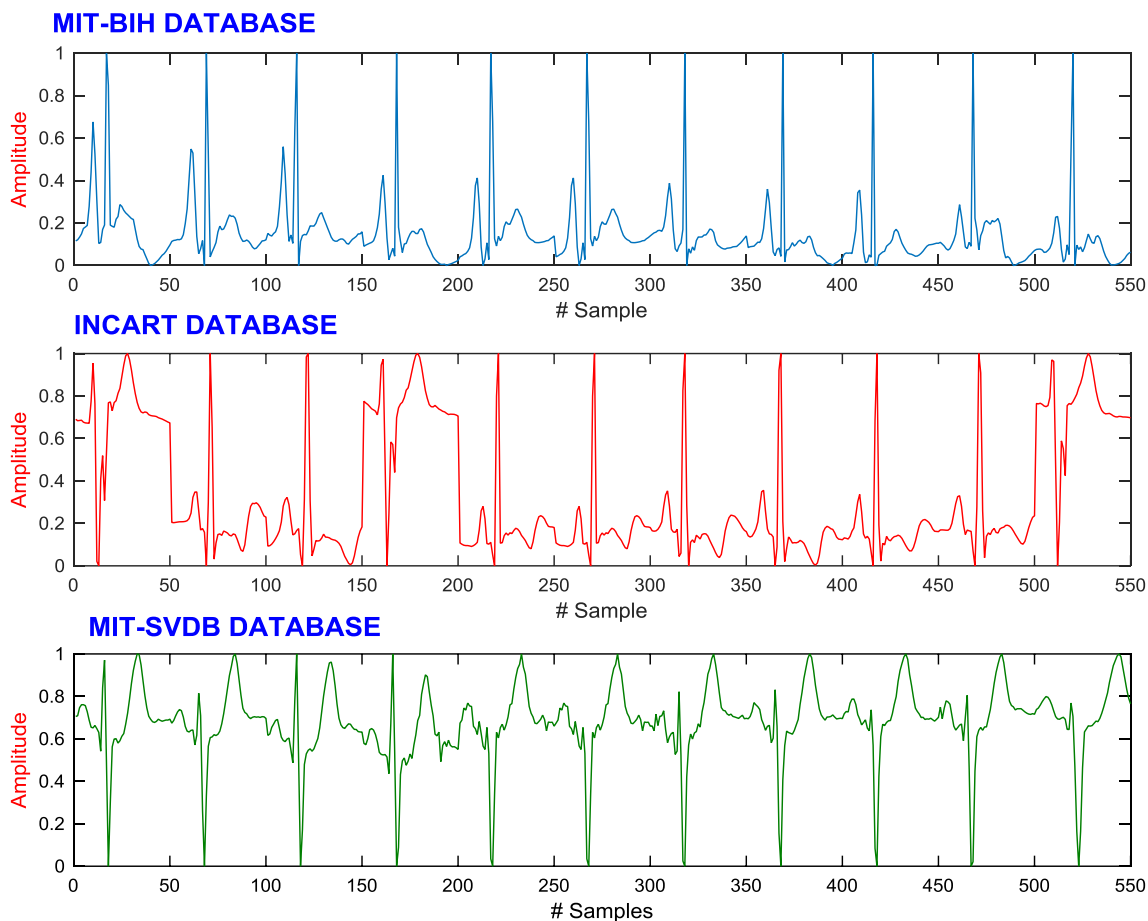


Fig. 2 ECG Enhancement data for three different datasets

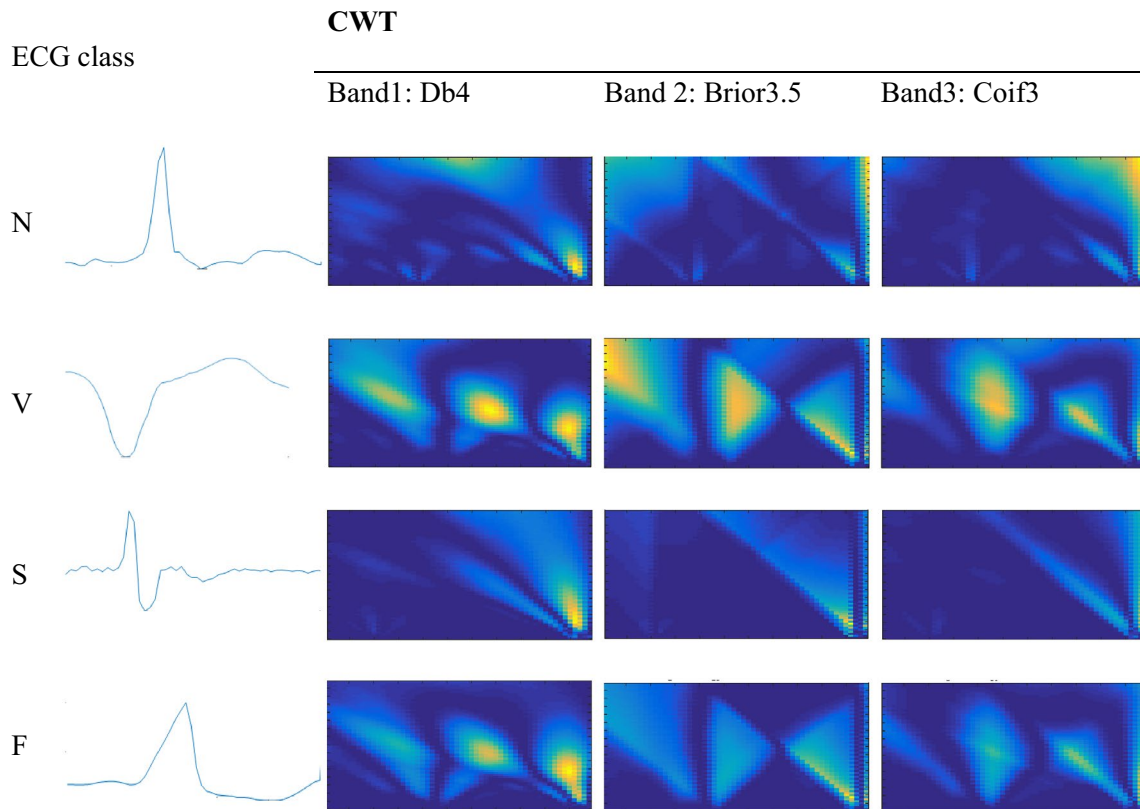


Fig. 3 Representing ECG signals as images using the CWT

ing set. Figure 4 shows the flowchart of the proposed method, whereas detailed descriptions are provided in next subsections.

### 3.1 ECG Signal Transformation Using CWT

The pretrained CNN accepts RGB images of dimensions ( $w \times h \times 3$ ) as inputs, whereas the ECG signals are raw

vectors of dimension  $d$ . Thus, it is necessary to first convert them into images using certain transformations. This work considers the CWT as a possible candidate solution since it has been shown to be efficient for analyzing the ECG signals [42–49]. Basically, the CWT allows for the mapping of the signals into a time-scale space. In addition, it allows for better visible localization of the frequency components in the analyzed signals.

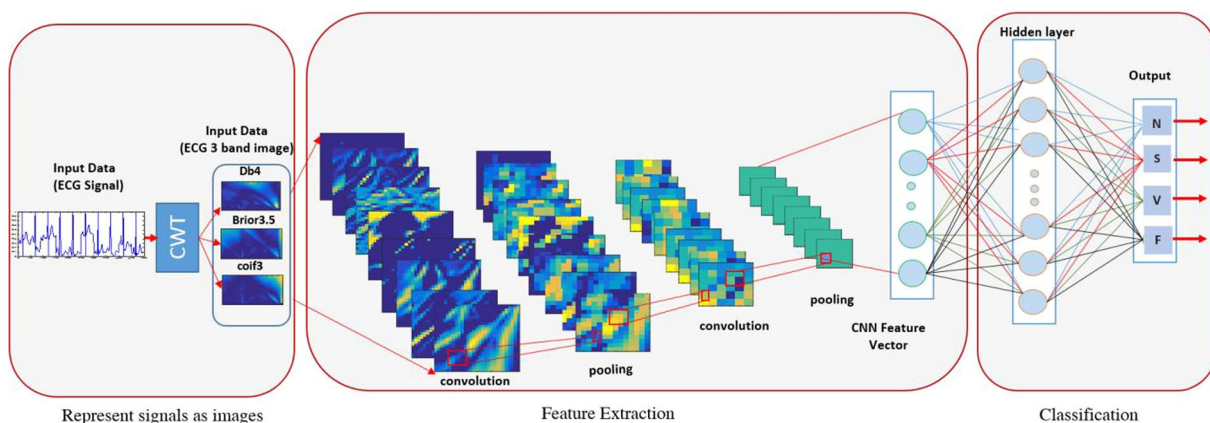


Fig. 4 Flowchart of the proposed method: Step (1) Apply the CWT to the ECG signals to generate an image-like representation. Step (2) Feature extraction. Step (3) Classification

Mathematically speaking, the CWT of a given function  $\mathbf{x}(t)$  is defined as the integral transform of  $\mathbf{x}(t)$  with a family of wavelet functions  $\psi_{a,b}(t)$ :

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} \psi_{ab}\left(\frac{t-b}{a}\right) \mathbf{x}(t) dt. \tag{1}$$

The CWT is defined as the sum of the signal multiplied by scaled and shifted versions of the wavelet function  $\psi$ .

$$CWT(\text{scale, position}) = \int_{-\infty}^{+\infty} x(t) \times \psi(\text{scale, position, } t) dt. \tag{2}$$

The function  $\psi(t)$  is known as the mother wavelet, and the components of functions  $\psi_{a,b}(t)$  are called daughter wavelets. The daughter wavelets are simply obtained by scaling and shifting the mother wavelet. The scale factor  $a$  represents the scaling of the function  $\psi(t)$ , while the shift factor  $b$  represents the temporal translation of the function. Then, the modulus  $\mathbf{I} = \text{moduls}(CWT)$  of the complex wavelet coefficients produced by this transformation can be viewed as an image. In this way, the training set  $Tr = \{\mathbf{x}_i, y_i\}_{i=1}^n$  in addition to the test record  $Ts = \{\mathbf{x}_j\}_{j=n+1}^{n+m}$  are transformed to  $Tr = \{\mathbf{I}_i, y_i\}_{i=n+1}^{n+m}$  and  $Ts = \{\mathbf{I}_j\}_{j=n+1}^{n+m}$ .

### 3.2 Feature Extraction Using a Pretrained CNN

CNN is one deep learning model that attempts to learn the feature representation for signal/image data with different levels of abstraction [12]. It is made up of several alternating convolutional and pooling layers, followed by fully connected layers. The convolutional layer is the main building block of CNN and it's the representative structure of deep models. The outputs of this layer are termed as feature maps and their number depend on the number of filters used in the convolution layer. The feature maps produced by convolving the learnable filters across the input image are usually fed to a non-linear gating function, such as the rectified linear unit (ReLU) [52]. Mathematically, let  $x^i$  and  $y^j$  denote the  $i$ -th input feature map and  $j$ -th output feature map of a convolutional layer [12]. The activation function applied in CNN can expressed as:

$$y^j = \max\left(0, b^j + \sum_i z^{ij} \times x^i\right) \tag{3}$$

where  $z^{ij}$  is the convolutional kernel between  $x^i$  and  $y^j$ , and  $b^j$  is the bias. The symbol  $*$  indicates the convolutional operation. When there are  $M$  input maps and  $N$  output maps, this layer will contain  $N$  3-D kernels of size  $d \times d \times M$  (here  $d \times d$

is the size of local receptive fields) and each kernel owns a bias.

Furthermore, the output of the activation function can be subjected to normalization to help in the generalization. The pooling layers shrink the size of the feature map (reduce the data dimensions) to produce a single output from each block. Typical popular ways to perform pooling are taking the average or the maximum.

Then, after several convolutional and pooling layers, the high-level reasoning in the neural network occurs using fully connected layers that are regarded as the classification phase at the output's end. The last layer takes all neurons in the previous layer and connects them to every single neuron it contains. In the case of classification, we add a *softmax* layer to the end of this network. Then, the complete weights are learned using the back-propagation algorithm.

### 3.3 ECG Classification

Since the CNN is composed of several layers, we can extract the features at different representation layers. Here, we take the output of the last dense fully connected layer to represent the data. That is, we feed each image  $\mathbf{I}_i$  as an input to the CNN and generate a CNN feature representation  $\mathbf{z}_i \in R^D$ .

$$\mathbf{z}_i = f^{CNN}(\mathbf{I}_i), i = 1, \dots, n + m \tag{4}$$

We feed these CNN feature vectors into an extra network placed on top of the pretrained CNN, as shown in Fig. 4. This last layer is composed of a hidden layer and a *softmax* regression layer. The hidden layer takes the input  $\mathbf{z}_i$  and maps it to another representation  $\mathbf{h}_i^{(1)} \in R^{D^{(1)}}$  of dimension  $D^{(1)}$  through the nonlinear activation function  $f$ :

$$\mathbf{h}_i^{(1)} = f(\mathbf{W}^{(1)}\mathbf{z}_i + \mathbf{b}^{(1)}) \tag{5}$$

where  $\mathbf{W}^{(1)} \in \mathfrak{R}^{D^{(1)} \times D}$  represents the mapping weight matrix and  $\mathbf{b}^{(1)} \in R^D$  is the mapping bias vector. A sigmoid function can be a typical choice of the activation function, such as  $f(v) = 1/(1 + \exp(-v))$ . For the ease of analysis, we omit the bias vector in the expression since it can be incorporated as an additional column vector in the mapping matrix. Then, in that case, the feature vector size is augmented by 1.

The *softmax* regression performs the multiclass classification and takes the resulting hidden representation  $\mathbf{h}_i^{(1)}$  as input. It produces an estimate of the posterior probability for each class label  $k = 1, 2, \dots, K$  as follows:

$$p(\hat{y}_i = k | \mathbf{x}_i) = \frac{\exp\left(\left(\mathbf{w}_k^{(2)}\right)^T \mathbf{h}_i^{(1)}\right)}{\sum_{j=1}^K \exp\left(\left(\mathbf{w}_j^{(2)}\right)^T \mathbf{h}_i^{(1)}\right)} \tag{6}$$

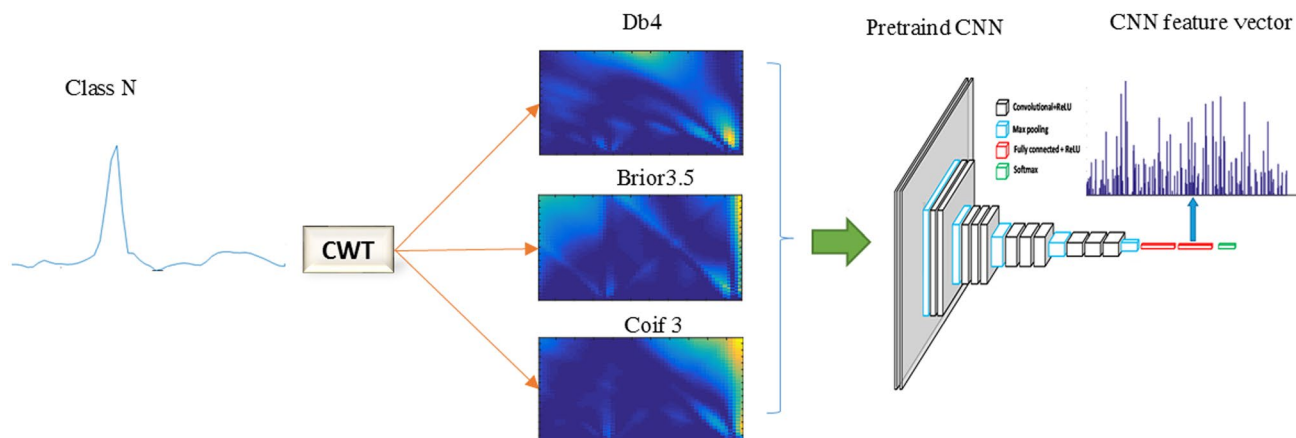


Fig. 5 Example of a CNN feature for class N

Where  $\mathbf{W}^{(2)} = [\mathbf{w}_1^{(2)} \mathbf{w}_2^{(2)} \dots \mathbf{w}_K^{(2)}] \in R^{D^{(1)} \times K}$  are the weights of the softmax regression layer and the superscript  $(\cdot)^T$  refers to the transpose operation.

We use the dropout technique introduced by Hinton [53] to prevent the network from over-fitting and to increase its generalization ability. This regularization technique aims to drop nodes of the hidden layer with their weights during the training phase. This permits us to generate a thinned network by temporarily removing the nodes from the original fully connected network, along with all its incoming and outgoing connections. Typically, the dropout regularization technique acts by defining  $\mathbf{r} \in R^{D^{(1)}}$  (the same dimension as the hidden representation  $\mathbf{h}_i^{(1)}$ ) as a vector of independent *Bernoulli* random variables, each of which has a probability  $\rho$  (which is usually set to 0.5) of being 1. At the training time, the output of the hidden layer after dropout is:

$$\begin{cases} \mathbf{r} = \text{Bernoulli}(\rho) \\ \mathbf{h}_i^{(1)} := \mathbf{h}_i^{(1)} \odot \mathbf{r} \end{cases} \quad (7)$$

with  $\odot$  denoting an element-wise product. At test time, the dropout is turned off and all hidden units are used, but the weights are scaled by the retaining probability  $\rho$ . For learning the vector of weights  $\theta = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}\}$  representing the complete network architecture, we minimize the error between the actual network outputs and the desired outputs of the training data. As the outputs of the network are probabilistic, we propose maximizing the log-posterior probability to learn the network weights, which is equivalent to minimizing the so-called cross-entropy error:

$$E_{\text{net}} = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K 1(y_i = k) \ln \left( \frac{\exp \left( \left( \mathbf{w}_k^{(2)} \right)^T \mathbf{h}_i^{(1)} \right)}{\sum_{j=1}^K \exp \left( \left( \mathbf{w}_j^{(2)} \right)^T \mathbf{h}_i^{(1)} \right)} \right) \quad (8)$$

where  $1(\cdot)$  is an indicator function that takes the value of 1 if the statement is true and otherwise it takes the value of 0, and superscript T refers to matrix transpose.

## 4 Experimental Results

### 4.1 Experiment Setup

For the pretrained CNN, we use the VGGNet model of [40] that is composed of 8 layers. It uses five convolutional filters of dimensions (number of filters  $\times$  filter height  $\times$  filter depth:  $96 \times 7 \times 7$ ,  $256 \times 5 \times 5$ ,  $512 \times 3 \times 3$ ,  $512 \times 3 \times 3$ , and  $512 \times 3 \times 3$ ) and three fully connected layers with the following number of hidden nodes (fc1: 4096, fc2: 4096, and *softmax*: 1000). This network was pretrained on the ILSVRC-12 challenge dataset. We recall that the ImageNet dataset used in this challenge is composed of 1.2 million RGB images of size  $224 \times 224$  pixels belonging to 1000 classes. These classes describe general images such as beaches, dogs, cats, cars, shopping carts, minivans, and more. As can be clearly seen, this auxiliary dataset is completely different from the ECG signals used in the experiments.

For training the extra network placed on top of the pretrained CNN, we follow the recommendations of [54] for training neural networks. We set the dropout probability  $p$  to 0.5. We use a *sigmoid* activation function for the hidden layer. For the backpropagation algorithm, we use a mini-batch gradient optimization method with the following parameters (i.e., learning rate: 0.01, momentum: 0.5, and mini-batch size: 50). The weights of the network are set initially in the range  $[-0.005, 0.005]$ . Regarding the Active learning (AL) step, we allow the expert to add  $N_{AL} = 10$  ECG beats for each iteration of the AL process.

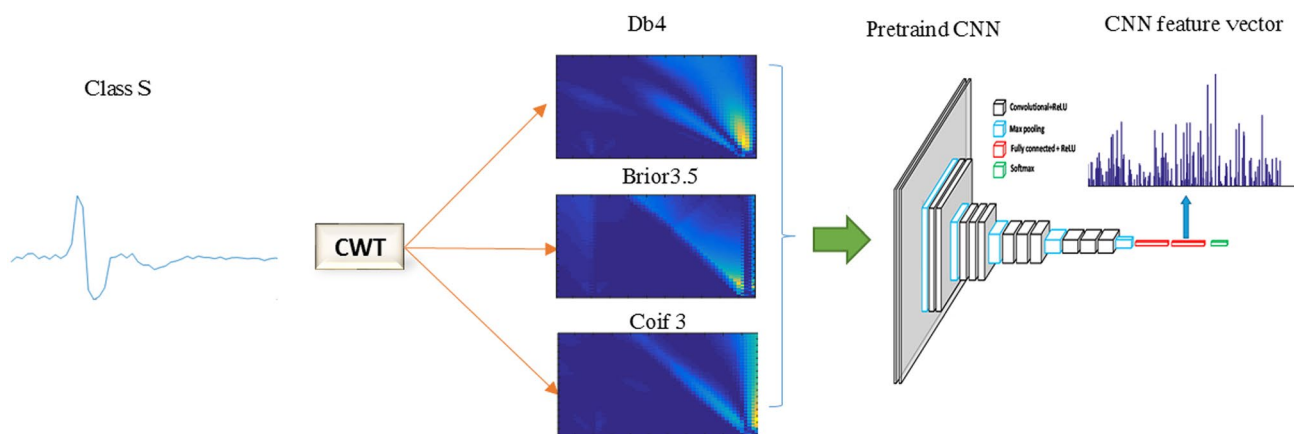


Fig. 6 Example of a CNN feature for class S

For the performance evaluation, we present the results in terms of VEB [V class versus (N, S, and F)] and SVEB [S class versus (N, V, and F)]. In particular, we use the standard measures of sensitivity ( $Se$ ), positive predictive value ( $Pp$ ), specificity ( $Sp$ ), and overall accuracy ( $OA$ ) [31, 37, 55].

### 5 Results and Discussions

As mentioned in the first step of the method, we first apply the CWT to the ECG signals to represent them as images before feeding them to the CNN network. In this context, we have explored various wavelet families with different scales for identifying a suitable initial representation for the different ECG classes. After an extensive analysis of the different wavelets, we experimentally found that the Daubechies wavelet (db4), the Biorthogonal wavelet (bior3.5), and the Coiflet wavelets (coif3) represent good choices. Therefore,

we apply these wavelets with scales from 1 to 64 to the ECG signals. Then, for every transformation, we compute the modulus of the obtained coefficients. Figure 3 depicts some views of the images obtained by applying these wavelets to the AAMI ECG classes. These views show interesting behaviors for different classes in terms of their discriminatory ability. Then, we resize the images to  $224 \times 224$ , feed them to the pretrained CNN and take the output of its last fully connected layer, which produces a CNN feature vector of dimension  $D = 4096$ .

For the MIT-BIH database, we present the results by considering three different scenarios for building the test set, as has been done by several works dealing with AAMI classes [28]. For the first scenario, we use the 11 common testing records for VEB {i.e., 200 202 210 213 214 219 221 228 231 233 234} and 14 testing records for SVEB {i.e., 200 202 210 212 213 214 219 221 222 228 231 232 233 234}. For the second one, we use the 24 common

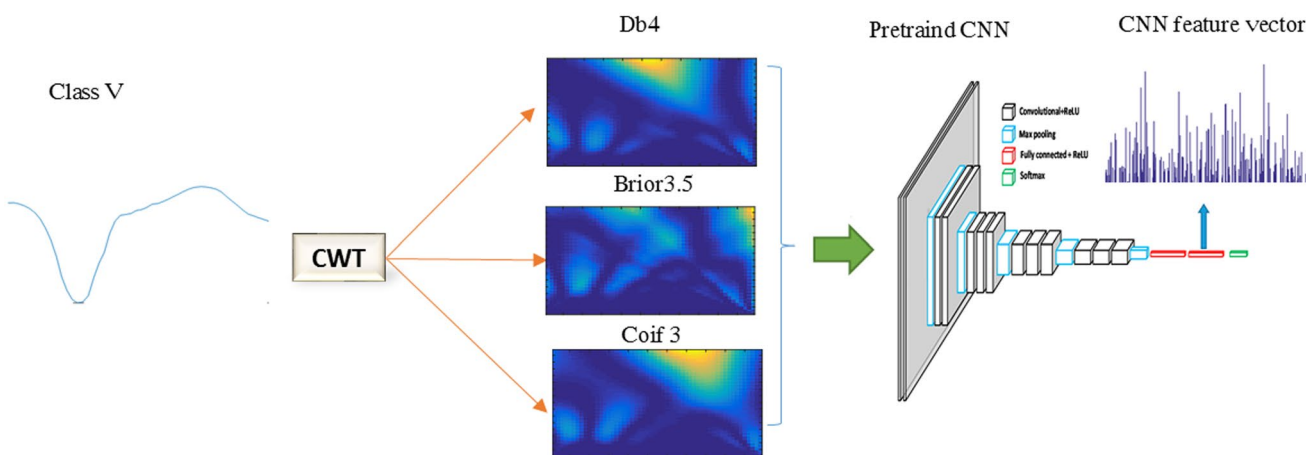


Fig. 7 Example of a CNN feature for class V



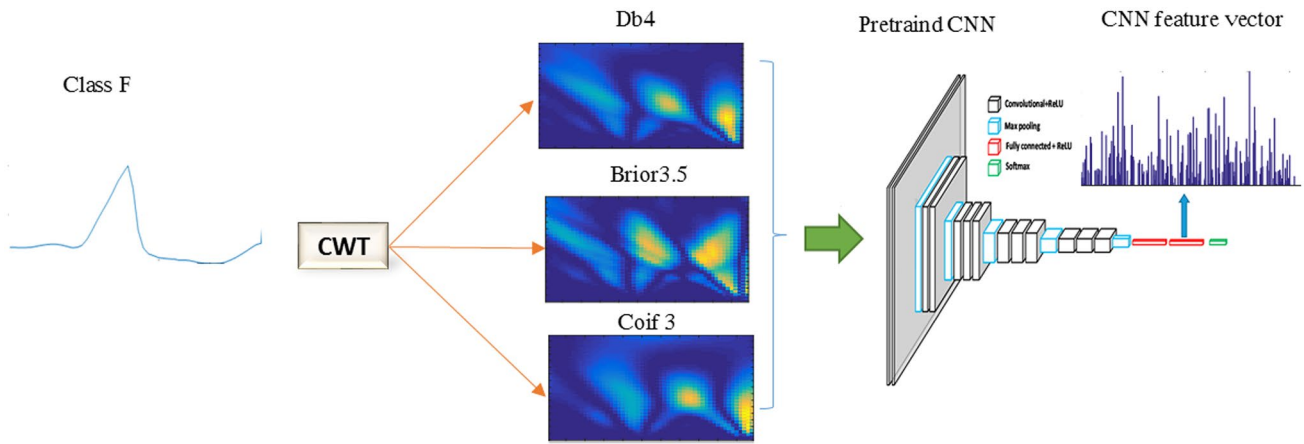


Fig. 8 Example of a CNN feature for class F

testing records from 200 to 234. For the third and last scenario, we use all 48 records (i.e., DS1 + DS2). Figures 5, 6, 7 and 8 show the CNN features obtained from the pretrained CNN for the different AAMI classes. A preliminary inspection shows that the learned features look different. Tables 2 and 3 show the classification results in

terms of (*OA, Se, SpandPp*) for SVEB and VEB, respectively, after adding 100 ECG beats per record. As seen here, these results are clearly better than those obtained by recent state-of-the-art methods for all scenarios. For the first scenario, our methods yield an (*OA, Se, Spand Pp*) equal to (99.9, 99.1, 100.0, and 99.3%) for VEB and (99.9, 97.3, 100.0 and 99.3%) for SVEB. For the second scenario,

Table 2 VEB Classification results for the MIT-BIH database

Method	Labeled by expert	VEB			
		Se	Pp	Sp	OA
De Chazaland Reilly [34]	500	94.3	96.2	N/A	99.4
Hu et al. [35]	300	78.9	75.8	96.8	94.8
Ince et al. [36]	300	90.3	92.2	98.8	97.9
Jaing and Kong [37]	300	94.3	95.8	99.4	98.8
Kiranyaz et al. [29]	300	95.9	96.2	99.4	98.9
Rahhal [56]	100	99.1	99.2	100.0	99.9
Rahhal et al. [28]	100	98.5	99.4	99.9	99.8
Proposed <sup>a</sup>	100	<b>99.6</b>	<b>99.3</b>	<b>100.0</b>	<b>99.9</b>
Jaing and Kong [37]	300	86.6	93.3	99.3	98.1
Ince et al. [36]	300	83.4	87.4	98.1	97.6
Kiranyaz et al. [29]	300	95.0	89.5	98.1	98.6
Rahhal [56]	100	98.7	98.4	99.8	99.7
Rahhal et al. [28]	100	98.5	99.2	99.8	99.8
Proposed <sup>b</sup>	100	<b>98.7</b>	<b>99.3</b>	<b>99.9</b>	<b>99.7</b>
Ince et al. [36]	300	84.6	87.4	98.7	98.3
Kiranyaz et al. [29]	300	93.9	90.6	98.9	99
Rahhal [56]	100	99.0	98.8	99.9	99.8
Rahhal et al. [28]	100	98.9	99.7	100.0	99.9
Proposed <sup>c</sup>	100	<b>99.0</b>	<b>99.6</b>	<b>100.0</b>	<b>99.9</b>

<sup>a</sup>The comparison result based on 11 common recording

<sup>b</sup>The comparison result based on 24 common testing records (200–234)

<sup>c</sup>The comparison result based on all training and testing records (DS1+DS2)

Table 3 SVEB classification results for the MIT-BIH database

Method	Labeled by Expert	SVEB			
		Se	Pp	Sp	OA
De Chazal and Reilly [34]	500	87.7	47	N/A	95.9
Ince et al. [36]	300	81.8	63.4	98.5	96.1
Jaing and Kong [37]	300	74.9	78.8	98.8	97.5
Kiranyaz et al. [29]	300	68.8	79.2	99.5	96.4
Rahhal [56]	100	84.0	93.8	100.0	99.9
Rahhal et al. [28]	100	93.8	98.8	99.9	99.6
Proposed <sup>a</sup>	100	<b>97.3</b>	<b>98.9</b>	<b>100.0</b>	<b>99.9</b>
Jaing and Kong [37]	300	50.6	67.9	98.8	96.6
Ince et al. [36]	300	62.1	56.7	98.5	96.1
Kiranyaz et al. [29]	300	64.6	62.1	98.6	96.4
Rahhal [56]	100	94.9	97.1	99.9	99.7
Rahhal et al. [28]	100	94.4	97.5	99.9	99.7
Proposed <sup>b</sup>	100	<b>98.3</b>	<b>98.1</b>	<b>99.4</b>	<b>99.3</b>
Ince et al. [36]	300	63.2	53.7	99	97.4
Kiranyaz et al. [29]	300	60.3	63.5	99.2	97.6
Rahhal [56]	100	94.4	96.7	99.9	99.8
Rahhal et al. [28]	100	95.3	97.6	99.9	99.8
Proposed <sup>c</sup>	100	<b>95.9</b>	<b>98.3</b>	<b>100.0</b>	<b>99.8</b>

<sup>a</sup>The comparison result based on 14 records for SVEB

<sup>b</sup>The comparison result based on 24 common testing records (200–234)

<sup>c</sup>The comparison result based on all training and testing records (DS1+DS2)

**Table 4** VEB classification results obtained for the INCART and SVDB databases

Dataset	Method	Labeled	VEB			
			Se	Pp	Sp	OA
INCART	Rahhal et al. [28]	100	94.59	97.18	99.62	99.07
	Proposed	100	<b>96.50</b>	<b>98.00</b>	<b>99.70</b>	<b>99.23</b>
SVDB	Rahhal et al. [28]	100	91.41	90.19	99.47	99.07
	Proposed	<b>100</b>	<b>91.70</b>	<b>96.70</b>	<b>99.80</b>	<b>99.40</b>

**Table 5** SVEB classification results obtained for the INCART and SVDB databases

Dataset	Method	Labeled	SVEB			
			Se	Pp	Sp	OA
INCART	Rahhal et al. [28]	100	83.91	96.25	99.96	99.78
	Proposed	100	<b>89.30</b>	<b>97.50</b>	<b>99.93</b>	<b>99.82</b>
SVDB	Rahhal et al. [28]	100	74.43	89.83	99.41	97.78
	Proposed	<b>100</b>	<b>80.20</b>	<b>94.90</b>	<b>99.70</b>	<b>98.40</b>

we obtain (99.7, 98.7, 99.9, and 98.8%) for VEB and (99.3, 98.3, 99.4 and 98.1%) for SVEB. Finally, for the last one, we obtain (99.9, 99.0, 100.0, and 99.6%) for VEB and (99.8, 95.9, 100.0 and 98.3%) for SVEB.

For the other databases, the (*OA*, *Se*, *Sp* and *Pp*) obtained by the method in the term of VEB are equal to (99.23, 96.5, 99.7, and 98.0%) for the INCART database and (99.4, 91.7, 99.8, and 96.7%) for the SVDB database, as shown in Table 4. For SVEB, the (overall and average) accuracies are (99.82, 89.30, 99.93, and 97.50%) for the INCART database and (98.4, 80.2, 99.7, and 94.9%) for the SVDB database, as shown in Table 5.

## 6 Conclusions

In this paper, we presented a method based on a CNN for the classification of ECG signals. Compared to the existing solutions, this method has the following attractive properties: (1) it transfers knowledge from a CNN pretrained on a different domain from computer vision (called ImageNet) with large labeled images; (2) it exploits the CWT to make the ECG signals suitable inputs for this network; and (3) it uses an efficient AL strategy to fine-tune the extra network placed on top of the CNN by minimizing the cross-entropy error with dropout regularization. Experiments carried out on a non-GPU unit and on three ECG cross-databases (obtained under different acquisition conditions) confirmed its efficiency and ability to provide improved classification results versus several other methods. In the future, we plan to improve this method by including several modifications to further reduce expert interactions while improving its accuracy. These improvements include: (1) learning suitable representations for transforming the ECG data into

images instead of applying the CWT, (2) fusing several pretrained CNN models for generating more robust feature representations, and (3) exploring other AL criteria for identifying the most relevant ECG beats in the test records.

**Acknowledgements** The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding this Research group NO.(RG -1435-050).

## Compliance with Ethical Standards

**Conflicts of interest** The authors that they declare no conflict of interest.

## References

- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- Sun, X., Nasrabadi, N. M., & Tran, T. D. (2015). Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8), 4457–4471. <https://doi.org/10.1109/TGRS.2015.2399978>.
- Li, J.-C., Ng, W. W. Y., Yeung, D. S., & Chan, P. P. K. (2014). Bi-firing deep neural networks. *International Journal of Machine Learning and Cybernetics*, 5(1), 73–83. <https://doi.org/10.1007/s13042-013-0198-9>.
- Zhang, J., Ding, S., Zhang, N., & Shi, Z. (2016). Incremental extreme learning machine based on deep feature embedded. *International Journal of Machine Learning and Cybernetics*, 7(1), 111–120. <https://doi.org/10.1007/s13042-015-0419-5>.
- Swietojanski, P., Ghoshal, A., & Renals, S. (2014). Convolutional neural networks for distant speech recognition. *IEEE Signal Processing Letters*, 21(9), 1120–1124. <https://doi.org/10.1109/LSP.2014.2325781>.

6. Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>.
7. Gao, Z., Wang, L., Zhou, L., & Zhang, J. (2017). HEp-2 cell image classification with deep convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 21(2), 416–428. <https://doi.org/10.1109/JBHI.2016.2526603>.
8. Li, W., Wu, G., Zhang, F., & Du, Q. (2016). Hyperspectral image classification using deep pixel-pair features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 844–853. <https://doi.org/10.1109/TGRS.2016.2616355>.
9. Huang, Y., Wu, R., Sun, Y., Wang, W., & Ding, X. (2015). Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 1951–1960. <https://doi.org/10.1109/TITS.2014.2387069>.
10. Hariharan, B., Arbelaez, P., Girshick, R., & Malik, J. (2017). Object instance segmentation and fine-grained localization using hypercolumns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 627–639. <https://doi.org/10.1109/TPAMI.2016.2578328>.
11. Wu, X., Du, M., Chen, W., Li, Z. (2016). Exploiting deep convolutional network and patch-level CRFs for indoor semantic segmentation. Presented at the 2016 IEEE. In: 11th Conference on Industrial Electronics and Applications (ICIEA). pp. 150–155. <https://doi.org/10.1109/iciea.2016.7603568>.
12. Liu, Y., Chen, X., Peng, H., & Wang, Z. (2017). Multi-focus image fusion with a deep convolutional neural network. *Information Fusion*, 36, 191–207. <https://doi.org/10.1016/j.inffus.2016.12.001>.
13. Prentašić, P., & Lončarić, S. (2016). Detection of exudates in fundus photographs using deep neural networks and anatomical landmark detection fusion. *Computer Methods and Programs in Biomedicine*, 137, 281–292. <https://doi.org/10.1016/j.cmpb.2016.09.018>.
14. Bai, J., Wu, Y., Zhang, J., & Chen, F. (2015). Subset based deep learning for RGB-D object recognition. *Neurocomputing*, 165, 280–292. <https://doi.org/10.1016/j.neucom.2015.03.017>.
15. Huang, Z., Wang, R., Shan, S., & Chen, X. (2015). Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning. *Pattern Recognition*, 48(10), 3113–3124. <https://doi.org/10.1016/j.patcog.2015.03.011>.
16. Tao, Q.-Q., Zhan, S., Li, X.-H., & Kurihara, T. (2016). Robust face detection using local CNN and SVM based on kernel combination. *Neurocomputing*, 211, 98–105. <https://doi.org/10.1016/j.neucom.2015.10.139>.
17. Cai, M., & Liu, J. (2016). Maxout neurons for deep convolutional and LSTM neural networks in speech recognition. *Speech Communication*, 77, 53–64. <https://doi.org/10.1016/j.specom.2015.12.003>.
18. Li, X., Yang, Y., Pang, Z., & Wu, X. (2015). A comparative study on selecting acoustic modeling units in deep neural networks based large vocabulary Chinese speech recognition. *Neurocomputing*, 170, 251–256. <https://doi.org/10.1016/j.neucom.2014.07.087>.
19. Jiao, Z., Gao, X., Wang, Y., & Li, J. (2016). A deep feature based framework for breast masses classification. *Neurocomputing*, 197, 221–231. <https://doi.org/10.1016/j.neucom.2016.02.060>.
20. Sun, W., Tseng, T.-L., Zhang, J., & Qian, W. (2017). Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data. *Computerized Medical Imaging and Graphics*, 57, 4. <https://doi.org/10.1016/j.compmedima.2016.07.004>.
21. Jiang, F., Li, H., Hou, X., Sheng, B., Shen, R., Liu, X.-Y., et al. (2017). Abdominal adipose tissues extraction using multi-scale deep neural network. *Neurocomputing*, 229, 23–33. <https://doi.org/10.1016/j.neucom.2016.07.059>.
22. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., et al. (2017). Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35, 18–31. <https://doi.org/10.1016/j.media.2016.05.004>.
23. Gao, X. W., Hui, R., & Tian, Z. (2017). Classification of CT brain images based on deep learning networks. *Computer Methods and Programs in Biomedicine*, 138, 49–56. <https://doi.org/10.1016/j.cmpb.2016.10.007>.
24. Kleesiek, J., Urban, G., Hubert, A., Schwarz, D., Maier-Hein, K., Bendszus, M., et al. (2016). Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. *NeuroImage*, 129, 460–469. <https://doi.org/10.1016/j.neuroimage.2016.01.024>.
25. Kamnitsas, K., Ledig, C., Newcombe, V. F. J., Simpson, J. P., Kane, A. D., Menon, D. K., et al. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36, 61–78. <https://doi.org/10.1016/j.media.2016.10.004>.
26. Spampinato, C., Palazzo, S., Giordano, D., Aldinucci, M., & Leonardi, R. (2017). Deep learning for automated skeletal bone age assessment in X-ray images. *Medical Image Analysis*, 36, 41–51. <https://doi.org/10.1016/j.media.2016.10.010>.
27. Tang, Z., Li, C., & Sun, S. (2017). Single-trial EEG classification of motor imagery using deep convolutional neural networks. *Optik - International Journal for Light and Electron Optics*, 130, 11–18. <https://doi.org/10.1016/j.ijleo.2016.10.117>.
28. Rahhal, M. M. A., Bazi, Y., AlHichri, H., Alajlan, N., Melgani, F., & Yager, R. R. (2016). Deep learning approach for active classification of electrocardiogram signals. *Information Sciences*, 345, 340–354. <https://doi.org/10.1016/j.ins.2016.01.082>.
29. Kiranyaz, S., Ince, T., & Gabbouj, M. (2016). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Bio-Medical Engineering*, 63(3), 664–675. <https://doi.org/10.1109/TBME.2015.2468589>.
30. Xiong, P., Wang, H., Liu, M., Zhou, S., Hou, Z., & Liu, X. (2016). ECG signal enhancement based on improved denoising auto-encoder. *Engineering Applications of Artificial Intelligence*, 52, 194–202. <https://doi.org/10.1016/j.engappai.2016.02.015>.
31. De Chazal, P., O'Dwyer, M., & Reilly, R. B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Bio-Medical Engineering*, 51(7), 1196–1206. <https://doi.org/10.1109/TBME.2004.827359>.
32. Homaeinezhad, M. R., Atyabi, S. A., Tavakkoli, E., Toosi, H. N., Ghaffari, A., & Ebrahimipour, R. (2012). ECG arrhythmia recognition via a neuro-SVM-KNN hybrid classifier with virtual QRS image-based geometrical features. *Expert Systems with Applications*, 39(2), 2047–2058. <https://doi.org/10.1016/j.eswa.2011.08.025>.
33. Zhang, Z., Dong, J., Luo, X., Choi, K. S., & Wu, X. (2014). Heartbeat classification using disease-specific feature selection. *Computers in Biology and Medicine*, 46, 79–89. <https://doi.org/10.1016/j.compbiomed.2013.11.019>.
34. De Chazal, P., & Reilly, R. B. (2006). A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features. *IEEE Transactions on Bio-Medical Engineering*, 53(12 Pt 1), 2535–2543. <https://doi.org/10.1109/TBME.2006.883802>.
35. Hu, Y. H., Palreddy, S., & Tompkins, W. J. (1997). A patient-adaptable ECG beat classifier using a mixture of experts approach. *IEEE Transactions on Bio-Medical Engineering*, 44(9), 891–900. <https://doi.org/10.1109/10.623058>.
36. Ince, T., Kiranyaz, S., & Gabbouj, M. (2009). A generic and robust system for automated patient-specific classification of ECG signals. *IEEE Transactions on Bio-Medical Engineering*, 56(5), 1415–1426. <https://doi.org/10.1109/TBME.2009.2013934>.
37. Jiang, W., & Kong, S. G. (2007). Block-based neural networks for personalized ECG signal classification. *IEEE Transactions*

- on *Neural Networks*, 18(6), 1750–1761. <https://doi.org/10.1109/TNN.2007.900239>.
38. Azizpour, H., Razavian, A. S., Sullivan, J., Maki, A., & Carlsson, S. (2016). Factors of transferability for a generic ConvNet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9), 1790–1802. <https://doi.org/10.1109/TPAMI.2015.2500224>.
  39. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). USA: Curran Associates Inc.
  40. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. <https://arxiv.org/abs/1409.1556>.
  41. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions, Presented at the 2015. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (pp. 1–9). <https://doi.org/10.1109/cvpr.2015.7298594>.
  42. Priya, K. D., Rao, G. S., & Rao, P. S. V. S. (2016). Comparative analysis of wavelet thresholding techniques with wavelet-Wiener filter on ECG signal. *Procedia Computer Science*, 87, 178–183. <https://doi.org/10.1016/j.procs.2016.05.145>.
  43. Yochum, M., Renaud, C., & Jacquir, S. (2016). Automatic detection of P, QRS and T patterns in 12 leads ECG signal based on CWT. *Biomedical Signal Processing and Control*, 25, 46–52. <https://doi.org/10.1016/j.bspc.2015.10.011>.
  44. Remya, R. S., Indiradevi, K. P., & Babu, K. K. A. (2016). Classification of myocardial infarction using multi resolution wavelet analysis of ECG. *Procedia Technology*, 24, 949–956. <https://doi.org/10.1016/j.protcy.2016.05.195>.
  45. Nannaparaju, V., & Narasimman, S. (2015). Detection of T-wave alternans in ECGs by wavelet analysis. *Procedia Materials Science*, 10, 307–313. <https://doi.org/10.1016/j.mspro.2015.06.055>.
  46. Thomas, M., Das, M. K., & Ari, S. (2015). Automatic ECG arrhythmia classification using dual tree complex wavelet based features. *AEU—International Journal of Electronics and Communications*, 69(4), 715–721. <https://doi.org/10.1016/j.aeue.2014.12.013>.
  47. Mahapatra, S., Mohanta, D., Mohanty, P., Nayak, Sk., & Behari, Pk. (2016). A neuro-fuzzy based model for analysis of an ECG signal using wavelet packet tree. *Procedia Computer Science*, 92, 175–180. <https://doi.org/10.1016/j.procs.2016.07.343>.
  48. Kumar, R., Kumar, A., & Singh, G. K. (2016). Hybrid method based on singular value decomposition and embedded zero tree wavelet technique for ECG signal compression. *Computer Methods and Programs in Biomedicine*, 129, 135–148. <https://doi.org/10.1016/j.cmpb.2016.01.006>.
  49. Mourad, K., & Fethi, B. R. (2016). Efficient automatic detection of QRS complexes in ECG signal based on reverse biorthogonal wavelet decomposition and nonlinear filtering. *Measurement*, 94, 663–670. <https://doi.org/10.1016/j.measurement.2016.09.014>.
  50. Vincent, P., Larochele, H., Lajoie, I., Bengio, Y., & Manzagol, P. (2010). Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11, 3371–3408.
  51. PhysioNet. (2016). ECGPUWAVE (MATLAB/Octave version). <https://physionet.org/physiostools/ecgpuwave/src/matlab>. Accessed 11 July 2017.
  52. Nocedal, J. (1980). Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151), 773–782. <https://doi.org/10.1090/S0025-5718-1980-0572855-7>.
  53. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.
  54. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828. <https://doi.org/10.1109/TPAMI.2013.50>.
  55. Kutlu, Y., & Kuntalp, D. (2012). Feature extraction for ECG heartbeats using higher order statistics of WPD coefficients. *Computer Methods and Programs in Biomedicine*, 105(3), 257–267. <https://doi.org/10.1016/j.cmpb.2011.10.002>.
  56. Rahhal, M. M. A., Bazi, Y., Alajlan, N., Malek, S., Al-Hichri, H., Melgani, F., et al. (2015). Classification of AAMI heartbeat classes with an interactive ELM ensemble learning approach. *Biomedical Signal Processing and Control*, 19, 56–67. <https://doi.org/10.1016/j.bspc.2015.03.010>.