applied sciences

MDPI

Article

An Efficient Algorithm for Cardiac Arrhythmia Classification Using Ensemble of Depthwise Separable Convolutional Neural Networks

Eko Ihsanto [1], Kalamullah Ramli [1,*], Dodi Sudiana [1] and Teddy Surya Gunawan [2]

[1] Department of Electrical Engineering, Universitas Indonesia, Depok, Jawa Barat 16424, Indonesia; eko.ihsanto@gmail.com (E.I.); dodi.sudiana@ui.ac.id (D.S.)

[2] Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia; tsgunawan@iium.edu.my

[*] Correspondence: kalamullah.ramli@ui.ac.id

Received: 30 November 2019; Accepted: 3 January 2020; Published: 9 January 2020

Abstract: Many algorithms have been developed for automated electrocardiogram (ECG) classification. Due to the non-stationary nature of the ECG signal, it is rather challenging to use traditional handcraft methods, such as time-based analysis of feature extraction and classification, to pave the way for machine learning implementation. This paper proposed a novel method, i.e., the ensemble of depthwise separable convolutional (DSC) neural networks for the classification of cardiac arrhythmia ECG beats. Using our proposed method, the four stages of ECG classification, i.e., QRS detection, preprocessing, feature extraction, and classification, were reduced to two steps only, i.e., QRS detection and classification. No preprocessing method was required while feature extraction was combined with classification. Moreover, to reduce the computational cost while maintaining its accuracy, several techniques were implemented, including All Convolutional Network (ACN), Batch Normalization (BN), and ensemble convolutional neural networks. The performance of the proposed ensemble CNNs were evaluated using the MIT-BIH arrythmia database. In the training phase, around 22% of the 110,057 beats data extracted from 48 records were utilized. Using only these 22% labeled training data, our proposed algorithm was able to classify the remaining 78% of the database into 16 classes. Furthermore, the sensitivity (S_n), specificity (S_p), and positive predictivity (P_p), and accuracy (Acc) are 99.03%, 99.94%, 99.03%, and 99.88%, respectively. The proposed algorithm required around 180 µs, which is suitable for real time application. These results showed that our proposed method outperformed other state of the art methods.

Keywords: depthwise separable convolution (DSC); all convolutional network (ACN); batch normalization (BN); ensemble convolutional neural network (ECNN); electrocardiogram (ECG); MIT-BIH database

1. Introduction

ECG signals can be easily acquired by putting one's finger on the sensor for about 30 s [1]. There are at least two types of important information contained in the ECG signal, including those related to health or biomedical [2–4] and those related to the person identification or biometrics [5–7]. Due to its convenience, many ECG classification algorithms have been developed, including handcraft [4,8,9] and machine learning [10–15] methods. The handcraft method is rather difficult to utilize on non-stationary signals, such as ECG, while machine learning methods normally require high computational resources. Due to its high accuracy, the machine learning method is preferable compared to the handcraft method. However, an efficient algorithm is required to minimize the computational requirements while still maintaining its high accuracy.

Many researches have been conducted on the implementation of handcraft techniques, including the extraction of time-based ECG features using Fourier [8] and wavelet [4,9] transforms. Both Fourier and wavelet transform can be used for ECG beats detection (QRS detection), as well as feature extraction, such as R-peak, RR-interval, T-wave region, and QT-zone. After QRS detection, amplitude and duration-based ECG features can also be measured using weighted diagnostic distortion (WDD) [16]. The feature extraction stage is usually followed by a classification stage with various methods such as vector quantization [17], random forest [18–20], k-nearest neighbor (kNN) [10,20,21], support vector machine (SVM) [10,13,18,20], multi-layer perceptron (MLP) [22–24] and convolutional neural network (CNN) [25].

If the feature extraction and classification stages are done separately, SVM can be used and optimized using particle swarm optimization (PSO) [10]. As presented in [10], after supervised training was conducted using 500 beats, the model can classify 40,438 test beats into five classes with accuracy of 89.72%, outperforming the other methods such as kNN or radial basis function (RBF). Nevertheless, the quality of this classification can still be improved in terms of increasing the number of classes and/or accuracy. For example, discrete orthogonal Stockwell transform (DOST) could be used during feature extraction followed by principal component analysis (PCA) to reduce feature dimensions [13]. As shown in [13], after supervised training was conducted on 23,996 beats, the remainder 86,113 test beats could be classified into 16 classes with better accuracy of 98.82%.

Another promising method for improving efficiency is by combining both feature extraction and classification stages using MLP [26] and CNN [27]. For example, in [27], a neural network model containing three layers of CNN and two layers of MLP was proposed. The input of this model is a raw ECG beat signal containing 64 or 128 samples centered on the R-peak. While the number of ECG beats used for training is kept at minimum at 245 and the testing beats is set to 100,144, it can achieve accuracy of 95.14% to classify five classes. In [28], autoencoder was utilized with a rather good result but it needs to fairly evaluate the performance with and without denoising. Moreover, the deep networks configuration could be further optimized to reduce computational time.

Although many researches have been conducted, an efficient algorithm for cardiac arrhythmia classification is still required. Therefore, the objective of this paper was to simplify the overall process to lower the computational cost, while maintaining high accuracy. A neural network model presented in [27] was adopted and modified to now classify 16 classes as in [13]. The performance of our proposed algorithms was evaluated using number of classes, prediction stages, and accuracy.

2. 1D Convolutional Neural Network and Its Enhancement

In this paper, we use two stages of ECG classification, namely beat segmentation and classification. For classification purposes, CNN could be used as stated in [27,29]. Although CNN hyperparameters such as number of filters, filter size, padding type, activation type, pooling, backpropagation, still have to be done intuitively or by trial and error, there are still some techniques that can be used to reduce the amount of trial and error attempts to achieve the best results. Further enhancement to the CNN could be done using All Convolutional Network (ACN) [30], Batch Normalization (BN) [31]. Depthwise Separable Convolution (DSC) [32], ensemble CNN [33], which are further elaborated in this section. Out of various methods for CNN enhancement, DSC has the greatest effect on decreasing training time, while ensemble CNN enables further improvement on the classification rate.

ACN is used to replace pooling layer with stride during convolution. Pooling or stride is used for downsampling as CNN output parameters are less than the input parameter. In our case, the CNN input parameter is set to 256, while the output parameter is set to 16. Normally, the last or output layer of CNN uses the SoftMax activation function to determine the output class based on its highest probability. To reduce the number of layers, pooling layer could be replaced with stride during convolution [30].

Input normalization is required to solve internal covariate shift, i.e., the change in the distribution of network activations due to the change in network parameters during training. Without normalization, this will slow down the training iteration or even stop the iteration before reaching adequate accuracy. To solve this issue, BN can be conducted for each training mini-batch [31]. To reduce computational cost, BN could be conducted during the convolution process before nonlinear activation, such as rectified linear unit (ReLU). On the other hand, DSC could be used to reduce the number of parameters and floating points multiplication operation, with negligible performance degradation [34]. DSC could be performed on one layer or a group of layers of CNNs.

Another thing that must be considered in designing the CNN model is the implementation of the Flatten layer before the Fully Connected (FC) layer. Even though Flatten technically can be replaced by the AveragePool layer, these two techniques differ in terms of execution time and the final results obtained. The Flatten process does not require any further calculations, it only changes the arrangement of parameters in the last layer, while AveragePool must perform arithmetic operations to get the average value of each group in the last layer according to its position. The next effect of Flatten causes more neurons connected to FC, in comparison to the number of neurons produced by AveragePool which is obviously related to the number of arithmetic operations at the FC layer. It should be noted that the number of neurons in the Flatten layer represents all local features on the last layer without having to be combined in the average value.

2.1. 1-D CNNs

As described in [27,29], during the forward propagation, the input map of the next layer neuron will be obtained by the cumulation of the final output maps of the previous layer neurons convolved with their individual kernels as follows:

$$x_k^l = \beta_k^l + \sum_{i=1}^{N_{l-1}} conv1D\left(\omega_{ik}^{l-1}, s_i^{l-1}\right) \tag{1}$$

where $conv1D(\cdot,\cdot)$ is 1-D convolution, x_k^l is the input, β_k^l is the bias of the k-th neuron at layer l, s_i^{l-1} is the output of the i-th neuron at layer $l-1$, ω_{ik}^{l-1} is the kernel (weight) from the i-th neuron at layer $l-1$ to the k-th neuron at layer l. Let $l = 1$ and $l = L$ be the input and output layers, respectively. The inter backpropagation delta error of the output s_k^l can be expressed as follows:

$$\Delta s_k^l = \sum_{i=1}^{N_{l+1}} conv1Dz\left(\Delta_i^{l+1}, rev\left(\omega_{ik}^l\right)\right) \tag{2}$$

where $rev(\cdot)$ flips the array, and $conv1Dz(\cdot,\cdot)$ performs full convolution in 1-D with $K-1$ zero padding. Lastly, the weight and bias sensitivities can be expressed as follows:

$$\frac{\partial E}{\partial \omega_{ik}^l} = conv1D\left(s_k^l, \Delta_i^{l+1}\right) \tag{3}$$

$$\frac{\partial E}{\partial \beta_k^l} = \sum_n \Delta_k^l(n) \tag{4}$$

2.2. All Convolutional Network

All Convolutional Network (ACN) [30] is utilized by removing the max pooling layer and replacing it with convolutional stride. As a result, the computational cost will be reduced, as it removes several layers as well as reducing the floating point operation on the convolution operation. For example, conv stride = 1 continued with max pooling 2 could be replaced with one layer, i.e., conv stride = 2 with an almost similar result.

Let f denote a feature map produced by some layer of a CNN, while N is the number of filters in this layer. Then p-norm subsampling with pooling size m (or half-length $\frac{m}{2}$) and stride r applied to the feature map f is a 3-dimensional array $s(f)$ with the following entries:

$$s_{i,j,u}(f) = \left(\sum_{h=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{w=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \left| f_g(h,w,i,j,u) \right|^p \right)^{\frac{1}{p}} \tag{5}$$

where $g(h,w,i,j,u) = (r \cdot i + h, r \cdot j + w, u)$ is the function mapping from positions in s to positions in f representing the stride, p is the order of the p-norm ($p \to \infty$ represents max pooling). If $r > m$, pooling regions do not overlap. The standard definition of a convolution layer c applied to the feature map f is given as:

$$c_{i,j,o}(f) = \sigma \left(\sum_{h=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{w=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{u=1}^{N} \theta_{h,w,u,o} \cdot f_{g(h,w,i,j,u)} \right) \tag{6}$$

where θ are the convolutional weights, $\sigma(\cdot)$ is the activation function, typically a rectified linear activation ReLU $\sigma(x) = max(x,0)$, and $o \in [1,M]$ is the number of output features of the convolutional layer. From the two equations, it is evident that the computational cost will be reduced.

2.3. Batch Normalization

Batch Normalization (BN) is intended to avoid non-linear saturation or internal covariate shifts causing a faster learning process [31]. The Batch Normalization allows much higher learning rates and reduces dependence on the initialization process. It can also act as a regularizer to reduce the generalization error and to avoid overfitting without the implementation of dropout layer. To reduce computation, the Batch Normalization is carried out directly after the convolution and before ReLU. ReLU before BN can mess up the calculations due to the non-linear nature of ReLU. Suppose we have network activation as follows:

$$z = g(\omega u + \beta) \tag{7}$$

where ω and β are learned parameters of the model, and $g(\cdot)$ is the nonlinear activation function such as sigmoid or ReLU. Since we normalize $\omega u + \beta$, the bias β can be ignored since its effect will be cancelled by the subsequent mean subtraction. Therefore, Equation (7) is replaced with:

$$z = g(BN(\omega u)) \tag{8}$$

where the BN transform is applied indenpendently to each dimension of $x = \omega u$, with a separate pair of learned parameters.

2.4. Depthwise Separable Convolution

In addition to the implementation of BN, the convolution part can also be further optimized, for example by using depthwise separable convolution (DSC) to reduce the computational costs by reducing the number of arithmetic operations while preserving the same final results [32,34,35]. This technique is applied with changes in filter sizes 3, 1, and 3, respectively. In addition to DSC, the size of the stride in the third convolution should be set to be greater than one, for example 2, 3, or 4, allowing the downsampling process to be implemented within the convolution process, instead of on a special layer such as the MaxPool layer [30]. DSC [34] splits the convolution into two calculation stages, i.e., depthwise and pointwise, as follows:

$$Conv(\omega, y)_{(i,j)} = \sum_{k.l.m}^{K.L.M} \omega_{(k.l.m)} \cdot y_{(i+k,j+l,m)} \tag{9}$$

$$PointwiseConv(\omega, y)_{(i,j)} = \sum_{m}^{M} \omega_m . y_{(i,j,m)} \tag{10}$$

$$DepthwiseConv(\omega, y)_{(i,j)} = \sum_{k,l}^{K,L} \omega_{(k,l)} \bigodot y_{(i+k,j+l)} \tag{11}$$

$$SepConv\Big(\omega_p, \omega_d, y\Big)_{(i,j)} = PointwiseConv_{(i,j)}\Big(\omega_p, DepthwiseConv_{(i,j)}(\omega_d, y)\Big) \tag{12}$$

3. Proposed Ensemble of Depthwise Separable Convolutional Neural Networks

Traditional methods of ECG classification are varied in stages, including four, three, and two. The four stages are beat detection morphology feature extraction, feature dimension reduction, and classification, as stated in [13,36]. The three stages are beat detection, feature detection, and classification [10]. Moreover, the two stages of classification are beat detection and classification [27,37], in which the feature extraction stage is combined with classification. In [38], one stage was used for arrhythmia detection using 34 layer CNNs. However, it cannot be directly compared to our proposed algorithm as they used a different database with 12 heart arrhythmias, sinus rhythm, and noise for a total output of 14 classes. This paper proposes a two-stage ECG beat detection, as one patient might experience a normal beat and another arrhythmia beats as described in the MIT-BIH database. Furthermore, as will be explained in Section 5, the beat detection and segmentation require minimum computational time while improving the classification process. In this section, beat detection and segmentation, and ensemble of Depthwise Separable CNNs consists of around 34,719 train parameters with 21 layer CNNs are explained.

3.1. Beat Detection and Segmentation

In this paper, beat detection, QRS detection, or R-peak detection are performed based on analysis of gradient, amplitude, and duration of the ECG signals similar to [39]. R-peak detection of ECG signals can be done through wavelet transforms with a detection accuracy of more than 99% [39,40]. R-peak detection was performed on 48 records of the MIT-BIH database. With a 360 Hz sampling rate, each record contains 650,000 samples or a duration of 30 min [41]. Each sample is a conversion of a range of 10 mV using an 11-bit ADC [41].

Figure 1 shows examples of ECG pieces of 3000 samples or 8.33 s from the 2nd, 18th, and 36th records. The RR intervals on ECG chunk from the 2nd and 36th files tend to be uniform, while the RR interval on the ECG chunk from the 18th file looks more diverse. Hence, the R-peak detection and the RR interval measurement affect the beat segmentation process.

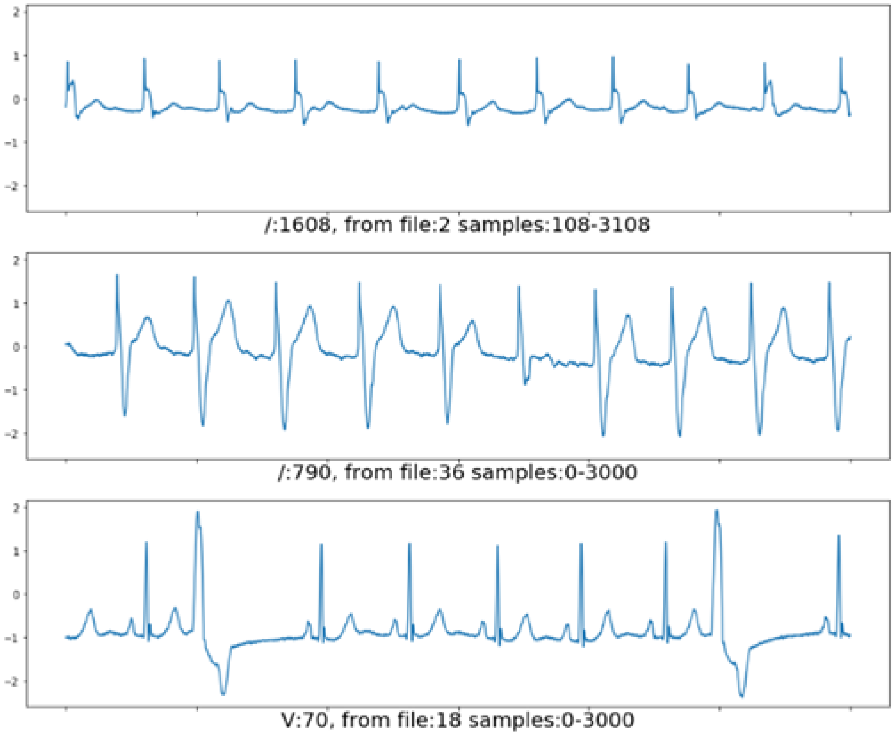


Figure 1. The ECG chunk contains 3000 samples from 3 files, i.e., the 2nd, 18th, and 36th files. The vertical axis is the voltage in mV, while the horizontal axis is time with a scale of 500 samples or 1388 s. The text on the horizontal axis contains the type of beat, and its R-peak position, for example, /: 608, means that the beat is Paced Beat (see Table 1) with R-peak position at the 1608th sample from 650,000 samples in the 2nd file.

In this paper, after R-peak detection, beat segmentation starts from $\frac{1}{4}L_i$ to $\frac{3}{4}R_i$, in which L_i is the RR-interval right before the detected R-peak, while R_i is the RR-interval right after the detected R-peak. This automatic segmentation window is necessary to ensure that there is only a single beat or R-peak in each segment. The maximum number of samples taken for each segment is 256 points which is equivalent to a duration of 256/360 s or 711 ms. The MIT-BIH database used a sampling rate of 360 Hz, so the 256 sample segment size will be more than adequate as the typical RR interval is around 500 ms [42].

Figure 2 shows the example of our proposed automatic beat segmentation. In our segmentation method, if the RR-interval is too short, or there is more than one R-peak at the segment, then zero-paddings are performed to keep the segment size to 256 samples with only one R-peak. If required, this segment size could be downsampled to 128 or even 64 samples to further reduce the computational cost. However, there will be a slight decrease in the classification accuracy. More experiments are conducted on the sample sizes and its accuracy in the next section.

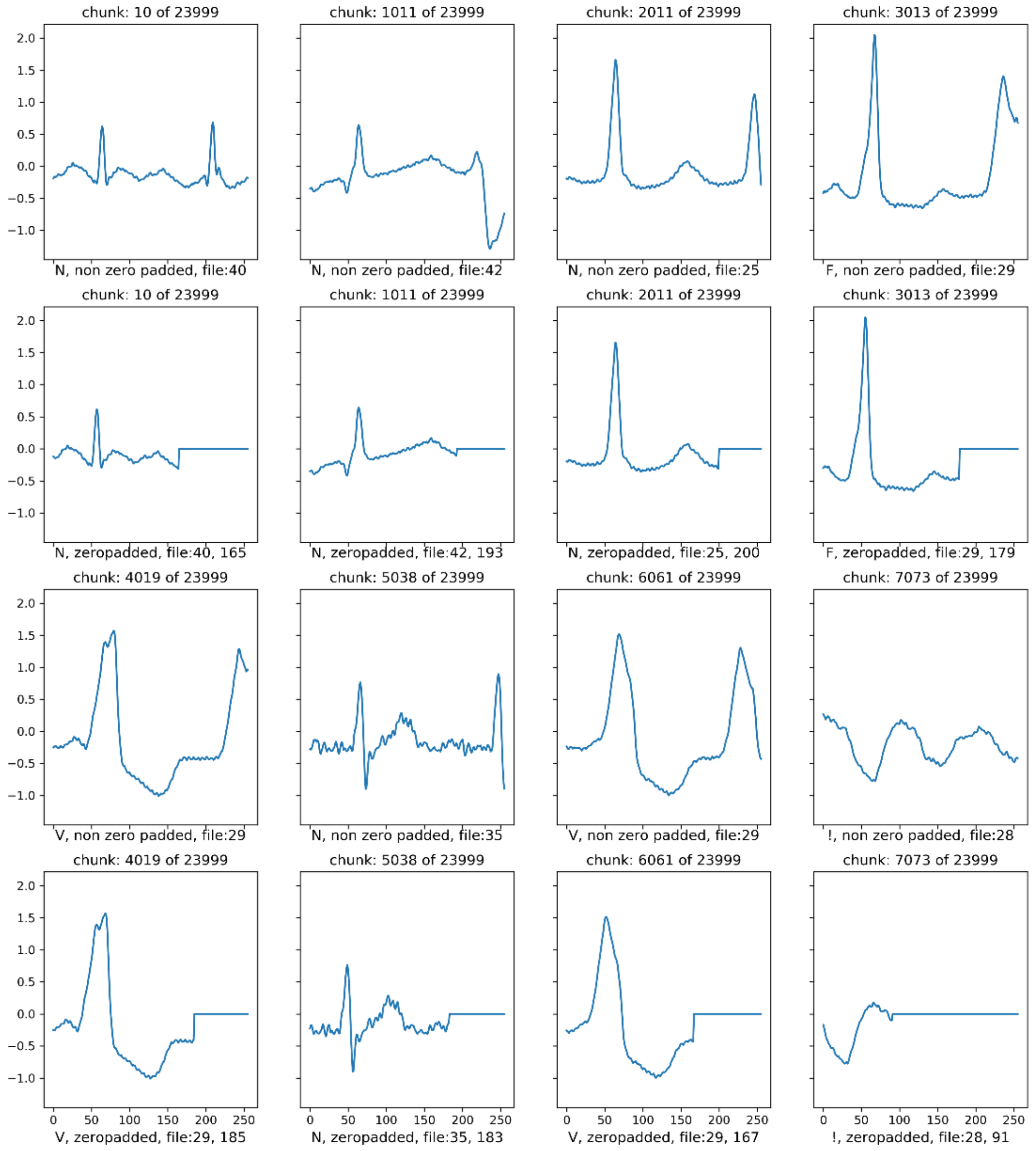


Figure 2. Samples of 16 chunks out of 23,999 beats training data. The 10th chunk, our top left chunk, contains the Normal beat taken from the 40th file. The first and third row were the original signals, while the second and fourth row were the zero-padded signals.

3.2. Ensemble CNNs

Each segmented ECG beat can be replicated into three beat sizes, i.e., 64, 128, and 256 samples. The 64 and 128 samples were the downsampled version of the original 256 beats segmented automatically as explained in Section 3.1. In total, we have three CNN configurations with a difference only in layer 1 (input size), i.e., 64, 128, and 256. After the training and testing phase, the three outputs from three CNNs were ensembled using the averaging method.

Using ensemble CNN by averaging, it can improve further the accuracy by reducing the variance [33]. In our proposed algorithm, we calculate the average of all tensor SoftMax from 3 CNNs as shown in Equation (13) and Figure 3. Let $l = L$ denote the last layer of the CNN model, and s_k^L is the output of the k-th neuron at the output layer, and let $m = 1, \ldots, M$ denotes the number of CNNs to be ensembled. The final output of the ensemble CNN using averaging can be calculated as follows:

$$\overline{P}(s) = \frac{1}{N}\sum_{m=1}^{M} s_k^L[m] \tag{13}$$

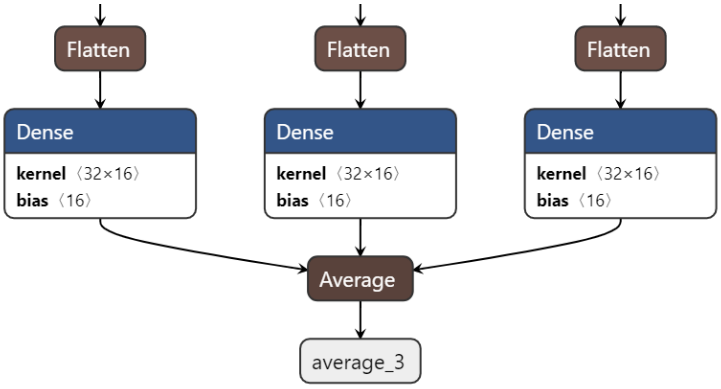


Figure 3. Ensemble CNN.

4. Implementation and Experimental Setup

This section discusses the ECG datasets from the MIT-BIH arrhythmia database, the computational platform used for simulation, and the proposed ensemble CNN algorithm.

4.1. MIT-BIH Arrhythmia Database and Computing Platform

In this paper, ECG datasets from the MIT-BIH arrhythmia database [41] are used for the performance evaluation of our proposed ensemble CNN algorithms. This database is the widely used database for testing classification performance. It contains 48 records, in which each contain two- channel ECG signals for 30 min duration selected from 24 h recording of 47 patients. The database consists of 19 types of ECG beats, in which 16 of them are related to cardiac arrhythmia.

From the total MIT-BIH database, we only used around 110,157 ECG beats out of 112,647 total beats which were categorized into 16 classes as shown in Table 1. These beat samples were then further divided into 23,999 beats for training and 86,158 beats for testing. As described in Figure 2, the total of 110,157 beats was also replicated with zero padding. Therefore, there are two sets of experimental data, with and without zero padding.

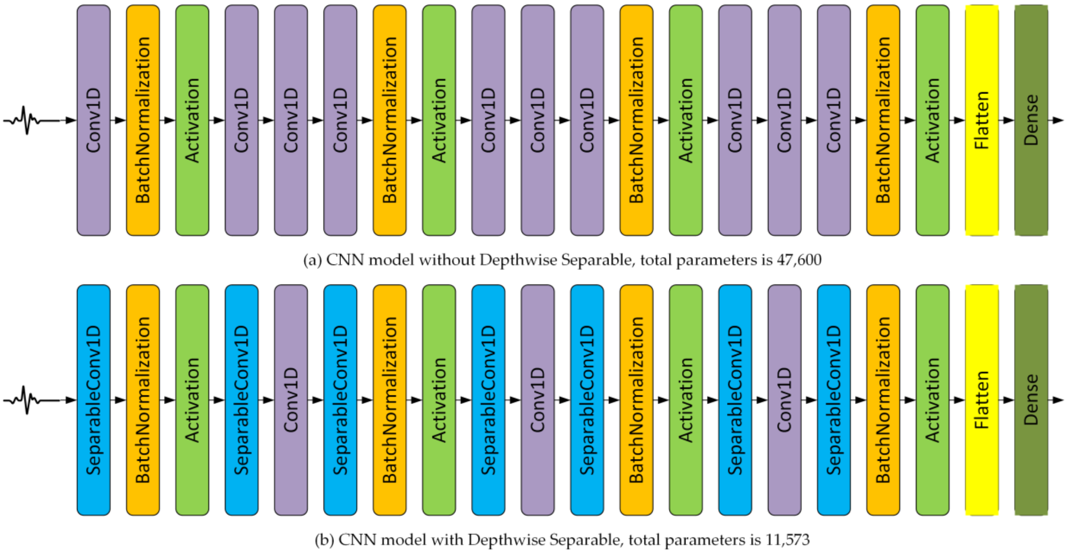
Table 1. Sixteen classes of cardiac arrhythmia ECG beats from the MIT-BIH Database and its amount of Training and Testing beats.

No	Symbol	Annotation Description	Total	Training	Testing
1	N	Normal beat	75,052	11,257	63,795
2	L	Left bundle branch block beat	8075	2826	5249
3	R	Right bundle branch block beat	7259	2540	4719
4	V	Premature ventricular contraction	7130	2495	4635
5	/	Paced beat	7028	2459	4569
6	A	Atrial premature contraction	2546	891	1655
7	f	Fusion of paced and normal beat	982	491	491
8	F	Fusion of ventricular and normal beat	803	401	402
9	!	Ventricular flutter wave	472	236	236
10	j	Nodal (junctional) escape beat	229	114	115
11	x	Non-conducted P-wave	193	96	97
12	a	Aberrated atrial premature beat	150	75	75
13	E	Ventricular escape beat	106	53	53
14	J	Nodal (junctional) premature beat	83	41	42
15	e	Atrial escape beat	16	8	8
16	Q	Unclassifiable beat	33	16	17
		Total 16-class beats	110,157	23,999	86,158
		Extracted from total 112,647 of labeled ECG beats			

The proposed algorithm was implemented in Python with Tensorflow with GPU [43] and Keras libraries [44]. The experiments were performed on a computer with Intel Core i7-7700 CPU with a total of eight logical processors, memory of 8 GBytes, graphic card Nvidia GeForce GTX 1060 6 GB DDR5, using Microsoft Windows 10 64 bits operating system. The experiments on the training and testing time using this computing platform is elaborated further in Section 5.

4.2. Depthwise Separable and Ensemble of Depthwise Separable CNN Models

Using a heuristic approach and optimization, the proposed CNN model contains 21 layers. Figure 4 shows the comparison between the CNN model with and without depthwise separable CNN in terms of structure and number of train parameters. As shown in the figure, the total train parameters are less with the implementation of depthwise separable. Hence, the training time will be faster for depthwise separable CNN. Table 2 shows the depthwise separable CNN model summary along with its total parameters' calculation.



(a) CNN model without Depthwise Separable, total parameters is 47,600

(b) CNN model with Depthwise Separable, total parameters is 11,573

Figure 4. CNN Models with and without the depthwise separable algorithm.

Table 2. Depthwise separable CNN model summary.

#	Layer	Number of Filters, Size, Stride	Output Shape	Number of Parameters
1	input_1		(256, 1)	0
2	separable_conv1d_1	32, 5, 4	(64, 32)	69
3	batch_normalization_1		(64, 32)	128
4	activation_1		(64, 32)	0
5	separable_conv1d_2	32, 5, 1	(64, 32)	1216
6	conv1d_1	32, 1, 1	(64, 32)	1056
7	separable_conv1d_3	32, 5, 4	(16, 32)	1216
8	batch_normalization_2		(16, 32)	128
9	activation_2		(16, 32)	0
10	separable_conv1d_4	32, 5, 1	(16, 32)	1216
11	conv1d_2	32, 1, 1	(16, 32)	1056
12	separable_conv1d_5	32, 5, 4	(4, 32)	1216
13	batch_normalization_3		(4, 32)	128
14	activation_3		(4, 32)	0
15	separable_conv1d_6	32, 5, 1	(4, 32)	1216
16	conv1d_3	32, 1, 1	(4, 32)	1056
17	separable_conv1d_7	32, 5, 4	(1, 32)	1216
18	batch_normalization_4		(1, 32)	128
19	activation_4		(1, 32)	0
20	flatten_1		(32)	0
21	dense_1 (Dense)		(16)	528
Total parameters: 11,573 Trainable parameters: 11,317 Non-trainable parameters: 256				

The depthwise separable CNN as described in Table 2 was used to improve the training time, while the ensemble of depthwise separable CNN, as depicted in Figure 5, was used to improve further its classification accuracy. The input layer size is 256 representing the raw ECG beat waveform, while the output layer size is 16 representing the number of classes as described in the MIT-BIH database (see Table 1). There is a group of layers repeated three times, i.e., layer 5 to 9, layer 10 to 14, and layer 15 to 19. These five layers contain three convolution layers with filter size of 5, 1, and 5, respectively. Layer 5 to 7 is configured according to the DSC algorithm. Moreover, layer 2, 7, 12, and 17 were using stride 4 to replace the function of pooling layer according to the ACN algorithm. Suppose that we use an ensemble of three DSC, the total number of parameters will be three times larger, i.e., 34,719 train parameters, which is still less than the train parameters without DSC, i.e., 47,600. As is discussed in Section 5, the use of ensemble improves its accuracy.

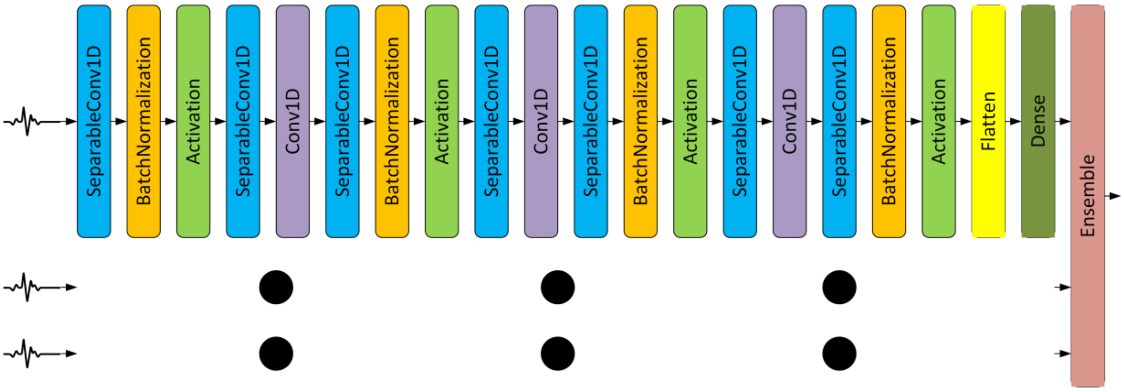


Figure 5. Proposed ensemble of depthwise separable CNNs algorithm.

5. Results and Discussion

This section elaborates beat segmentation and detection, the training process, experiments on the CNN model with and without DSC, the experiments on zero padding and various ensemble configurations, and benchmarking with other algorithms.

5.1. Experiment on ECG Beat Segmentation and Detection

The ECG beat segmentation and detection algorithm described in [39] were implemented in the computing platform described in Section 4. We evaluated the algorithm on the 48 records of the MIT- BIH database. The experiment was repeated 10 times for each record and then the average time was calculated. The experimental result showed that, on average, it requires around 26 µs to detect and segment one ECG beat with zero padding. This result justified the use of two stage ECG classification as proposed in this paper, as the segmentation time is minimal compared to the improvement achieved in the classification stage.

5.2. Training Process

Figure 6 shows the training history for 1000 epochs using the CNN model as described in Table 2 and for the training data described in Table 1. There is a gap of accuracy around 1.5% between training and testing. The accuracy curve for validation shows that the best accuracy can be achieved at epoch of 50. The training accuracy has convergent tendency reaching almost 100% with fluctuation around 99.7%. This result is achieved for a zero-padding input size of 256.

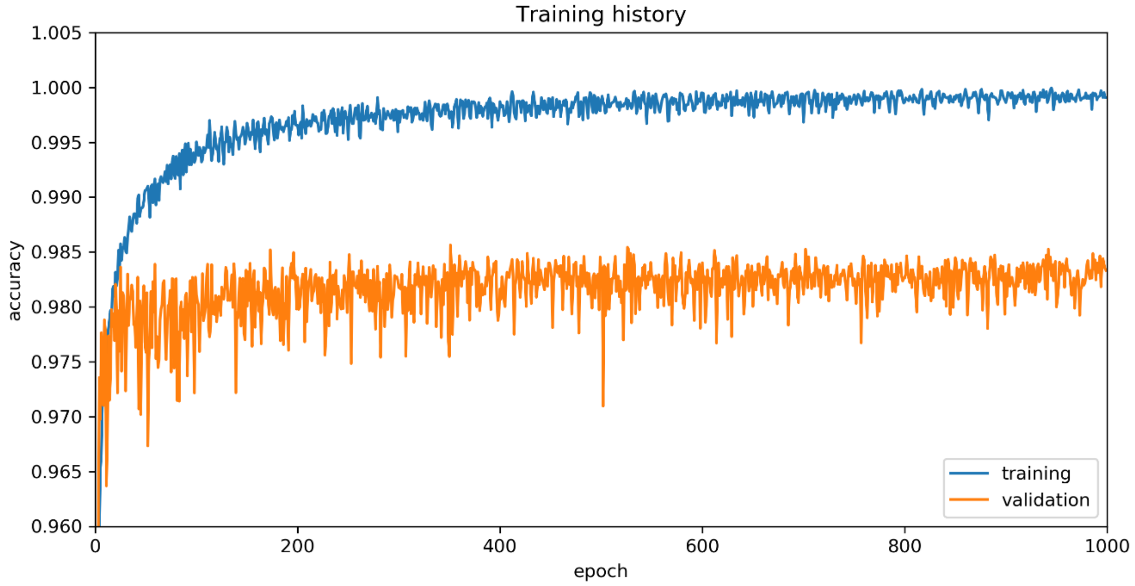
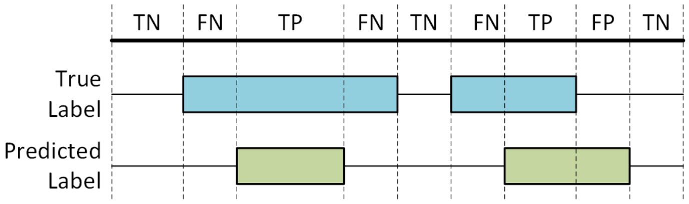


Figure 6. Training history of 1000 epochs iteration of CNN model as described in Table 2. The blue curve is the classification accuracy of the training data. The orange curve is the classification accuracy of the testing data. There is a visible gap around 1.5% between the two curves.

5.3. *Performance Measures*

We used the performance measure of sensitivity, specificity, positive predictivity, and accuracy as shown in Figure 7 and described in [45]. Sensitivity is the rate of correctly classified classes among all classes. Specificity is the rate of correctly classified nonevents among all events. Positive predictivity is the fraction of real events in all detected events. Finally, the accuracy is the percentage of correctly predicted classes.



$$Sn = \frac{TP}{TP+FN}$$ **Sensitivity**

$$Sp = \frac{TN}{TN+FP}$$ **Specificity**

$$Pp = \frac{TP}{TP+FP}$$ **Positive Predictivity**

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$ **Accuracy**

Figure 7. Performance measures using sensitivity, specificity, positive predictivity, and accuracy.

5.4. *On the Effect of Depthwise Separable CNN*

Table 3 shows the experimental result for CNN models with and without depthwise separable algorithms as described in Figure 4 for the input size of 256 samples. The total parameters were 47,600 and 11,573 for without and with DSC algorithm, respectively. The training time for 1000 epochs and the accuracy of both CNN models were comparable. This could be due to the use of GPU during training, in which a faster training time was not apparent in the result. When CPU only was used during training, a faster training time was evident for the CNN model with DSC. Nevertheless, the smaller parameters in the CNN model with DSC provided a faster computation time during the classification stage compared to the model without DSC. Therefore, the CNN model with DSC is used in our next experiments.

Table 3. Effect of DSC implementation on CNN layers

#	DSC	Total Parameters	TN	FN	TP	FP	*Sn*	*Sp*	*Pp*	*Acc*	Training Time	Testing Time
1	Without DSC	47,600	1,291,271	1099	85,059	1099	98.72	99.91	98.72	99.84	711.9 s	81 µs
2	With DSC	11,573	1,290,966	1404	84,754	1404	98.37	99.89	98.37	99.80	745.8 s	62 µs

5.5. On the Effect of Zero Padding

Zero padding in ECG beats was required due to there being a possibility that the distance between RR intervals is less than 256, as well as to avoid more than one R-peak being detected in one segment (see Section 3.1). Furthermore, zero padding has a positive impact on the performance, in terms of sensitivity, specificity, positive predictivity, and accuracy, as evidenced in Table 4. The best performance is highlighted in bold. As shown in Table 4, our proposed ECG segmentation of 256 samples with zero padding has the highest performance. Hence, zero padding is used in our next experiments.

Table 4. Effect of zero padding on various input sizes.

#	Zero Padding	Input Size	TN	FN	TP	FP	*Sn*	*Sp*	*Pp*	*Acc*
1	No	64	1,291,190	1180	84,978	1180	98.63	99.91	98.63	99.83
2	No	128	1,291,413	957	85,201	957	98.89	99.93	98.89	99.86
3	No	256	1,291,411	959	85,199	959	98.89	99.93	98.89	99.86
4	Yes	64	1,291,207	1163	84,995	1163	98.65	99.91	98.65	99.83
5	Yes	128	1,291,389	981	85,177	981	98.86	99.92	98.86	99.86
6	Yes	256	1,291,531	839	85,319	839	**99.03**	**99.94**	**99.03**	**99.88**

5.6. On the Effect of Various Ensemble Configurations

As described in Section 4.2, ensemble of depthwise separable CNNs has been experimented with various input sizes and various numbers of ensembles. The result is shown in Table 5, in terms of sensitivity, specificity, positive predictivity, and accuracy, in which the [256 256 256] ensemble configuration shows the best performance. Hence, the [256 256 256] ensemble configuration is used for benchmarking purposes with other algorithms.

Table 5. Performance of various ensemble of depthwise separable CNN configurations.

#	Ensemble Configuration	TN	FN	TP	FP	*Sn*	*Sp*	*Pp*	*Acc*	Testing Time
1	[128 64]	1,291,194	1176	84,982	1176	98.64	99.91	98.64	99.83	111 µs
2	[256 128]	1,291,369	1001	85,157	1001	98.84	99.92	98.84	99.85	113 µs
3	[256 256]	1,291,460	910	85,248	910	98.94	99.93	98.94	99.87	111 µs
4	[64 64 64]	1,291,207	1163	84,995	1163	98.65	99.91	98.65	99.83	143 µs
5	[128 128 128]	1,291,389	981	85,177	981	98.86	99.92	98.86	99.86	146 µs
6	[256 256 256]	1,291,531	839	85,319	839	**99.03**	**99.94**	**99.03**	**99.88**	150 µs
7	[256 128 64]	1,291,408	962	85,196	962	98.88	99.93	98.88	99.86	150 µs

In the ensemble configuration, the CNN model was trained separately. For example, the [256 256 256] ensemble is trained three times. Therefore, the training time is increased by three times with increased accuracy. As expected, the classification stage (testing time) will be increased around three times as well but within the range of 150 µs. Hence, we can conclude that the proposed two stage ECG classification requires around 180 µs for QRS detection and classification using the computing platform described in Section 4.1.

For more detailed evaluation, the [256 256 256] ensemble confusion matrix is shown in Table 6. The performance assessment of each class using the [256 256 256] ensemble configuration can be seen in Table 7, in which the ECG class is sorted from the largest to the smallest available samples in the MIT-BIH database. As can be seen from the table, fewer samples of data cause inaccurate classification.

In conclusion, the [256 256 256] ensemble achieves the best performance in terms of sensitivity, specificity, positive predictivity, and accuracy.

Table 6. Confusion matrix for [256 256 256] ensemble of CNNs.

		Prediction															
		N	**L**	**R**	**V**	**/**	**A**	**f**	**F**	**!**	**j**	**x**	**a**	**E**	**J**	**e**	**Q**
Ground Truth		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
N	0	63,521	4	13	70	2	79	11	36	3	44	6	2	0	1	0	3
L	1	22	5211	0	13	1	0	0	0	0	0	0	0	1	0	0	1
R	2	11	0	4699	0	0	9	0	0	0	0	0	0	0	0	0	0
V	3	46	10	0	4528	1	11	0	22	10	0	0	7	0	0	0	0
/	4	3	0	0	3	4553	0	9	0	0	0	0	0	1	0	0	0
A	5	143	0	5	16	0	1484	0	2	0	3	0	0	0	2	0	0
f	6	13	0	0	1	6	0	470	0	0	0	0	0	0	1	0	0
F	7	52	0	0	36	0	0	0	313	0	0	1	0	0	0	0	0
!	8	2	0	0	8	0	0	0	0	225	0	0	1	0	0	0	0
j	9	15	0	1	0	0	0	0	0	0	97	0	0	0	2	0	0
x	10	1	0	0	1	0	0	0	0	5	0	89	1	0	0	0	0
a	11	6	0	0	15	0	9	0	0	1	0	0	44	0	0	0	0
E	12	2	0	0	1	0	0	0	0	0	0	0	0	50	0	0	0
J	13	7	0	0	0	0	3	0	0	0	0	0	0	0	32	0	0
e	14	3	0	0	0	0	1	1	0	0	0	0	0	0	0	3	0
Q	15	11	0	2	1	0	0	3	0	0	0	0	0	0	0	0	0

Table 7. Performance evaluation for each class using [256 256 256] ensemble of depthwise separable CNNs.

ECG Class	**Total Beats**	**Train Beats**	**Test Beats**	**TN**	**FN**	**TP**	**FP**	***Sn***	***Sp***	***Pp***	***Acc***
N	75,052	11,257	63,795	22,026	274	63,521	337	99.57	98.49	99.47	99.29
L	8075	2826	5249	80,895	38	5211	14	99.28	99.98	99.73	99.94
R	7259	2540	4719	81,418	20	4699	21	99.58	99.97	99.56	99.95
V	7130	2495	4635	81,358	107	4528	165	97.69	99.80	96.48	99.68
/	7028	2459	4569	81,579	16	4553	10	99.65	99.99	99.78	99.97
A	2546	891	1655	84,391	171	1484	112	89.67	99.87	92.98	99.67
f	982	491	491	85,643	21	470	24	95.72	99.97	95.14	99.95
F	803	401	402	85,696	89	313	60	77.86	99.93	83.91	99.83
!	472	236	236	85,903	11	225	19	95.34	99.98	92.21	99.97
j	229	114	115	85,996	18	97	47	84.35	99.95	67.36	99.92
x	193	96	97	86,054	8	89	7	91.75	99.99	92.71	99.98
a	150	75	75	86,072	31	44	11	58.67	99.99	80.00	99.95
E	106	53	53	86,103	3	50	2	94.34	100.00	96.15	99.99
J	83	41	42	86,110	10	32	6	76.19	99.99	84.21	99.98
e	16	8	8	86,150	5	3	0	37.5	100.00	100.00	99.99
Q	33	16	17	86,137	17	0	4	0	100.00	0.00	99.98
Σ	**110,157**	**23,999**	**86,158**	**1,291,531**	**839**	**85,319**	**839**	**99.03**	**99.94**	**99.03**	**99.88**

5.7. On Comparison with Other Algorithms

Table 8 shows the benchmarking results with another 10 algorithms, in terms of number of classes (some of the algorithms did not classify all 16 classes), methods, number of prediction stages, and accuracy. In terms of accuracy, refs. [46] and [47] have the closest but still a lower accuracy than our proposed algorithm, i.e., 99.61% and 99.80%, respectively. Note that, ref. [46] only performs classification for eight classes, while [47] only performs classification for four classes, while we perform classification for sixteen classes. Hence, it is evident that our proposed algorithm with the [256 256 256] ensemble of CNNs outperforms the other algorithms. Particularly, we managed to classify all 16 classes from the MIT-BIH database, while reducing the number of stages to two (lower computational cost), and achieving highest accuracy.

Table 8. Benchmarking of the proposed algorithm with other algorithms using MIT-BIH database.

No	Author, Year	#Class	Methods	Prediction Stage	Accuracy
1	Melgani and Bazi, 2008 [10]	6	SVM and PSO	3	89.72%
2	Ince et al., 2009 [36]	5	DWT, PCA, and ANN	4	98.30%
3	Wen et al., 2009 [37]	16	Self Organizing CMAC Neural Network	2	98.21%
4	Sarfraz et al., 2014 [46]	8	ICA and BPNN	3	99.61%
5	Kiranyaz et al., 2015 [27]	5	1D-CNN	2	95.14%
6	Raj and Ray, 2017 [13]	16	DCT_DOST, PCA, SVM_PSO	4	98.82%
7	Nanjun and Meshram, 2018 [48]	2	DWT and DNN	3	98.33%
8	Zhai and Tin, 2018 [29]	5	2D-CNN	2	96.05%
9	Rangappa and Agarwal, 2018 [49]	2	k-NN	3	98.40%
10	Xia et al., 2018 [47]	4	SDAE, DNN	4	99.80%
11	Proposed Algorithm	16	Ensemble CNNs	2	**99.88%**

6. Conclusions

In this paper, we presented an efficient algorithm for cardiac arrhythmia classification using the ensemble of depthwise separable convolutional neural networks. First, we optimized the beat segmentation by taking ECG samples centered around the R-peak. Second, we used all convolutional network, batch normalization, and depthwise separable convolution, to achieve the best accuracy while reducing the computational cost. Finally, we ensembled three depthwise separable CNNs by averaging three CNNs of 256 sample input size. Performance evaluation showed that our proposed algorithms achieved around 99.88% accuracy in 16 classes classification. The proposed two-stage ECG classification required around 180 µs, which can be implemented in a real time application. Future work will include the implementation of the current CNN on GPU to speed up its training, as well as to vary the input segment for various patients, the use of different databases, the use of other optimization methods, and the implementation in clinical application validated by a cardiologist.

Author Contributions: Formal analysis, E.I.; Funding acquisition, K.R.; Methodology, E.I.; Supervision, K.R.; Writing—original draft, E.I.; Writing—review and editing, K.R., D.S. and T.S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This article publication is supported by Universitas Indonesia through Q1Q2 International Journal Publication Grant Scheme under the contract number NKB-0306/UN2.R3.1/HKP.05.00/2019.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Evans, G.F.; Shirk, A.; Muturi, P.; Soliman, E.Z. Feasibility of using mobile ECG recording technology to detect atrial fibrillation in low-resource settings. *Glob. Heart* **2017**, *12*, 285–289. [CrossRef] [PubMed]
2. Sun, W.; Zeng, N.; He, Y. Morphological Arrhythmia Automated Diagnosis Method Using Gray-Level Co-occurrence Matrix Enhanced Convolutional Neural Network. *IEEE Access* **2019**, *7*, 67123–67129. [CrossRef]
3. Laguna, P.; Cortés, J.P.M.; Pueyo, E. Techniques for ventricular repolarization instability assessment from the ECG. *Proc. IEEE* **2016**, *104*, 392–415. [CrossRef]
4. Satija, U.; Ramkumar, B.; Manikandan, M.S. A New Automated Signal Quality-Aware ECG Beat Classification Method for Unsupervised ECG Diagnosis Environments. *IEEE Sens. J.* **2018**, *19*, 277–286. [CrossRef]

5. Lynn, H.M.; Pan, S.B.; Kim, P. A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks. *IEEE Access* **2019**, *7*, 145395–145405. [CrossRef]
6. Chu, Y.; Shen, H.; Huang, K. ECG Authentication Method Based on Parallel Multi-Scale One-Dimensional Residual Network With Center and Margin Loss. *IEEE Access* **2019**, *7*, 51598–51607. [CrossRef]
7. Kim, H.; Chun, S.Y. Cancelable ECG Biometrics Using Compressive Sensing-Generalized Likelihood Ratio Test. *IEEE Access* **2019**, *7*, 9232–9242. [CrossRef]
8. Dokur, Z.; Olmez, T.; Yazgan, E. Comparison of discrete wavelet and Fourier transforms for ECG beat classification. *Electron. Lett.* **1999**, *35*, 1502–1504. [CrossRef]
9. Banerjee, S.; Mitra, M. Application of cross wavelet transform for ECG pattern analysis and classification. *IEEE Trans. Instrum. Meas.* **2013**, *63*, 326–333. [CrossRef]
10. Melgani, F.; Bazi, Y. Classification of electrocardiogram signals with support vector machines and particle swarm optimization. *IEEE Trans. Inf. Technol. Biomed.* **2008**, *12*, 667–677. [CrossRef]
11. Venkatesan, C.; Karthigaikumar, P.; Paul, A.; Satheeskumaran, S.; Kumar, R. ECG signal preprocessing and SVM classifier-based abnormality detection in remote healthcare applications. *IEEE Access* **2018**, *6*, 9767–9773. [CrossRef]
12. Chen, X.; Wang, Y.; Wang, L. Arrhythmia Recognition and Classification Using ECG Morphology and Segment Feature Analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2018**, *16*, 131–138.
13. Raj, S.; Ray, K.C. ECG signal analysis using DCT-based DOST and PSO optimized SVM. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 470–478. [CrossRef]
14. Pasolli, E.; Melgani, F. Active learning methods for electrocardiographic signal classification. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 1405–1416. [CrossRef]
15. Li, Z.; Feng, X.; Wu, Z.; Yang, C.; Bai, B.; Yang, Q. Classification of Atrial Fibrillation Recurrence Based on a Convolution Neural Network With SVM Architecture. *IEEE Access* **2019**, *7*, 77849–77856. [CrossRef]
16. Zigel, Y.; Cohen, A.; Katz, A. The weighted diagnostic distortion (WDD) measure for ECG signal compression. *IEEE Trans. Biomed. Eng.* **2000**, *47*, 1422–1430.
17. Gerencsér, L.; Kozmann, G.; Vago, Z.; Haraszti, K. The use of the SPSA method in ECG analysis. *IEEE Trans. Biomed. Eng.* **2002**, *49*, 1094–1101. [CrossRef]
18. Rahman, Q.A.; Tereshchenko, L.G.; Kongkatong, M.; Abraham, T.; Abraham, M.R.; Shatkay, H. Utilizing ECG-based heartbeat classification for hypertrophic cardiomyopathy identification. *IEEE Trans. Nanobiosci.* **2015**, *14*, 505–512. [CrossRef]
19. Sopic, D.; Aminifar, A.; Aminifar, A.; Atienza, D. Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE Trans. Biomed. Circuits Syst.* **2018**, *12*, 982–992. [CrossRef]
20. Lai, D.; Zhang, Y.; Zhang, X. An automated strategy for early risk identification of sudden cardiac death by using machine learning approach on measurable arrhythmic risk markers. *IEEE Access* **2019**, *7*, 94701–94716. [CrossRef]
21. Rad, A.B.; Eftestol, T.; Engan, K.; Irusta, U.; Kvaloy, J.T.; Kramer-Johansen, J.; Wik, L.; Katsaggelos, A.K. ECG-based classification of resuscitation cardiac rhythms for retrospective data analysis. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 2411–2418. [CrossRef] [PubMed]
22. Fira, C.M.; Goras, L. An ECG signals compression method and its validation using NNs. *IEEE Trans. Biomed. Eng.* **2008**, *55*, 1319–1326. [CrossRef] [PubMed]
23. Mar, T.; Zaunseder, S.; Martínez, J.P.; Llamedo, M.; Poll, R. Optimization of ECG classification by means of feature selection. *IEEE Trans. Biomed. Eng.* **2011**, *58*, 2168–2177. [CrossRef]
24. Bouaziz, F.; Oulhadj, H.; Boutana, D.; Siarry, P. Automatic ECG arrhythmias classification scheme based on the conjoint use of the multi-layer perceptron neural network and a new improved metaheuristic approach. *IET Signal Process.* **2019**, *13*, 726–735. [CrossRef]
25. Huang, J.; Chen, B.; Yao, B.; He, B. ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network. *IEEE Access* **2019**, *7*, 92871–92880. [CrossRef]
26. Mai, V.; Khalil, I.; Meli, C. ECG biometric using multilayer perceptron and radial basis function neural networks. In Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 30 August–3 September 2011.
27. Kiranyaz, S.; Ince, T.; Gabbouj, M. Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Trans. Biomed. Eng.* **2015**, *63*, 664–675. [CrossRef]

28. Nurmaini, S.; Partan, R.U.; Caesarendra, W.; Dewi, T.; Rahmatullah, M.N.; Darmawahyuni, A.; Bhayyu, V.; Firdaus, F. An Automated ECG Beat Classification System Using Deep Neural Networks with an Unsupervised Feature Extraction Technique. *Appl. Sci.* **2019**, *9*, 2921. [CrossRef]
29. Zhai, X.; Tin, C. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access* **2018**, *6*, 27465–27472. [CrossRef]
30. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv* **2014**, arXiv:1412.6806.
31. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
32. Kaiser, L.; Gomez, A.N.; Chollet, F. Depthwise separable convolutions for neural machine translation. *arXiv* **2017**, arXiv:1706.03059,.
33. Polikar, R. Ensemble learning. In *Ensemble Machine Learning*; Springer: Berlin, Germany, 2012; pp. 1–34.
34. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
35. Zhang, T.; Zhang, X.; Shi, J.; Wei, S. Depthwise Separable Convolution Neural Network for High-Speed SAR Ship Detection. *Remote Sens.* **2019**, *11*, 2483. [CrossRef]
36. Ince, T.; Kiranyaz, S.; Gabbouj, M. A generic and robust system for automated patient-specific classification of ECG signals. *IEEE Trans. Biomed. Eng.* **2009**, *56*, 1415–1426. [CrossRef] [PubMed]
37. Wen, C.; Lib, T.-C.; Chang, K.-C.; Huang, C.-H. Classification of ECG complexes using self-organizing CMAC. *Measurement* **2009**, *42*, 399–407. [CrossRef]
38. Rajpurkar, P.; Hannun, A.Y.; Rajpurkar, P.; Haghpanahi, M.; Tison, G.H.; Bourn, C.; Turakhia, M.P.; Ng, A.Y. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv* **2017**, arXiv:1707.01836.
39. Pan, J.; Tompkins, W.J. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* **1985**, *32*, 230–236. [CrossRef]
40. Li, C.; Zheng, C.; Tai, C. Detection of ECG characteristic points using wavelet transforms. *IEEE Trans. Biomed. Eng.* **1995**, *42*, 21–28.
41. Moody, G.B.; Mark, A.R.G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. [CrossRef]
42. Dupre, A.; Vincent, S.; Iaizzo, P.A. Basic ECG Theory, Recordings, and Interpretation. In *Handbook of Cardiac Anatomy, Physiology, and Devices*; Springer: Berlin, Germany, 2005; pp. 191–201.
43. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI16), Savannah, GA, USA, 2–4 November 2016.
44. Chollet, F. Keras: Deep Learning Library for Theano and Tensorflow. 2015. Available online: https://keras.io/ (accessed on 20 May 2019).
45. Hu, Y.H.; Palreddy, S.; Tompkins, W.J. A patient-adaptable ECG beat classifier using a mixture of experts approach. *IEEE Trans. Biomed. Eng.* **1997**, *44*, 891–900.
46. Sarfraz, M.; Khan, A.A.; Li, F.F. Using independent component analysis to obtain feature space for reliable ECG Arrhythmia classification. In Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, UK, 2–5 November 2014.
47. Xia, Y.; Zhang, H.; Xu, L.; Gao, Z.; Zhang, H.; Liu, H.; Li, S. An automatic cardiac arrhythmia classification system with wearable electrocardiogram. *IEEE Access* **2018**, *6*, 16529–16538. [CrossRef]
48. Nanjundegowda, R.; Meshram, V.A. Arrhythmia Detection Based on Hybrid Features of T-wave in Electrocardiogram. *Int. J. Intell. Eng. Syst.* **2018**, *11*, 153–162. [CrossRef]
49. Rangappa, V.G.; Prasad, S.V.A.V.; Agarwal, A. Classification of Cardiac Arrhythmia stages using Hybrid Features Extraction with K-Nearest Neighbour classifier of ECG Signals. *Learning* **2018**, *11*, 21–32.