Computer Methods and Programs in Biomedicine 184 (2020) 105286

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Multi-branch fusion network for Myocardial infarction screening from 12-lead ECG images

Pengyi Hao[a], Xiang Gao[a], Zhihe Li[a], Jinglin Zhang[b,1,*], Fuli Wu[a], Cong Bai[a]

[a] College of Computer Science and Technology, Zhejiang University of Technology, hangzhou, China
[b] School of Atmospheric Science, Nanjing University of Information Science, China

ARTICLE INFO

Article history:
Received 8 September 2019
Revised 8 December 2019
Accepted 12 December 2019

Keywords:
Myocardial infarction
ECG Images
Multi-branch fusion network

ABSTRACT

Background and Objective: Myocardial infarction (MI) is a myocardial anoxic incapacitation caused by severe cardiovascular obstruction that can cause irreversible injury or even death. In medical field, the electrocardiogram (ECG) is a common and effective way to diagnose myocardial infarction, which often requires a wealth of medical knowledge. It is necessary to develop an approach that can detect the MI automatically.

Methods: In this paper, we propose a multi-branch fusion framework for automatic MI screening from 12-lead ECG images, which consists of multi-branch network, feature fusion and classification network. First, we use text detection and position alignment to automatically separate twelve leads from ECG images. Then, those 12 leads are input into the multi-branch network constructed by a shallow neural network to get 12 feature maps. After concatenating those feature maps by depth fusion, classification is explored to judge the given ECG is MI or not.

Results: Based on extensive experiments on an ECG image dataset, performances of different combinations of structures are analyzed. The proposed network is compared with other networks and also compared with physicians in the practical use. All the experiments verify that the proposed method is effective for MI screening based on ECG images, which achieves accuracy, sensitivity, specificity and F1-score of 94.73%, 96.41%, 95.94% and 93.79% respectively.

Conclusions: Rather than using the typical one-dimensional electrical ECG signal, this paper gives an effective model to screen MI by analyzing 12-lead ECG images. Extracting and analyzing these 12 leads from their corresponding ECG images is a good attempt in the application of MI screening.

1. Introduction

Myocardial infarction (MI), also known as heart attack, is a desperate form of coronary artery disease that arises from clogged arteries of the heart [1]. Infarction means the death of tissue due to lack of blood supply [2]. Because MI can lead to irreversible heart damage or even to death with symptoms comprising chest pain, difficulty breathing and irregular heartbeats typically, it is critical to screen and treat the disease at an early stage [3]. The common ways of myocardial infarction screening are based on ECG, blood test and angiography. ECG is a very important diagnostic assistant for cardiac diseases [4]. ECG is a visual signal captured or measured by placing electrodes on the surface of the body to detect voltage changes. By recording over a period of time, ECG can represent potential cardiac abnormalities.

Usually, if there are rises in the ST segments, ST-segment shift or bias, changes in the shape or flipping of T waves, or new Q waves, it may be MI [5]. Nevertheless, it is ambiguous because some other heart diseases may also lead to such bias. Therefore, traditional clinical diagnosis requires expertise, experience, and much effort from doctors. For this reason, computer-aided detection (CADe) methods have been researched.

Typically, ECG is one-dimensional voltage amplitude data representing as time-series signals. In the fields of CADe with one-dimensional signals, classical machine learning algorithms are always used, such as support vector machine (SVM) [6], classification tree [7] and hidden markov model (HMMs) [8]. For example, Ubeyli [9] extracted features of four types of ECG beats by eigen-

* Corresponding author. Tel.: +008613022573330.
E-mail addresses: haopy@zjut.edu.cn (P. Hao), gaoxiang1001@outlook.com (X. Gao), spooseageese@gmail.com (Z. Li), jinglin.zhang@nuist.edu.cn (J. Zhang), fuliwu@zjut.edu.cn (F. Wu), congbai@zjut.edu.cn (C. Bai).
[1] ORCID(s): 0000-0001-7499-1992

https://doi.org/10.1016/j.cmpb.2019.105286
0169-2607/

vector methods and then input the statistical features into recurrent neural networks to classify ECG signals. Sharma et al. [10] extracted features of ECG beats from time-frequency representation by using improved eigenvalue decomposition of Hankel matrix and Hilbert transform, and applied random forest classifier in order to detect coronary avert disease. Padmavathi et al. [11] used magnitude squared coherence technique and SVM with kernel function to classify the inferior myocardial infarction. Sun et al. [12] proposed a method to classify myocardial infarction by applying and improving multiple instance learning, in which ECG signals were mapped into a topic space defined by a number of topics identified over all the unlabeled heartbeats and SVM was used to the ECG-level topic vectors. In practice, these classical methods did not perform well due to several reasons. One is that the features extracted in these methods are always manually selected which cannot be robust for different application environments. Besides, the noises like baseline wanders, power line interference, and muscle contraction, make these methods difficult to get stable performances.

In recent years, the research on deep learning is gaining momentum and showing encouraging results in detection and classification tasks of medical images [13]. From the perspective of deep learning, some researches treated the one-dimensional ECG signals as the problem of time-series classification. For example, Xia et al. [14] proposed an automatic wearable ECG classification and monitoring system with staked denoising autoencoder. Acharya et al. [15] implemented a convolutional neural network for the classification of normal and MI using ECG signals. Huang et al. [16] transformed five types of heart beats' signals into time-frequency spectrograms and then trained a 2D-CNN for classifying arrhythmia types. Pourbabaee et al.[17] learned features from time domain ECG signals by using a CNN with one fully connected layer for screening paroxysmal atrial fibrillation patients. Liu et al. [18] employed fuzzy information granulation to get beat segments from raw ECG signal data with multilead, and then optimized a multilead-CNN targeting mobile ECG healthcare system. Strodthoff & Strodthoff [19] extracted random subsequences from the original time series, and fed them into a 1D fully convolutional neural network. Baloglua et al [20] trained an 1D convolutional neural network using each lead of 12-lead ECG signals for automated detection of MI. Compared with classical methods, the CNN-based models can be a better choice. However, in the real application, there are several limitations. First, time domain ECG signals are usually hard to be gained out of the hospital. Second, for different hospitals they may use different ECG equipments, resulting in non-uniform formats of ECG signals.

In the real life, it is difficult to get the original ECG signals, but it is easy to collect the printed ECGs or screenshots of ECGs. When we scan the printed ECGs, digital images about ECG can be obtained. Concerning about ECG images, however, the research is relatively rare. Lu et al. [21] created 2D images by connecting the dots of the 1D single heartbeat signals, then PQRST features were extracted. Ji et al. [22] also converted 1D ECG beats to be 2D images, then used faster R-cnn to classify the ECG beats. Another work was presented by Jun et al. [23], they directly converted the 1D ECG signals to be 2D images, then a two-dimensional CNN was used to classify these grayscale ECG images for detecting arrhythmia. Although there were some research work exploring ECG images, they used the clean ECG images converted from 1D signals. For the real use, we always have ECG images with diverse colors, resolutions and qualities.

In this paper, we present a novel end-to-end approach named as multi-branch fusion framework for myocardial infarction screening from 12-lead ECG images like printed ECGs or screenshots of ECGs. It consists of multi-branch network, fusion module and classification module. In order to exploit as many features from images as possible, this paper gives a text detection and position alignment method to divide ECG images into 12 pieces evenly base on 12 leads, then an architecture including 12 individual networks extracts features from each ECG lead respectively. After that, features are fused in the dimension of depth and fed into the classification model. Performances of different structures of multi-branch network, fusion model and classification model are analyzed. The proposed network is compared with other networks and also compared with physicians in the practical use. All the experiments verify that the proposed method is effective for MI screening from ECG images. The main contributions of this study are as follows. (1) Rather than using one-dimensional electrical signals, we utilize and analyze ECG images like printed ECGs and screenshots of ECGs, which is not limited to different ECG devices and their different sampling rates. (2) A simple but effective text detection and position alignment method is proposed to automatically crop and label 12 leads from ECG images, which is much similar to the physician's diagnosis process. (3) We put forward a novel CNN-based architecture consisting of the multi-branch network, feature fusion and dense block-based classification. This architecture enables features of ECG images to be extracted effectively and reaches human-level performance with the trade-off between high performance and simple prepossessing of the limited data.

2. Method

2.1. Problem formulation

The task of MI screening is a binary classification task which takes the 12 leads (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6) of ECG images as input. Suppose that there are n ECG images in the training set. It can be described as $X = \{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\}$. And X_i is processed into twelve bounding boxes. More specifically, $X_i = \{x_i^1, x_i^2, x_i^3, x_i^4, x_i^5, x_i^6, x_i^7, x_i^8, x_i^9, x_i^{10}, x_i^{11}, x_i^{12}\}$ is obtained by cropping the 12 leads from its corresponding ECG image. The label Y_i is 1 when the ECG image shows MI. Otherwise, Y_i is 0.

For n instances in the training set, we choose the cross-entropy function as the cost function, which is defined as

$$-\frac{1}{n}\sum_{i=0}^{n}[Y_i lna + (1 - Y_i) ln(1 - a)] \quad (1)$$

where a is the probability calculated by the input X_i, and Y_i is its true label.

2.2. Model architecture

In this section, MI screening framework is created. Fig. 1 shows a flow graph of the entire framework. It is composed of leads extraction, multi-branch network, feature fusion, and classification network. One combination of these structures is shown in Fig. 2 to present the details.

2.2.1. Multi-branch network

In order to make better use of the information of each lead, we design a multi-branch network to extract features of each lead. Multi-branch network consists of twelve independent branches. For each branch, the same feature extraction network is included. Here we will try the following three structures, as shown in Fig. 3.

(a) Residual block.

This structure was first proposed in [24], which used the idea of identity mapping to construct the deep neural network. In this model, we construct three residual structures based on shortcut. The output channels of them are set as 8, 16 and 32. Each residual

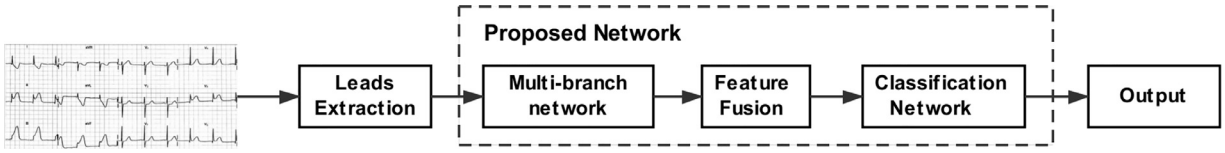

Fig. 1. The flow graph of the proposed approach.

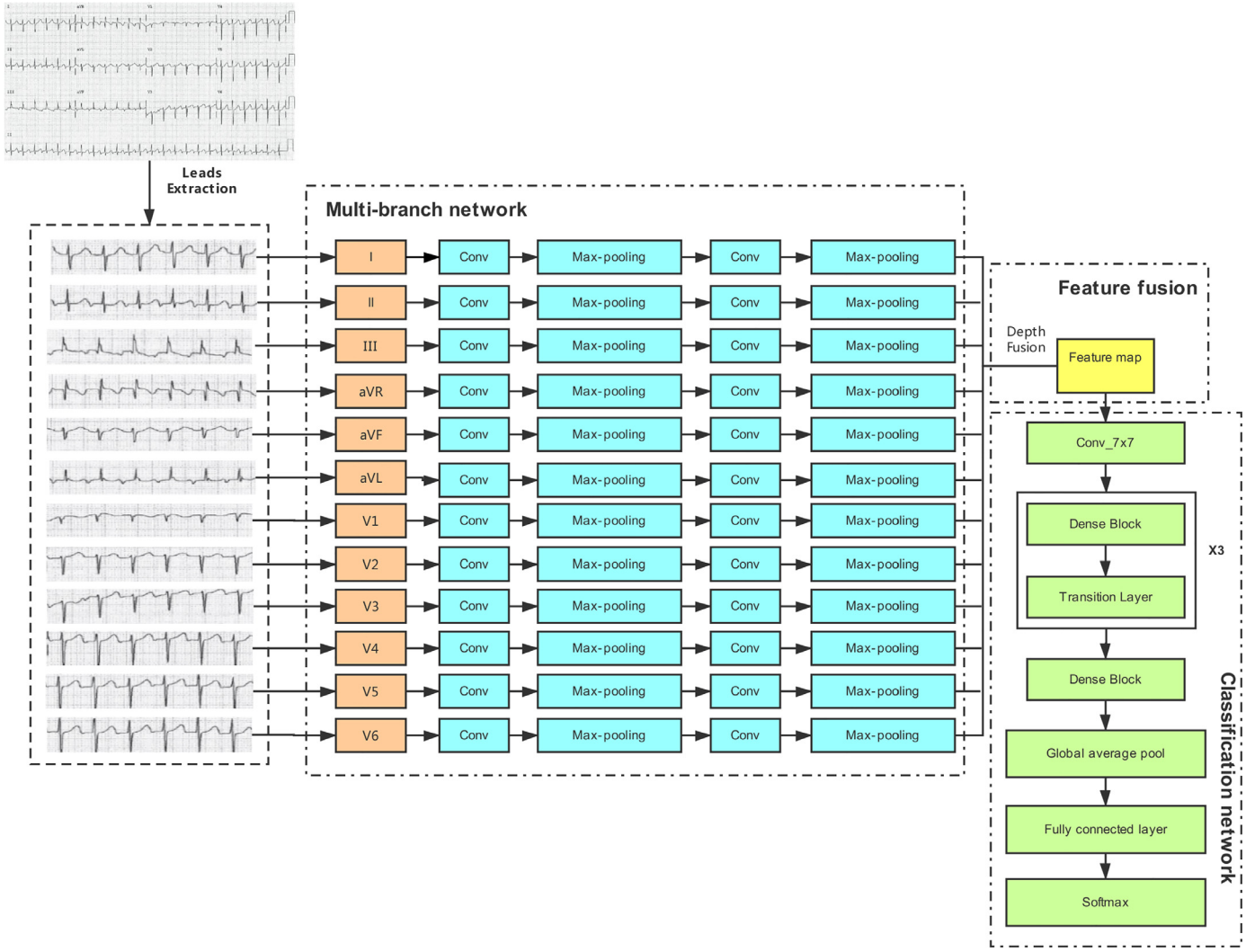

Fig. 2. One architecture of the proposed network.

structure is composed of three 2-layer residual blocks. Each residual block includes two 3 × 3 convolutional layers with the same output channels. When the dimensions increase, the shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions.

(b) Dense block.

This structure was first proposed in [25], which connected each layer to every layer in a feed-forward fashion. In this structure, we construct two dense blocks to extract the features of images. The first block has 6 bottleneck layers while the other one has 12 bottleneck layers. Each bottleneck layer has a dropout rate of 0.2 and a growth rate of 24. Additionally, there is a transition layer after each dense block to reduce the parameters.

(c) Shallow neural network.

This network has very few layers, only consisting of two 3 × 3 convolutional layers and two max-pooling layers. To be more specific, each convolutional layer has a filter length of 3 and a stride of 1. The input of each branch is a 128 × 128 × 3 matrix of corresponding single lead, then each branch outputs a 32 × 32 × 32 feature map.

2.2.2. Feature fusion

In order to integrate the information extracted from the multi-branch network, we need to conduct feature fusion on those feature maps. There are two strategies, one is depth fusion, the other is length fusion. Depth fusion concatenates the feature map in the dimension of depth while length fusion concatenates the feature map in the dimension of length. The output of feature fusion will be fed into the classification network.

Taking the shallow neural network as the multi-branch network, its outputs are 32 × 32 × 32 feature maps. If using depth fusion, the feature map of 32 × 32 × 384 is obtained by fusing 12 feature maps in depth. If using length fusion, the feature map of 384 × 32 × 32 is obtained by fusing 12 feature maps in length.

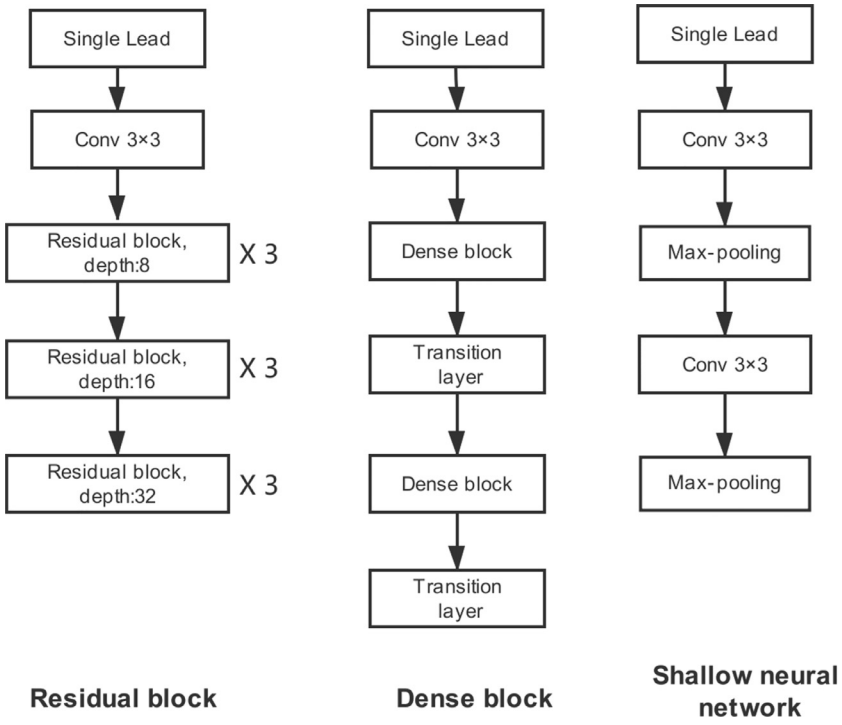

Fig. 3. Three structures of feature extraction in each branch of multi-branch network.

Table 1
Details in classification network based on ResNet

Layers	Type	Inner structure		Output shape	Kernel size	Stride
0	Input	- - - - - - - -		32 × 32 × 384	- - - - -	- - -
1	Convolution	Convolutional layer		32 × 32 × 16	3 × 3	1
2	Residual Block	Convolutional layer Convolutional layer	× 2	32 × 32 × 16	3 × 3 3 × 3	1 1
3	Pooling	Average pooling		16 × 16 × 16	2 × 2	2
4	Residual Block	Convolutional layer Convolutional layer	× 2	16 × 16 × 32	3 × 3 3 × 3	1 1
5	Pooling	Average pooling		8 × 8 × 32	2 × 2	2
6	Residual Block	Convolutional layer Convolutional layer	× 2	8 × 8 × 64	3 × 3 3 × 3	1 1
7	Pooling	Average pooling		4 × 4 × 64	2 × 2	2
8	Pooling	Global ave-pooling		1 × 1 × 64	4 × 4	1
9	Fully-connected	Fully-connected		2	- - - - -	- - -

2.2.3. Classification network

As for classification network, there are a lot of choices. Here we will mainly try the following three structures.

(a) ResNet.

The network structure here is built on ResNet-18 [24]. Different modifications are made to this structure depending on the different fusion strategies. When using depth fusion, we set the channel of filters as 384 and the number of filters as 16 in the first convolutional layer. When using length fusion, we set the channel of filters as 32 and the number of filters as 16 in the first convolutional layer. Furthermore, no matter which kind of fusion, the output channels of residual blocks are set to be 16, 32, 64 respectively. Table 1 summarizes the details of the network, where the input takes the output of depth fusion as an example.

(b) DenseNet.

The network structure here is built on DenseNet-201 [25]. It contains one convolutional layer at the beginning of the network, four dense blocks, three transition layers, one global average pooling layer, and one full-connected layer. To be more specific, the output of each layer in dense block is

$$t_l = H_l[(t_0, t_1, ..., t_{l-1})] \quad (2)$$

where t_l represents the output of layer l and $t_0 \sim t_{l-1}$ are the outputs of all previous layers. $H_l()$ is mainly composed of two convolutional layers, Conv(1 × 1) followed by Conv(3 × 3). Additionally, there are a batch normalization and relu before each Conv.

Between each two dense blocks, there is also a transition layer to reduce the parameters. This layer let input pass through a batch normalization, relu, Conv(1 × 1) and an average pooling layer successively. At the same time, dropout is used to reduce the overfitting.

In order to make the model more suitable for our classification task, we modify the classic DenseNet-201 in several ways. Table 2 summarizes the details of this network, where the input takes the output of depth fusion as an example. We remove the first max-pooling layer. Then, the stride in the first convolutional layer is changed from 2 to 1 and the number of output nodes of the fully connected layer is set to be 2. The growth rate and dropout rate in this structure are set to be 24 and 0.2 respectively.

(c) Shallow neural network.

Besides using classical networks, we also design a shallow neural network that only consists of two convolutional layers, two max-pooling layers and three fully connected layers. The numbers of nodes in these three fully connected layers are set to be 1024, 512 and 2 respectively. Table 3 summarizes the details of this net-

Table 2
Details in classification network based on DenseNet

Layers	Type	Inner structure	Output shape	Kernel size	Stride
0	Input	- - - - - - - -	32 × 32 × 384	- - - - -	- - -
1	Convolution	Convolutional layer	32 × 32 × 48	7 × 7	1
2	Dense Block	Convolutional layer Convolutional layer × 6	32 × 32 × 192	1 × 1 3 × 3	1 1
3	Transition Layer	Convolutional layer Average pooling	16 × 16 × 24	1 × 1 2 × 2	1 2
4	Dense Block	Convolutional layer Convolutional layer × 12	16 × 16 × 312	1 × 1 3 × 3	1 1
5	Transition Layer	Convolutional layer Average pooling	8 × 8 × 24	1 × 1 2 × 2	1 2
6	Dense Block	Convolutional layer Convolutional layer × 48	8 × 8 × 1176	1 × 1 3 × 3	1 1
7	Transition Layer	Convolutional layer Average pooling	4 × 4 × 24	1 × 1 2 × 2	1 2
8	Dense Block	Convolutional layer Convolutional layer × 32	4 × 4 × 792	1 × 1 3 × 3	1 1
9	Pooling	Global ave-pooling	1 × 1 × 792	4 × 4	1
10	Fully-connected	Fully-connected	2	- - - - -	- - -

Table 3
Details in classification network based on Shallow neural network

Layers	Type	Inner structure	Output shape	Kernel size	Stride
0	Input	- - - - - - - -	32 × 32 × 384	- - - - -	- - -
1	Convolution	Convolutional layer	32 × 32 × 128	3 × 3	1
2	Pooling	Max-pooling	16 × 16 × 128	2 × 2	2
3	Convolution	Convolutional layer	16 × 16 × 64	3 × 3	1
4	Pooling	Max-pooling	8 × 8 × 64	2 × 2	2
5	Fully-connected	Fully-connected	1024	- - - - -	- - -
6	Fully-connected	Fully-connected	512	- - - - -	- - -
7	Fully-connected	Fully-connected	2	- - - - -	- - -

work, where the input takes the output of depth fusion as an example.

2.3. Training

A standard backpropagation in a batch size of 16 is executed in this work. We use the stochastic gradient descent (SGD) optimizer with the learning rate of 0.05 that will gradually decrease with the increase of the training epoch. The type of GPU we used is GeForce GTX 1080Ti. The whole model is implemented on Tensorflow. More details of the implementation can be seen in the website of this project[2].

3. Dataset and leads extraction

3.1. ECG data

Despite in our best efforts, all public datasets that we can retrieve are time-series data, which are inconsistent with the data required in this work. Zhejiang Second People's Hospital of China provided us 957 12-lead ECG images from distinct patients, among which there are 483 MI images and 474 non-MI images. It means that we have 11484 leads in our dataset. Each image does not contain personal information about the patient. The images have been annotated by several cardiologists and rechecked by an expert. The ordering of 12 leads in one image is 3 × 4. The resolutions of these images are diverse. Some 12-lead ECG images from this dataset are shown in Fig. 4.

3.2. Extraction of leads

In this section, we are going to focus on how to conduct leads extraction automatically. When directly using the object detection network to locate 12 leads, it often results in incomplete or wrong segmentations. However, it is easier to identify the text above each lead in ECG images. Therefore, a new approach based on text detection is given in this section to get the bounding box of each lead as accurately as possible. This approach consists of two steps. The first step is to detect the text above each lead. In the second step the positions of those texts are used to locate the position of each lead.

3.2.1. Text detection

We use the Yolo3 model [26] to detect texts in ECG images. Yolo3 is trained by a manually collected dataset with annotations of 12 kinds of texts including I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6. Then ECG images are input to the trained Yolo3, bounding boxes of texts above each lead will be output. Fig. 5 gives an ECG image with 12 bounding boxes of texts. The 12 bounding boxes obtained from an ECG image can be described as $\{B_I, B_{II}, B_{III}, B_{aVR}, B_{aVL}, B_{aVF}, B_{V1}, B_{V2}, B_{V3}, B_{V4}, B_{V5}, B_{V6}\}$. Each $B_i = \{x_i^1, y_i^1, x_i^2, y_i^2\}$, where i $\in$ {*I, II, III, aVR, aVL, aVF, V*1, *V*2, *V*3, *V*4, *V*5, *V*6}. $\{x_i^1, y_i^1\}$ and $\{x_i^2, y_i^2\}$ represent the coordinates of the upper-left and lower-right corners of the bounding box.

3.2.2. 12 leads detection

In this step, we focus on how to use these bounding boxes to locate the positions of 12 leads.

The core idea is to obtain the position of a lead by using the two bounding boxes around this lead. For instance, the position of lead I can be got by bounding boxes of "I" and "aVL", the specific formula is as follows.

$$X_I^1 = x_I^1, Y_I^1 = y_I^2, X_I^2 = x_{aVL}^1, Y_I^2 = y_{aVL}^1 \tag{3}$$

where $\{X_I^1, Y_I^1\}$ and $\{X_I^2, Y_I^2\}$ are the upper-left and lower-right corners of the bounding box of lead I.

Similarly, the position of lead II can be got according to the bounding boxes of "II" and "aVF". By this way, leads I, II, aVR, aVL, V1, and V2 can be easily detected. Then we use the widths and heights of leads I, aVR, and V1 as references to get the positions of leads III, aVF, and V3. Leads V4, V5, and V6 can be detected by referencing the widths and heights of leads V1, V2, and V3. Now taking lead III as an example, the specific formula is as follows.

$$X_{III}^1 = x_{III}^1, Y_{III}^1 = y_{III}^2,$$

$$X_{III}^2 = X_{III}^1 + X_I^2 - X_I^1,$$

[2] *https://github.com/GXgaoxiang/ECG_MI_classification*

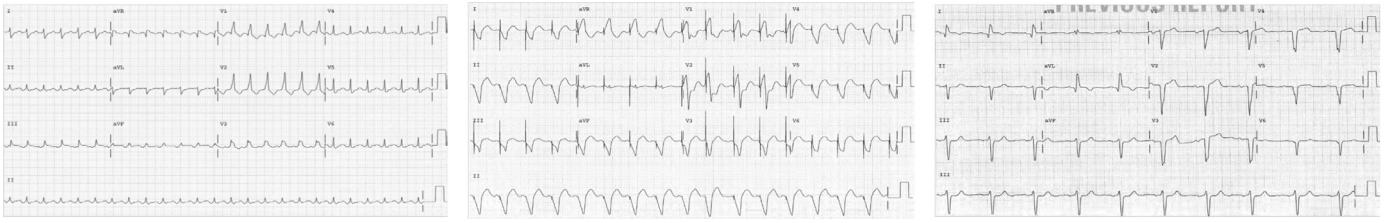

Fig. 4. Some 12-lead ECG images in the dataset.

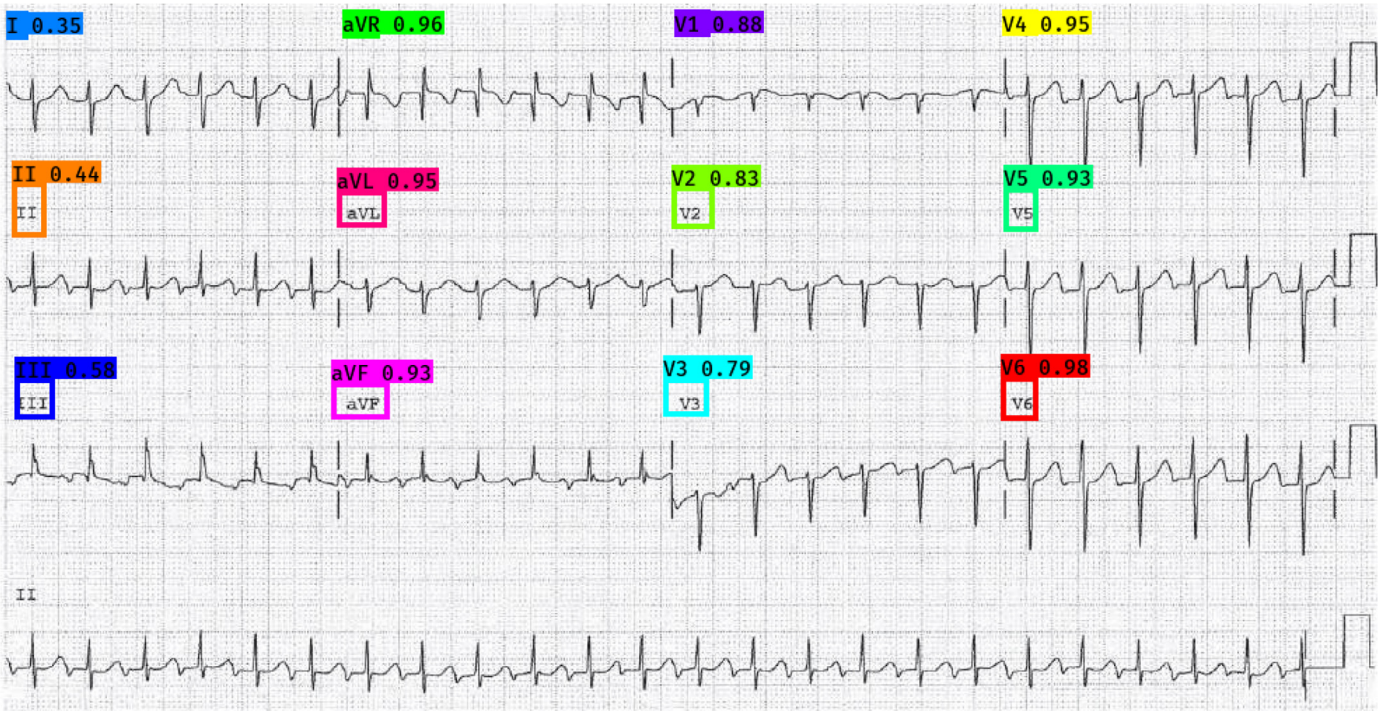

Fig. 5. Text detection in the ECG image.

$$Y_{III}^2 = Y_{III}^1 + Y_I^2 - Y_I^1. \tag{4}$$

After this step, each lead will get one bounding box, as shown in Fig. 6. These leads and their textual annotations are exactly what we will use for the multi-branch fusion network.

4. Experiments

4.1. Performance evaluation

We use accuracy (ACC), sensitivity (SEN), specificity (SPE) and F1-score to evaluate the proposed MI screening framework. Those four criteria are calculated based on the values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) as following.

$$ACC = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{5}$$

$$SEN = \frac{TP}{(TP + FN)} \tag{6}$$

$$SPE = \frac{TN}{(TN + FP)} \tag{7}$$

$$PRE = \frac{TP}{(TP + FP)} \tag{8}$$

$$F1 - score = \frac{2 \times (SEN \times PRE)}{(SEN + PRE)} \tag{9}$$

Due to the limitation of data size, we use k-fold cross validation to ensure the reliability of experiments. We randomly divide the data into seven parts, among which six parts are used to train the model, and the remaining part is used to test. The same procedure is repeated seven times. Then, ACC, SEN, SPE and F1-score of these procedures are averaged to be the final results for measuring the performance of each method.

4.2. Selection of networks' structures

The main purpose of this experiment is to find the most suitable network structure for the proposed myocardial infarction screening framework. The framework is composed of a multi-branch network, feature fusion and a classification network. Several candidate structures are verified for each part. The optimal model among these combinations will be chosen in response to their performances.

We combine the various architectures given in section 2. It should be noted that when using the deep neural network of ResNet or DenseNet to extract features in the multi-branch network, not only the training time is increased, but also the performance is not satisfied, comparing with shallow neural network. Therefore, in order to simplify the tests in the combinations, we

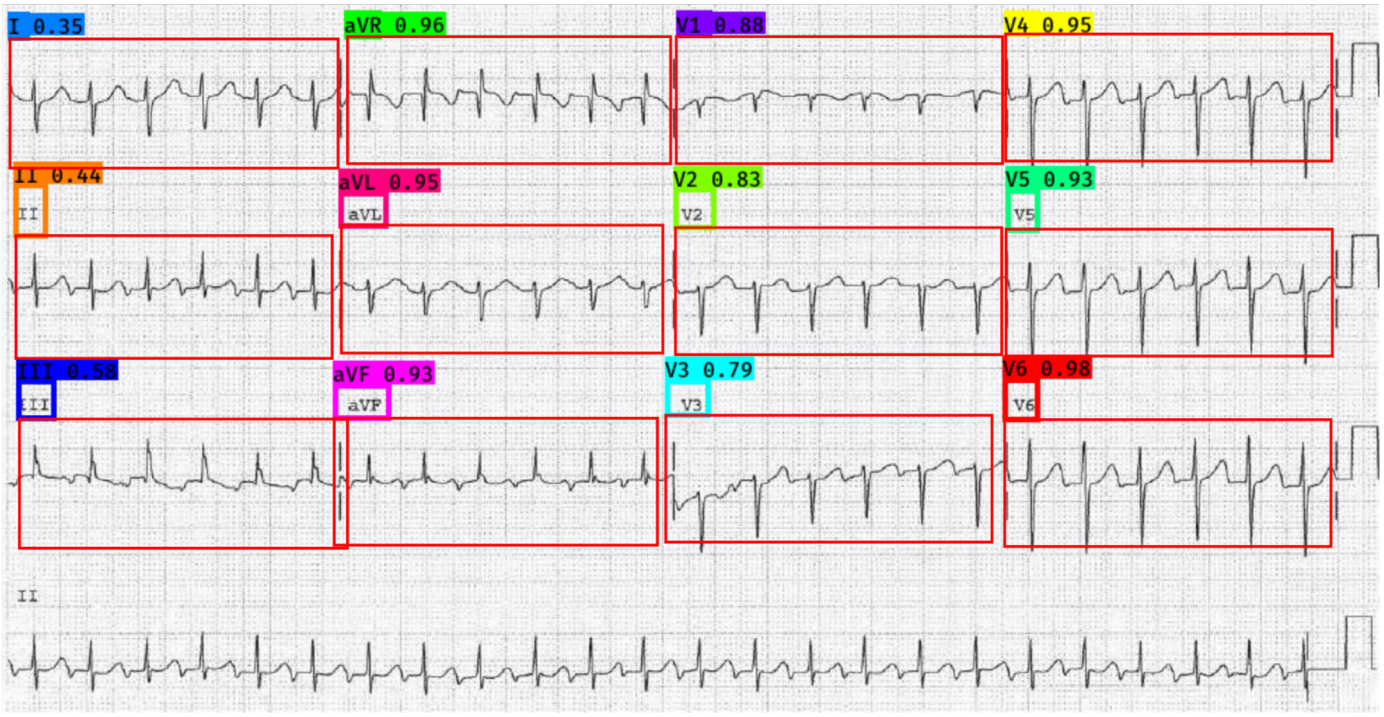

Fig. 6. The ECG image with 12 bounding boxes of 12 leads.

Table 4
Performances of different combinations in the proposed framework.

Multi-branch network	Feature fusion	Classification network	Performance			
			Acc(%)	*Sen*(%)	*Spe*(%)	*F*1 − *score*(%)
Residual block	Length fusion	Shallow neural network	87.50	87.07	89.01	86.52
	Depth fusion		89.06	85.99	93.75	90.11
Dense block	Length fusion	Shallow neural network	89.58	91.53	92.12	89.03
	Depth fusion		91.18	95.00	90.59	90.71
Shallow neural network	Length fusion	Shallow neural network	92.97	93.92	91.96	93.19
		ResNet	93.75	91.41	**96.43**	93.87
		DenseNet	92.19	93.92	91.04	91.86
	Depth fusion	Shallow neural network	90.63	89.50	92.49	90.44
		ResNet	91.67	88.08	95.14	91.75
		DenseNet	**94.73**	**96.41**	95.94	**93.79**

give up some combinations those use DenseNet or ResNet as the multi-branch network. Table 4 shows the performances of the possible combinations of these three parts in the proposed MI screening framework.

As can be seen from Table 4, the optimal combination takes the shallow neural network as the multi-branch network, DenseNet as the classification network, and depth fusion. It achieves 94.73% in accuracy, 96.41% in sensitivity, 95.94% in specificity, 93.79% in f1-score. In terms of accuracy, this combination is approximately 0.98% to 7.23% higher than others. The sensitivity of this combination is 1.41% to 10.42% higher than others. The F1-score of this combination is 0.08% to 7.27% higher than others. As for specificity, although this combination did not achieve the best results, it was only 0.49% lower than the best one.

4.3. Structural parameters

In order to obtain the optimal network, we carry out the second experiment on various structural parameters in the proposed approach.

(1) **Kernel size**

As mentioned in the previous section, the shallow neural network is chosen as the multi-branch network that has two convolutional layers. We test the kernel size of 3 × 3 and the kernel size of 5 × 5. Their performances are recorded in Table 5.

Table 5
The comparisons of model's performances on different kernel sizes

Kernel size	Performance			
	Acc(%)	*Sen*(%)	*Spe*(%)	*F*1 − *score*(%)
3 × 3	**94.73**	**96.41**	**95.94**	**93.79**
5 × 5	92.71	93.15	92.59	92.22

As can be seen from Table 5, when the size of kernel is set as 3 × 3, it is much better than the case of 5 × 5. For all the four criteria, using kernels with 3 × 3 makes the performance of our module be 1.57% to 3.35% higher than using 5 × 5.

(2)**Learning rate**

It is well known that the learning rate is one of the most important hyper-parameters that will affect the performance of the model. It controls the rate of gradient descent. In order to find the most suitable learning rate, we set up five groups of comparative experiments by setting the initial learning rates as 0.03, 0.05, 0.07, 0.1 and 0.2 respectively.

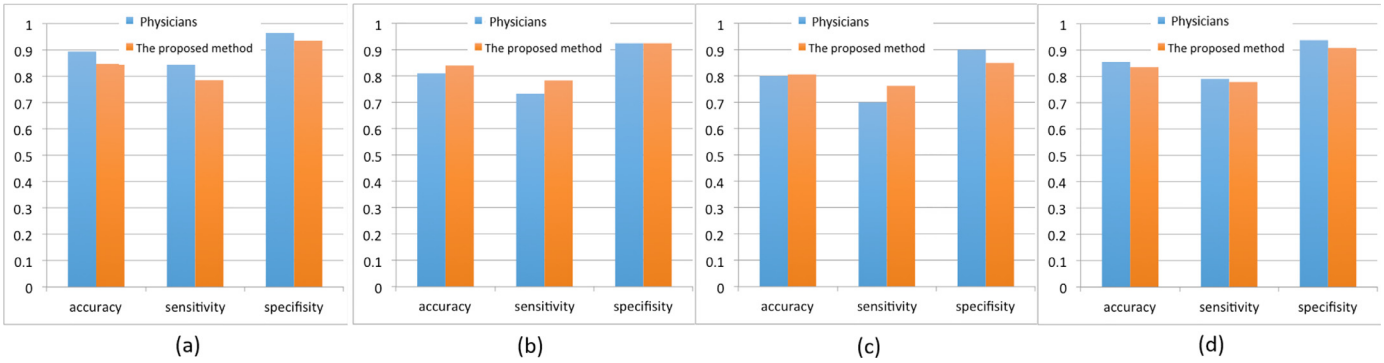

Fig. 7. Practical test between the proposed method and physicians.(a) Comparisons on ECG images generated from signal data. (b) Comparisons on ECG images generated by taking pictures of printed ECGs. (c) Comparisons on ECG images generated by taking pictures of electrocardiograph's screen. (d) Comparisons on the whole practical test dataset.

Table 6
The comparisons of model's performances on different learning rates

Learning rate	Performance			
	Acc(%)	*Sen*(%)	*Spe*(%)	$F1-score(\%)$
0.03	93.75	93.37	**96.97**	93.23
0.05	**94.73**	**96.41**	95.94	**93.79**
0.07	92.97	93.75	92.59	92.52
0.10	92.19	95.49	91.47	90.61
0.20	89.06	90.40	90.41	85.81

Table 7
The comparisons of model's performances on different growth rates

Growth rate	Performance			
	Acc(%)	*Sen*(%)	*Spe*(%)	$F1-score(\%)$
12	91.41	93.19	91.78	91.64
24	**94.73**	**96.41**	**95.94**	93.79
32	94.53	94.32	95.44	**94.53**

As can be seen from Table 6, when the initial learning rate is set at 0.05, the model performs best. In addition, we can see that when the learning rates are set to be 0.03, 0.05, 0.07 and 0.1, model's performances are relatively close. However, when the learning rate is set to be 0.2, the performance declines dramatically. It verifies that too large learning rate will make the model difficult to converge to the optimal value.

(3)**Growth rate**

From Table 4, we select DenseNet as the classification network in the proposed structure. This network has a hype-parameter named growth rate which controls the number of feature maps output from each layer in Dense block. Therefore, we carry out an experiment to test the impact of this parameter on the performance of the model.

As can be seen from Table 7, when the growth rate is set to 24 or 36, the accuracy and specificity of the model are almost the same. The model has higher sensitivity when growth rate is 24, while the model has higher F1-score when the growth rate is 36. When the growth rate is set to be 12, the performance of the model is worst. We choose 24 as the growth rate in our model by considering of the shorter training time.

4.4. Comparison with different classifiers

In the proposed framework, DenseNet is chosen as the classification network which uses the fully connected layer to do the classification task. Therefore, we test whether using different classifiers can optimize the performance of the model or not.

Table 8
Comparisons with different classifiers

Classifier	Accuracy %
The proposed network	**94.73**
KNN classifier	89.84
SVM classifier	92.19

At first, under the growth rate of 24, the output of the last Dense block in the classification network is extracted as the feature vector with the dimension of 792. Then, the feature vectors of all the test ECG images are fed into KNN classifier or SVM classifier for classification. The results by using these two classifiers are presented in Table 8.

(1)**KNN classifier**

KNN uses the distances between samples for classification. The strategy is that the label of a sample depends on labels of the k existing samples closest to it. In this experiment, k is set to 3 and the distances between the samples are calculated by the Euclidean distance.

(2)**SVM classifier**

SVM is a binary classification model, which constructs a maximum-margin hyperplane in high dimensional space to separate positive and negative samples. Here, linear kernel SVM is used and the penalty factor is set to be 1.

As can be seen from Table 8, the accuracies of KNN and SVM are 89.84% and 92.19% respectively, which are 4.89% and 2.54% lower than the proposed network respectively.

4.5. Comparison with different approaches

We test the performances of some other approaches, such as ResNet-34 [24] and Densenet [25]. We also implemented the network proposed by Jun et al. [23]. It should be noted that these three methods all take complete ECG images as input instead of 12 crops of ECG images. Since there are not enough ECG images, data augmentation is used to expand the data for these three methods. Three strategies including moving down 30 pixels, increasing the brightness, and adding salt-and-pepper noise, are used to increase the data. Finally, there are 3828 ECG images for these three methods. Since both of the length and width of ECG images in the dataset are over 1000, but the input of the network proposed in [23] was set as 128, which will loss a large amount of information. Therefore, in order to make a better comparison, we made a slight change to the network of [23] by changing the input to be 448 and adding one convolution layer and one pooling layer. Same with the

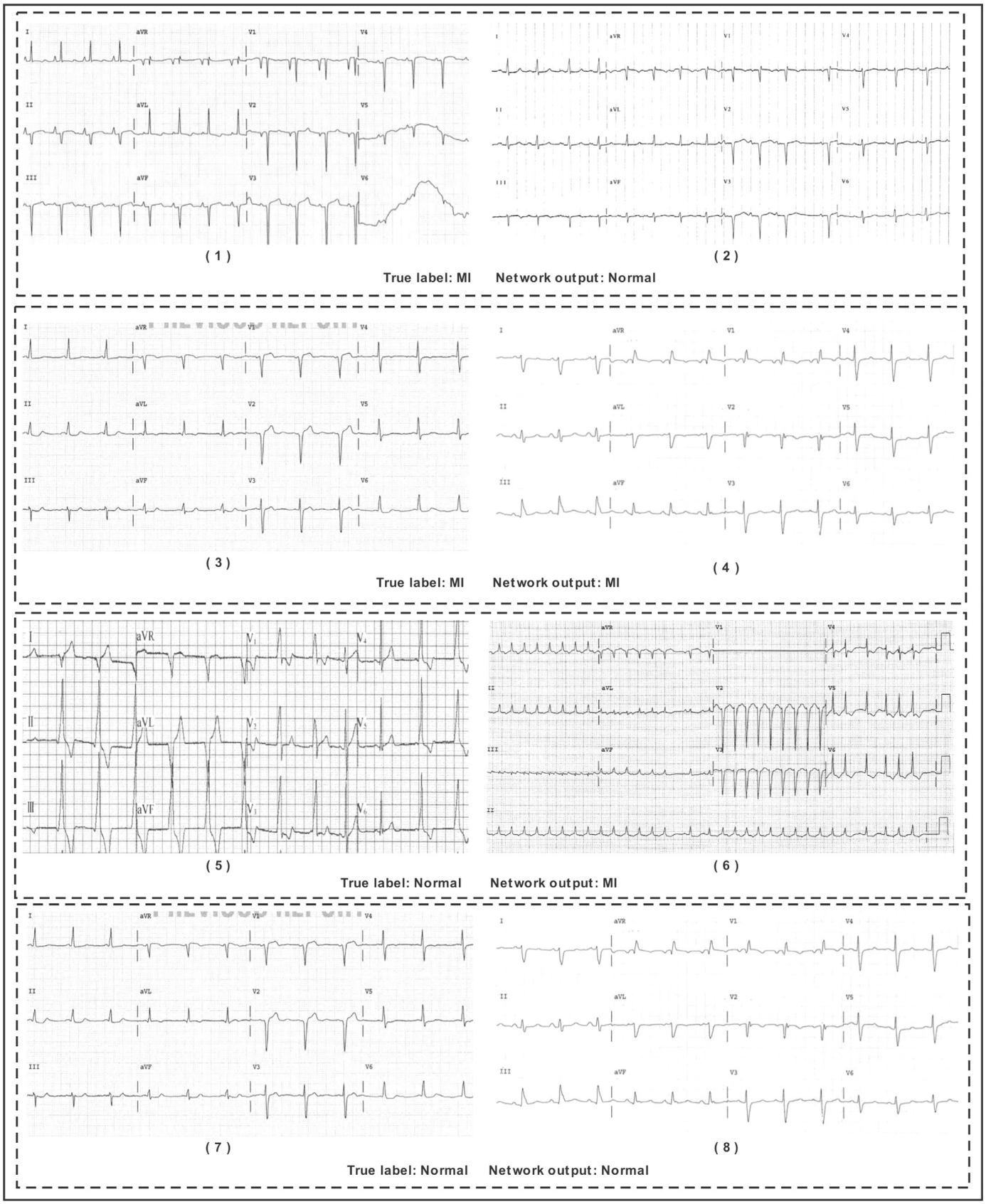

Fig. 8. Some results from the proposed network and their true labels.

Table 9
Comparisons with different approaches

Network	Performance			
	Acc(%)	*Sen*(%)	*Spe*(%)	$F1-score$(%)
ResNet-34 [24]	90.05	91.93	89.88	89.81
DenseNet [25]	86.03	88.07	86.44	82.80
Method of [23]	87.63	90.83	86.06	87.06
The proposed method	**94.73**	**96.41**	**95.94**	**93.79**

evaluation of the proposed method, we also adopt 7-fold cross validation for all these three methods. Table 9 shows the results.

As can be seen from Table 9, the proposed approach by dividing the whole ECG image into 12 leads and feeding them into the network performs much better than other methods using the whole ECG images. The ResNet-34 performs better than DenseNet, but compared with the proposed approach, the accuracy, sensitivity, specificity and F1-score is 4.68%, 4.48%, 6.06%, and 3.98% lower. Compared with the method in [23], the accuracy, sensitivity, specificity and F1-score obtained by the proposed method is 7.10%, 5.58%, 9.88%, and 6.73% higher than [23]. The main reason may be that the proposed method separately extracts features from each lead of ECG image which can obtain much more details from the imperceptible changes.

4.6. Practical test

For testing how the proposed method performs in the real cases. Zhejiang Second People's Hospital of China provides another test dataset of 600 ECG images including 340 MI ECGs and 260 normal ECGs. In this dataset, there are three kinds of ECG images by concerning how they came from. Among them, there are 440 digital ECG images generated from signal data, 100 ECG images generated by taking pictures of printed ECGs, and 160 ECG images collected by taking pictures of electrocardiograph's screen. Then the tests are carried out by the following way. First, 5 voluntary physicians give the assessments on these 600 ECG images separately, then one expert will judge these assessments correct or not. After that, accuracy, sensitive and specificity are calculated for each physician's results and the average ones are thought as physicians' level. At the same time, all the 600 ECG images are tested by the proposed model, then model's outputs are judged by the same expert, and accuracy, sensitive and specificity are calculated. The comparisons are shown in Fig. 7.

As shown in Fig. 7, when testing on the ECG images generated from signal data, it means that physicians will read these digital ECG images by using computer. For this type of ECG images, the accuracy achieved by physicians is 0.47% higher than the proposed method. While, testing on the ECG images generated by taking pictures of printed ECGs shows that the proposed method achieves a little higher sensitivity and accuracy than physicians, which is 0.3% and 0.5% higher respectively. For the ECG images collected by taking pictures of electrocardiograph's screen, the proposed method has a similar performance with physicians. From Fig. 7, it can be seen that the proposed method performs better on the printed ECG images, which is a good point for the future application. From the results on the whole dataset, we can find that the proposed method almost reaches the level of physicians with much faster speed.

5. Discussion

Most of the researches mentioned in the introduction are based on time series data. We can also see most of the researches not only require the detection of R-peaks or ST segments, but also need pre-processing to remove the noise like baseline wander. However, the proposed method can skip these steps, which is a complete end-to-end structure.

Very rare researches did researches on ECG images. The method in [23] transformed the ECG time series data to be clean ECG images, then arrhythmia were detected by doing classification on these ECG images. This method obtained high accuracy on these clean ECG images, but from Table 9, it can be seen that it did not work well on the non-clean ECG images like scanned ECG images and screenshots of ECGs. In contract, since the proposed method can fully extract the detailed features from 12 leads, it achieves high accuracy and also specificity.

Fig. 8 shows some classification results of the proposed MI screening framework. In Fig. 8, the ground truth of image (1) and image (2) are MI, but they were predicted to be normal. Compared with image (3) and image (4), the lead V5 and lead V6 show large baseline wander, and the grid lines in image (2) are not very clear and partially missed. What's more, we can see that image (5) and image (6) were also wrongly predicted by the network. The reason may be that the position of texts (I, II, III, etc.) are not suitable for our automatic lead extraction method, which leads to incomplete annotations.

The proposed method needs to detect the text above each lead. Then based on the positions of texts, the 12 leads can be automatically segmented. It is easy and performs very well, but for some ECG images with missing texts or unclear texts, it cannot work well. Therefore, in the future work, how to make leads extraction more accurate and flexible needs to be explored. In addition, some heart diseases show special characteristics in only a few specific leads. So, we also plan to carry out researches to extend the proposed approach for more kinds of heart diseases.

6. Conclusion

In this paper, we proposed a model to screen the myocardial infarction from 12-lead ECG images. The proposed approach consists of multi-branch network, feature fusion and classification network. Based on extensive experiments, shallow CNN was chosen as the multi-branch network to extract the features of each lead, feature maps were fused by depth fusion, and finally features were fed into a classification network which was built based on DenseNet. The proposed model achieved very high sensitivity and specificity on MI screening.

Declaration of Competing Interest

The authors declare no conflict of interest that could affect their work.

Acknowledgments

This work is supported by Zhejiang Provincial Natural Science Foundation of China under grants no. LY18F020034 and National Natural Science Foundation of China under grants no. 61702275, 61801428 and 61976192. We also give our sincere thanks to Zhejiang Second People's Hospital of China for providing data and helping us with the practical test.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cmpb.2019.105286

References

[1] L. Xu, X. Huang, J. Huang, Y. Fan, H. Li, J. Qiu, H. Zhang, W. Huang, Value of three-dimensional strain parameters for predicting left ventricular remodeling after st-elevation myocardial infarction, Int. J. Cardiovasc.Imag. 33 (5) (2017) 663–673.

[2] C. Bradley, A. Bowery, R. Britten, ea V. Budelmann, Opencmiss: a multi-physics & multi-scale computational infrastructure for the vph/physiome project, Progr. Biophys. Mol. Biol. 107 (1) (2011) 32–47.
[3] C. Xu, L. Xu, Z. Gao, S. Zhao, H. Zhang, Y. Zhang, X. Du, S. Zhao, D. Ghista, H. Liu, S. Li, Direct delineation of myocardial infarction without contrast agents using a joint motion feature learning architecture, Med. Image Anal. 50 (2018) 82–94.
[4] L. Wang, H. Zhang, K. Wong, H. Liu, P. Shi, Physiological-model-constrained noninvasive reconstruction of volumetric myocardial transmembrane potentials, IEEE Trans. Biomed. Eng. 57 (2) (2010) 296–315.
[5] K. Thygesen, J.S. Alpert, A.S. Jaffe, B.R. Chaitman, J.J. Bax, D.A. Morrow, H.D. White, Fourth universal definition of myocardial infarction, J. Am. College Cardiol. 72 (18) (2018) 2231–2264.
[6] S. Osowski, T. Hoai, T. Markiewicz, Support vector machine-based expert system for reliable heartbeat recognition, IEEE Trans. Biomed. Eng. 51 (4) (2004) 582–589.
[7] J. Fayn, A classification tree approach for cardiac ischemia detection using spatiotemporal information from three standard ecg leads, IEEE Trans. Biomed. Eng. 44 (9) (2011) 95–102.
[8] D.A. Coast, R.M. Stern, G.G. Cano, S.A. Briller, An approach to cardiac arrhythmia analysis using hidden markov models, IEEE Trans. Biomed. Eng. 37 (9) (1990) 826–836.
[9] E.D. Ubeyli, Combining recurrent neural networks with eigenvector methods for classification of ecg beats, Digital Signal Process. 19 (2) (2009) 320–329.
[10] R.R. Sharma, M. Kumar, R.B. Pachori, Automated cad identification system using time frequency representation based on eigenvalue decomposition of ecg signals, in: Machine Intelligence and Signal Analysis, 2019, pp. 597–608.
[11] K. Padmavathi, K. Sri Rama Krishna, Myocardial infarction detection using magnitude squared coherence and support vector machine, in: International Conference on Medical Imaging, M-Health and Emerging Communication Systems (MedCom), IEEE, 2014.
[12] L. Sun, Y. Lu, K. Yang, S. Li, Ecg analysis using multiple instance learning for myocardial infarction detection, IEEE Trans. Biomed. Eng. 59 (12) (2012) 3348–3356.
[13] P. Hao, K. You, H. Feng, X. Xu, F. Zhang, F. Wu, P. Zhang, W. Chen, Lung adenocarcinoma diagnosis in one stage, Neurocomputing (2019).
[14] Y. Xia, H. Zhang, L. Xu, Z. Gao, H. Zhang, H. Liu, S. Li, An automatic cardiac arrhythmia classification system with wearable electrocardiogram, IEEE Access 6 (2018) 16529–16538.
[15] U.R. Acharya, H. Fujita, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals, Inf. Sci. 415-416 (2017) 190–198.
[16] J. Huang, B. Chen, B. Yao, W. He, Ecg arrhythmia classification using stft-based spectrogram and convolutional neural network, IEEE Access 7 (2019) 92871–92880.
[17] B. Pourbabaee, M.J. Roshtkhari, K. Khorassani, Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients, IEEE Trans. Syst. Man Cybernet. 48 (12) (2017) 2095–2104.
[18] W. Liu, M. Zhang, Y. Zhang, Y. Liao, Q. Huang, S. Chang, H. Wang, J. He, Real–time multilead convolutional neural network for myocardial infarction detection, IEEE J. Biomed. Health Inf. 22 (5) (2017) 1434–1444.
[19] N. Strodthoff, C. Strodthoff, Detecting and interpreting myocardial infarction using fully convolutional neural networks, Physiol. Measur. (2018).
[20] U.B. Baloglu, M. Talo, O. Yildirim, R.S. Tan, U.R. Acharya, Classification of myocardial infarction with multi-lead ecg signals and deep cnn, Pattern Recognit. Lett. 122 (2019) 23–30.
[21] W. Lu, H. Hou, J. Chu, Feature fusion for imbalanced ecg data analysis, Biomed. Signal Process. Control 41 (2018) 152–160.
[22] Y. Ji, S. Zhang, W. Xiao, Electrocardigram classification based on faster regions with convolutional neural network, Sensors 19 (2019).
[23] T.J. Jun, H.M. Nguyen, D. Kang, D. Kim, Y. Kim, Ecg arrhythmia classification using a 2-d convolutional neural network, arxiv.org/abs/1804.06812 (2018).
[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[25] G. Huang, Z. Liu, L.V.D. Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: CVPR, 2017, pp. 4700–4708.
[26] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, Tech report, University of Washington (2018).