

# Study on the use of convolutional neural networks for the diagnosis of atrial fibrillation

M. Maione<sup>1</sup>

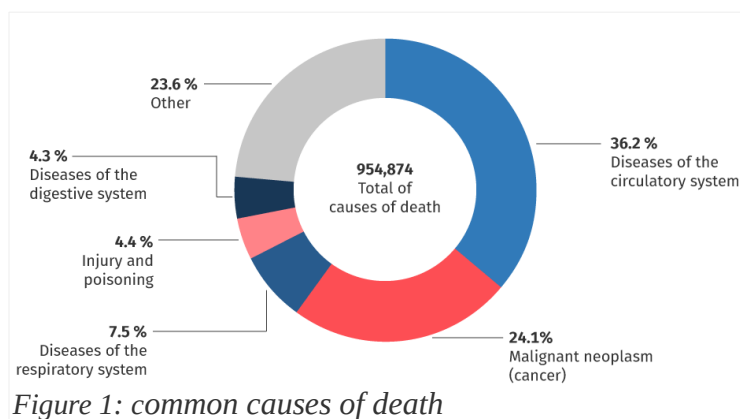
(November 2020)

**Abstract** – In this study we will introduce atrial fibrillation, one of the cardiac arrhythmias, and see how it can be diagnosed using convolutional neural networks in combination with various methods and supervised learning models, including: gray-level co-occurrence matrix, short-time Fourier transform based spectrogram, support-vector machines, k-nearest neighbors, multi-layer perceptron, focal loss, multi-scale decomposition, time-frequency analysis.

**Index terms** – Biomedical monitoring, electrocardiogram, arrhythmia, atrial fibrillation, deep learning, convolutional neural network, classification.

## 1. Introduction

Cardiovascular diseases are a group of diseases affecting the heart and/or blood vessels. On a world level, and in particular in countries with a typically Western lifestyle, cardiovascular diseases are the cause of 17 million deaths per year.



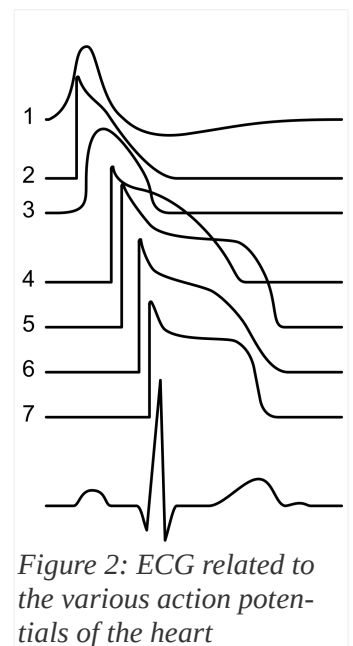
The narrowing, obstruction or excessive enlargement (aneurysm) of the blood vessels that can accompany this disease are in fact responsible for very widespread pathologies, such as coronary (angina pectoris and heart attack), cerebrovascular (stroke) and peripheral vascular diseases. In the family of cardiovascular pathologies, all congenital heart defects, rheumatic diseases involving myocardium, various forms of arrhythmia, pathologies affecting the heart valves and heart failure are also included. Cardiac pathologies are divided into: cardiovascular disease, coronary heart disease, heart "muscle" disease, heart valve disease, pericardial disease, heart conduction disease, vessel disease.

In this study we focus on heart conduction disease in particular on atrial fibrillation, one of the two arrhythmias<sup>[b]</sup>.

### 1.1. Electrocardiogram

The electrocardiogram (ECG) is the graphic reproduction of the electrical activity of the heart during its operation,

recorded at the level of the body surface. On the surface of the body, low intensity electric fields (1mV) are present and can be recorded, which in the individual at rest are mainly due to periodic depolarizations and repolarizations of the heart. To record an electrocardiogram it is necessary to have electrodes placed on the body surface, forming leads arranged in such a way as to be able to better analyze the variations of the heart's dipole vector. In order to record the potentials, 10 electrodes are placed on the body: 4 peripheral (wrists and ankles) and 6 precordial, so as to record 12 leads.



In general, the electrocardiogram (ECG) signals consist of six components that are designated as P, Q, R, S, T, and U:

- The P wave represents atrial depolarization [0.2 – 0.4mV];
- The QRS complex represents ventricular depolarization [1 – 2mV];
- The T wave represents ventricular repolarization [0.4 – 0.5mV];
- The U wave represents papillary muscle repolarization.

Usually an electrocardiograph has an amplification of a factor of 60db only in the useful band. For monitoring it uses frequencies between 0.05 – 50Hz, while for diagnostics purpose it goes up to 1kHz<sup>[13]</sup>.

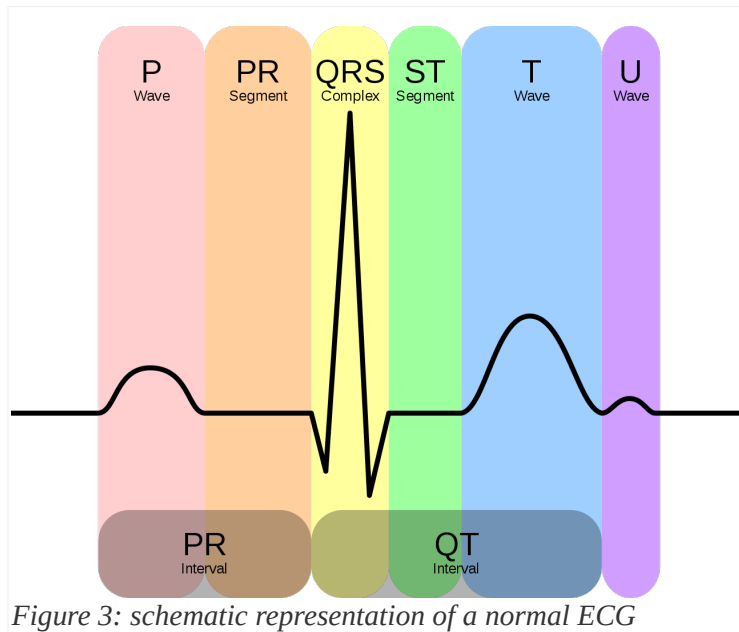


Figure 3: schematic representation of a normal ECG

## 1.2. Arrhythmia

Arrhythmia is a clinical condition in which the normal frequency or regularity of the heart rhythm is missing, or the physiological atrio-ventricular activation sequence is altered. Common symptoms are:

- Extrasystole: it is like a “void”, a missed beat;
- Tachycardia: the sensation is of an increase in beats, which can be regular but also irregular, fatigue, difficult breathing, dizziness;
- Bradycardia: fatigue, dizziness and possible loss of consciousness.

The alterations can have the following origins: at the level of the sinoatrial node, of supraventricular origin, at the level of the atrioventricular node, of ventricular origin. As for the supraventricular and ventricular origins, we have two types of fibrillation: atrial and ventricular<sup>[c]</sup>.

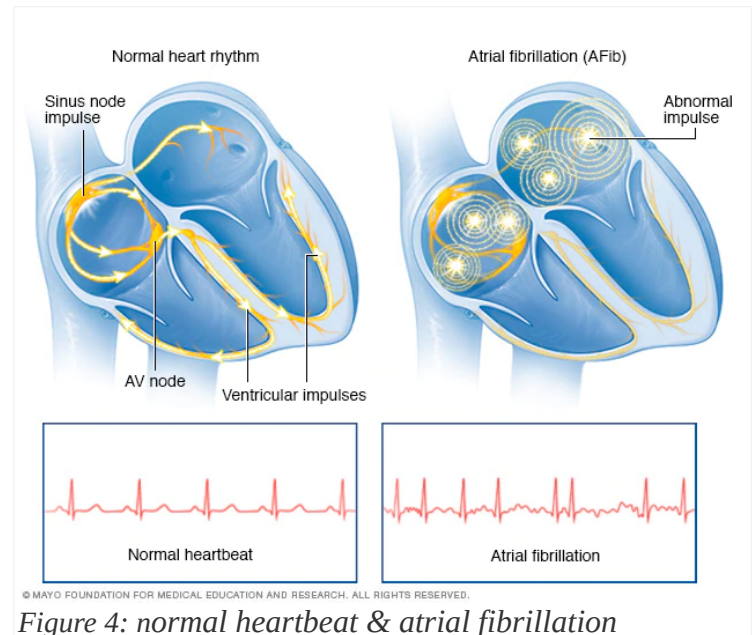
### a) Ventricular fibrillation

Ventricular fibrillation (VF) is a very rapid, chaotic cardiac arrhythmia that causes uncoordinated contractions of the heart muscle. With the onset of this arrhythmia, blood circulation slows down considerably, up to cardio-circulatory arrest and subsequent respiratory arrest, until death if cardioversion of the rhythm is not intervened through defibrillation and cardiopulmonary resuscitation<sup>[c]</sup>.

### b) Atrial fibrillation

Atrial fibrillation (AF) is a cardiac arrhythmia that originates in the atria. The electrical impulses that give rise to the contraction of the atria are activated in a totally chaotic and fragmentary way, giving rise to multiple wave fronts and disorganized and fragmentary contractions. The current clinical approach aims to treat symptoms by:

- Rhythm control (i.e. recovery and maintenance of sinus rhythm with anti-arrhythmic drugs or catheter ablation);
- Heart rate control with drugs that regulate the conduction of atrial stimuli to the ventricles associated with anti-thrombotic therapy.



© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

Figure 4: normal heartbeat & atrial fibrillation

From Figure 5 it can be observed that for AF patient there are tiny irregular fluctuations in the P-wave and QRS complex. Given the presence of these irregularities, for the purpose of arrhythmia screening various morphological features, including the peaks and widths corresponding to different ECG segments are typically used.

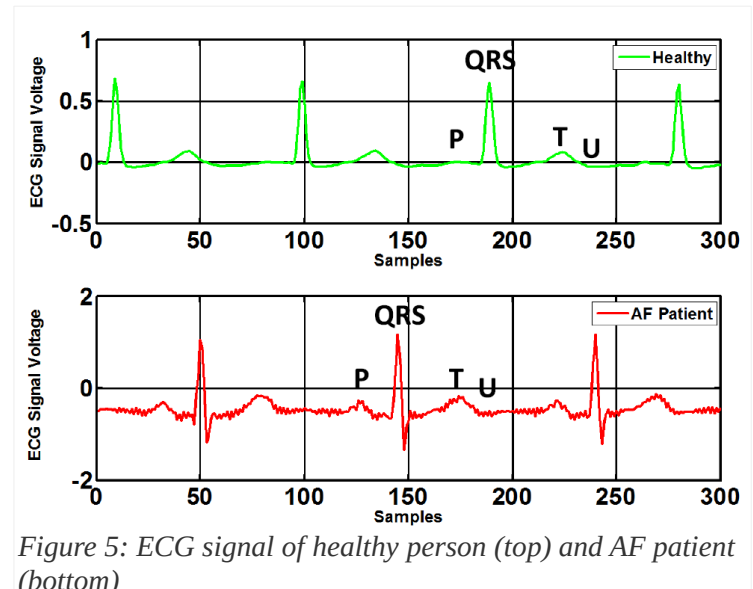


Figure 5: ECG signal of healthy person (top) and AF patient (bottom)

In Figure 6 are visible the F-waves with their saw-tooth appearance.

AF occurs in 2% of the population, and increases to 6% by the age of 65. The most serious complication of AF is thromboembolic stroke, which leads to permanent disability or even death. Since in many cases the symptoms are initially imperceptible to patients, diagnosing it as soon as possible becomes difficult, for this reason with the current technological progress, diagnostic tools can

be included in wearable devices, for example in smart watches or in affordable medical devices<sup>[4]</sup>.

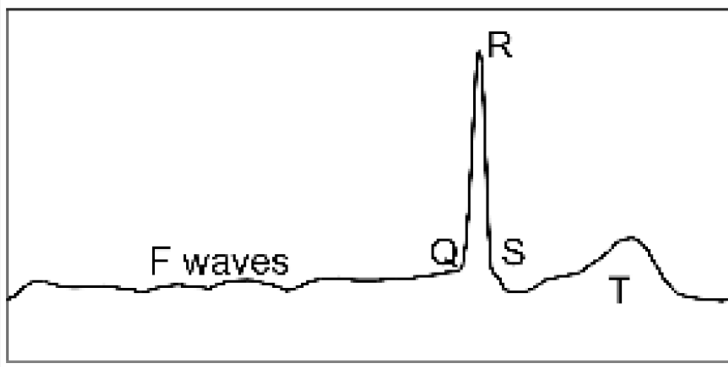


Figure 6: AF beat

## 2. Notation and relevant definitions

The introduction of the computational models used will now follow.

### 2.1. Artificial neural network

An artificial neural network (ANN) is an interconnected group of nodes, inspired by a simplification of neurons in a brain. As in Figure 7 each circular node represents an artificial neuron and a line represents a connection from the output of one artificial neuron to the input of another.

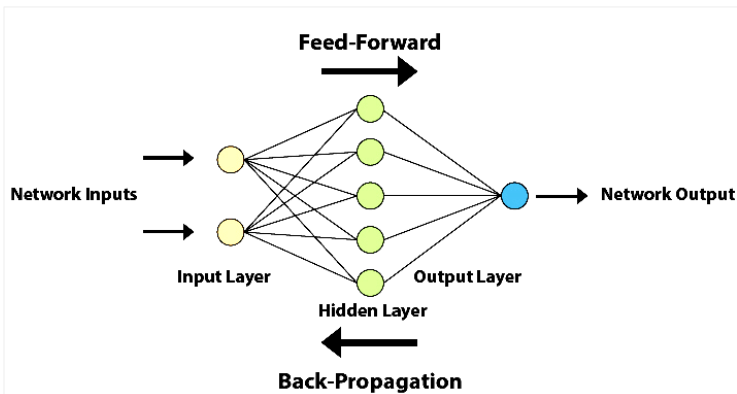


Figure 7: artificial neural network scheme

Typically neural networks are obtained through the combination of simple predictor of the form  $g(x) = \sigma(w^T x)$ . The function  $\sigma: \mathbb{R} \rightarrow \mathbb{R}$  is known as activation function<sup>[12]</sup>.

To define the neural computation we must specify: the neural model, the model dimensions, the configuration procedure<sup>[11]</sup>.

#### a) Neural model

Neural model is composed by activation function and network topology.

Common activation function are: binary step, linear activation, sigmoid, tanH, ReLU, softmax, swish, etc...

The neural network topology represents the way in which neurons are connected to form a network. In other words, the neural network topology can be seen as the relation-

ship between the neurons by means of their connections. The topology of a neural network plays a fundamental role in its functionality and performance. Some famous networks are: feed-forward NN, regulatory feedback networks, radial basis function network, recurrent neural network, modular neural network, etc...

#### b) Configuration procedure

Configuration procedure is composed by: configuration algorithm, training set, validation set.

Configuration algorithm must be chosen based on the learning paradigm: supervised, unsupervised and reinforcement.

In supervised learning the data set contain both the data points  $\mathcal{X}$  and the labels  $\mathcal{Y}$ . The learning task is to produce the desired output  $y$  for each input  $x$ . A cost function  $\ell$  is used to estimate the correctness of the predicted label  $\hat{y}$  compared to the desired output  $y$ .

The simplest division of the data-set is using the Pareto principle, that divide in 80/20 the training and the test set.



Figure 8: 80/20 rule

When evaluating different settings for estimators, there is still a risk of over-fitting on the test set because the parameters can be tweaked until the estimator performs optimally. This way, knowledge about the test set can “leak” into the model and evaluation metrics no longer report on generalization performance. To solve this problem, yet another part of the data-set can be held out as a so-called “validation set”: training proceeds on the training set, after which evaluation is done on the validation set, and when the experiment seems to be successful, final evaluation can be done on the test set.

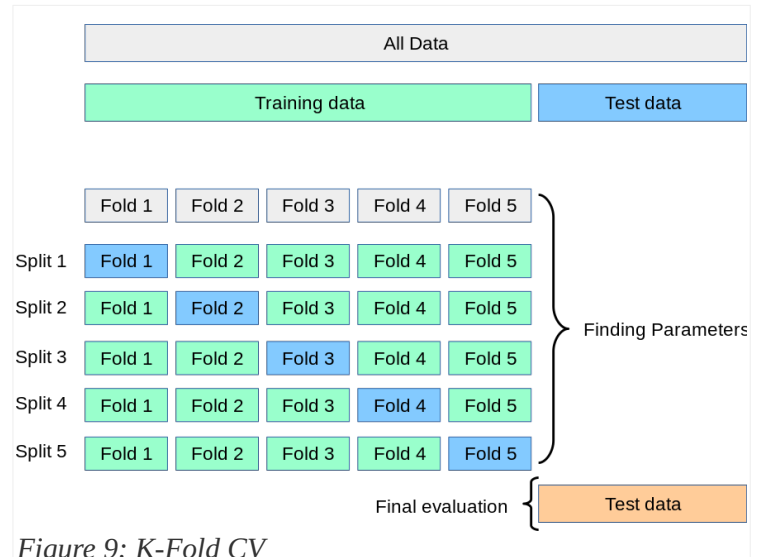


Figure 9: K-Fold CV

However, by partitioning the available data into three sets, we drastically reduce the number of samples which

can be used for learning the model, and the results can depend on a particular random choice for the pair of (train, validation) sets.

A solution to this problem is a procedure called cross-validation (CV). A test set should still be held out for final evaluation, but the validation set is no longer needed when doing CV. In the basic approach, called K-Fold CV (Figure 9), the training set is split into  $k$  smaller sets<sup>[1]</sup>.

Instead unsupervised learning looks for previously undetected patterns in a data set with no pre-existing labels.

Reinforcement learning minimize long-term cost modifying the network's weights. At each point in time an action is performed and an observation is received with a cost. At this point the algorithm/agent decides whether to perform new actions to uncover their cost or to exploit prior learning. In this case NN is used as the learning component.

## 2.2. Feed-forward neural network

In a feed-forward neural network connections between units do not form loops and information only moves in one direction, forward, with respect to entry nodes, through hidden nodes (if any) to exit nodes. Feed-forward NN computes a function  $f: \mathbb{R}^d \rightarrow \mathbb{R}^n$ . A parameter  $w_{ij} \in \mathbb{R}$  (called weight) is associated with every edge  $(i, j)$ . NNs are trained using algorithms that reduce the training error. Fixed a cost function  $\ell$ , an example  $(x_t, y_t)$ , defined  $\ell_t(W) = \ell(f_{G,W,\sigma}(x_t), y_t)$  and  $Z_t$  as the index of a random training example, then the standard training algorithm for NNs is stochastic gradient descent:

$$w_{i,j} \leftarrow w_{i,j} - \eta_t \frac{\partial \ell_{Z_t}(W)}{\partial w_{i,j}} : (i, j) \in E \quad (1)$$

This procedure is known as error back-propagation algorithm<sup>[12]</sup>.

## 2.3. Convolutional neural network

Convolutional neural networks (CNN) function like all feed-forward neural networks: an input layer, one or more hidden layers, which perform calculations using activation functions, and an output layer with the result. The difference is precisely the convolution in place of general matrix multiplication. The typical architecture of a CNN is formed by: convolutional layer, pooling layer, ReLU layer, fully connected layer and loss layer.

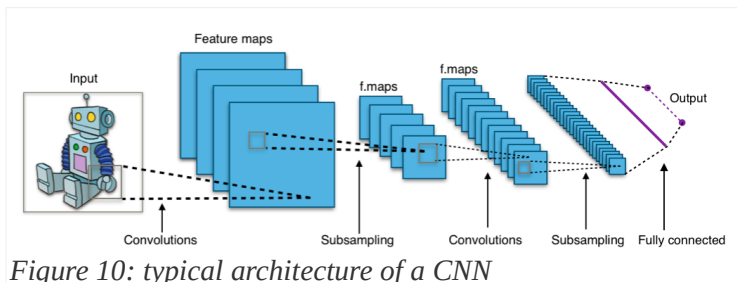


Figure 10: typical architecture of a CNN

## a) Convolution

Discrete convolution is an operation between two functions  $f$  and  $h$  that produce a third function  $G$  which consists in integrating the product between the first and the second translated by a certain value. For complex-valued function  $f$  and  $h$  defined on the set  $\mathbb{Z}$  the discrete convolution  $G$  of  $f$  and  $h$  is given by<sup>[11]</sup>:

$$G[m, n] = (f * h)[m, n] \quad (2)$$

$$(f * h)[m, n] \stackrel{\text{def}}{=} \sum_j \sum_k h[j, k] \cdot f[m-j, n-k] \quad (3)$$

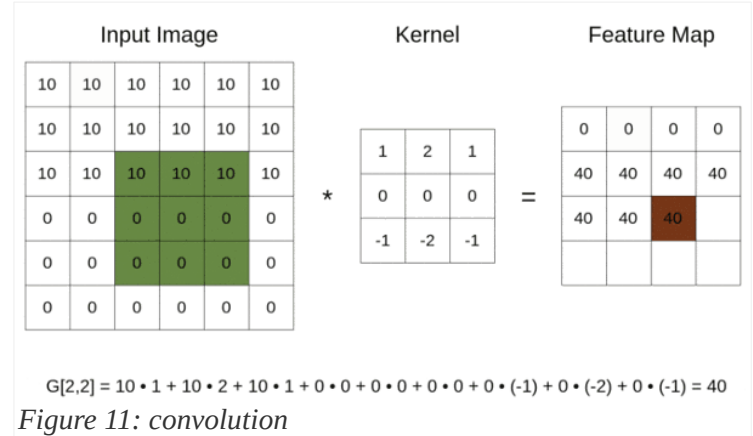


Figure 11: convolution

## 2.4. QRS recognition algorithms

QRS recognition algorithms are used to facilitate the annotation of the QRS complex in databases.

Various classes of QRS recognition algorithms have been proposed in the literature, which can be classified according to their complexity and performance. The Pan-Tompkins algorithm is one of the most used methods for real-time recognition of the QRS complex from the ECG signal<sup>[13]</sup>.

The performance of the method was tested on an annotated arrhythmia database and evaluated also in presence of noise. Pan and Tompkins reported that the 99.3% of QRS complexes was correctly detected<sup>[e]</sup>.

## 2.5. Assessment indicators

To evaluate the performance of the proposed classifiers, five statistical indicator are commonly used:

### a) Accuracy

It's the ratio of the correctly labeled subjects to the whole pool of subjects. Accuracy answers the following question: How many patients did we correctly label out of all the patients?

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

### b) Sensitivity

Sensitivity is the ratio of the correctly AF labeled by our program to all who have AF in reality.

Sensitivity answers the following question: of all the people who have *AF*, how many of those we correctly predict?

$$\frac{TP}{TP+FN} \quad (5)$$

### c) Specificity

Specificity is the correctly *healthy* labeled by the program to all who are *healthy* in reality. Specificity answers the following question: of all the people who are *healthy*, how many of those did we correctly predict?

$$\frac{TN}{TN+FP} \quad (6)$$

### d) Precision

Precision is the ratio of the correctly *AF* labeled by our program to all *AF* labeled. Precision answers the following: how many of those who we labeled as *AF* are actually *AF*?

$$\frac{TP}{TP+FP} \quad (7)$$

### e) F1-score

F1-score considers both precision and sensitivity. It's the average of the precision and sensitivity (for example, you'd rather get some *healthy* labeled *AF* over leaving an *AF* labeled *healthy*).

$$\frac{TP}{TP + \frac{1}{2}(FP+FN)} \quad (8)$$

## 3. CNN and AF detection

Below is the introduction of ten methodologies for AF detection using convolutional neural networks.

### 3.1. Time-frequency analysis and CNN (detect 12 heart rhythm)

Z. Wu, T. Lan, C. Yang, and Z. Nie<sup>[1]</sup> proposed a method formed of three steps: preprocessing, time-frequency transform and the convolutional neural network. This method is able to recognize 12 different heart rhythms:

1. Normal sinus rhythm (NSR);
2. Paced rhythm (P);
3. Atrial bigeminy (AB);
4. Atrial fibrillation (AF);
5. Atrial flutter (AFL);
6. Ventricular flutter (VF);
7. First degree heart block (BI);
8. Premature ventricular contractions (PVC);

9. Sinus bradycardia (SBR);
10. Ventricular tachycardia (VT);
11. Supraventricular tachy-arrhythmia (SVTA);
12. Noise and signal contaminated by noise.

The fact that this method has not specialized only on atrial fibrillation, and that its output consists in listing the percentage of the input belonging to these 12 classes, makes it in addition to having given excellent results (97%) one of the best.

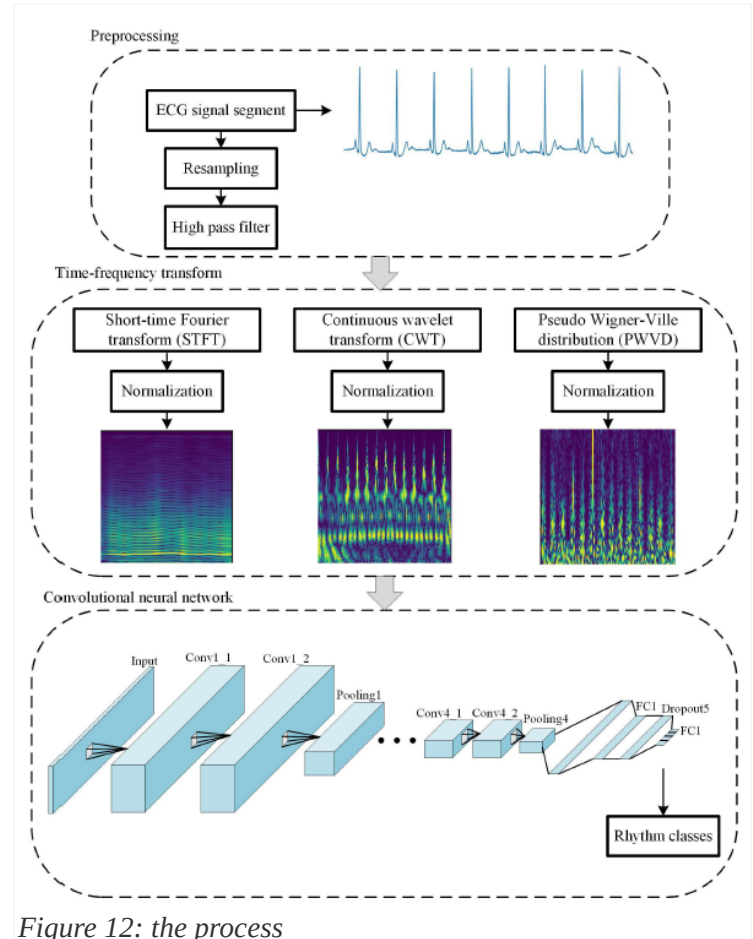


Figure 12: the process

#### a) Database

The authors used 6 different databases from PhysioNet (that are labeled with the 12 different heart rhythms listed before):

1. MIT-BIH arrhythmia;
2. MIT-BIH malignant ventricular arrhythmia;
3. MIT-BIH atrial fibrillation;
4. Long-term AF;
5. MIT-BIH normal sinus rhythm;
6. MIT-BIH noise stress test.

#### b) Preprocessing

The ECG signal is formed of 12 leads and need to be pre-processed, so the signal was splitted in segments of 10s (using rhythms annotations included in the databases) and resampled to 125Hz. Then a second order high-pass

filter with a cutoff frequency of 0.5Hz was used to remove the baseline of each signal.

The “MIT-BIH noise stress test” (it contains noise from baseline wander, muscle artifact and electrode motion artifact) was used to add noise to signals to make the network more reliable to noise. To the signal  $x$  was added the noise signal  $n$  multiplied by the gain  $a$

$$y = x + a \cdot n \quad (9)$$

The gain  $a$  is calculated using the power of the raw signal and the power of the noise signal.

### c) Time-frequency transform

Time-frequency analysis is one of the important methods to process non-stationary signals and provides information on time domain and frequency domain.

The authors use three different methods: short-time Fourier transform (STFT), continuous wavelet transformation (CWT) and pseudo Wigner-Ville distribution (PWVD).

The STFT of the sequence  $x(t)$  is defined as:

$$STFT(t, w) = \int_{-\infty}^{+\infty} x(\tau) \cdot w(\tau - t) \cdot e^{-jw\tau} dt \quad (10)$$

where  $w(t)$  is the window function of 2s and the step length is 0.08s. The CWT is defined as:

$$CWT(a, b) = \int_{-\infty}^{+\infty} \frac{x(t)}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (11)$$

where  $a$  is the scale factor and  $b$  is the time shift factor,  $\psi(t)$  is the Morlet wavelet basis:

$$\psi(t) = \exp\left(\frac{-t^2}{2}\right) \cdot \cos(5t) \quad (12)$$

At last, the PWVD is defined as:

$$PWVD(t, w) = \int_{-\infty}^{+\infty} w(\tau) \cdot x\left(t + \frac{\tau}{2}\right) \cdot \bar{x}\left(t - \frac{\tau}{2}\right) \cdot e^{-jw\tau} d\tau \quad (13)$$

with  $\bar{x}$  the complex conjugate of  $x$ .

After the time-frequency transform (one of this) we can calculate the time-frequency distribution matrix as:

$$|Y(a, b)| = \sqrt{Y^2 \cdot (a, b)} \quad (14)$$

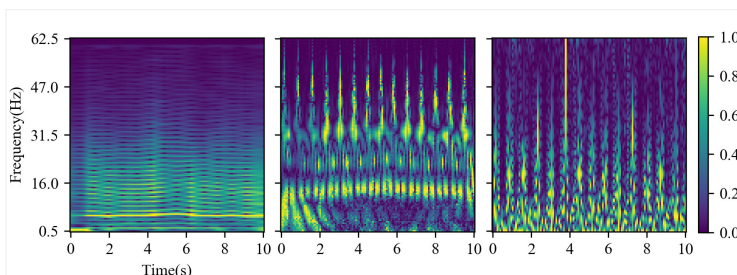


Figure 13: STFT, CWT & PWVD

### d) CNN

The network is formed by a convolutional block and a fully connected layer.

The convolutional block is formed by the concatenation of a block repeated 4 times. This block is formed by:

1. Convolution layer: 64 kernels (5×5 step 1), padding same, ReLU;
2. Convolution layer: 64 kernels (5×5 step 1), padding same, ReLU;
3. Max pool layer: (2×2);
4. Dropout layer: dropout rate 0.3.

The fully connected layer is formed by:

1. Flatten layer;
2. Fully Connected layer: 128 cells, ReLU;
3. Dropout layer: dropout rate 0.5;
4. Fully Connected layer: 12 outputs, Softmax.

The last layer outputs the probability for each of the 12 different heart rhythms.

### e) Training method

The authors used a 5-fold cross validation with the cross-entropy as the loss function:

$$\ell = - \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) \quad (15)$$

and the root mean square prop (RMSprop) method was used to update the weights of the network:

$$w_t = w_{t-1} - \alpha \cdot \frac{dw}{\sqrt{(s_{dw})} + \epsilon} \quad (16)$$

where  $\alpha$  is the learning rate,  $s_{dw}$  is the accumulation of momentum and  $\epsilon > 0$ .

### f) Results

Four metrics were used to evaluate the three methods, the ranking was: STFT, CWT, PWVD.

Table 1: classification performance

Method	Accuracy	Sensitivity	Specificity	F1
STFT	96.65	96.47	99.68	96.27
CWT	95.26	94.71	99.55	94.80
PWVD	92.07	92.19	99.25	92.26

## 3.2. Lightweight CNN

D. Lai, X. Zhang, Y. Bu, Y. Su and C. Ma<sup>[2]</sup> proposed a method formed of three steps: preprocessing, extraction of cardiac rhythms features and a lightweight CNN.

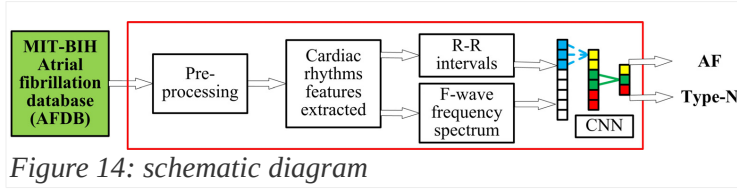


Figure 14: schematic diagram

### a) Database

The database used was the MIT-BIH atrial fibrillation database from PhysioNet formed of 23 long-term ECG with a sample rate of 250Hz that are labeled with 4 different heart rhythms:

1. Atrial fibrillation (AF);
2. Atrial flutter (AFL);
3. Atrial-ventricular junctional rhythm (AVJ);
4. Other rhythms (N).

### b) Preprocessing

In this step each recording is divided in segmented of 10s, a median filter (0.5Hz) and a band-pass filter (100Hz) are applied, to reduce noise and increase precision and efficiency of the network. Early normalization was used on the data:

$$\hat{x} = \frac{x - \mu}{\sigma} \quad (17)$$

and batch normalization was used to accelerate the training during the back-propagation, using the following update equation (18), where  $\beta$ ,  $\gamma$  are trained at each iteration on  $k$  iterations:

$$y^{(k)} = \gamma^{(k)} \cdot \hat{x}^{(k)} + \beta^{(k)} \quad (18)$$

### c) Extraction of cardiac rhythms features

A convolutional layer is used to obtain a feature maps  $c$ :

$$c = \sigma \left( b + \sum_n w \cdot x_{i+k} \right) \quad (19)$$

where  $\sigma$  is the activation function (ReLU),  $b$  is the bias of the activation map,  $n$  the size of the kernel,  $w$  is the weight and  $k$  is the stride. The result of the convolutional layer is fed into a pooling layer  $P$ , to reduce the dimension of the feature map  $c$  and the number of parameters, as follows:

$$P = \max_{t \in T} c_{i+ts} \quad (20)$$

where  $t$  is the pooling window size and  $s$  the stride.

Then two different test were done (as shown in Figure 14), one using raw data and one using ECG signal analysis:

- R-wave detection;
- R-R interval calculation;

- F-wave transformation (in Figure 6).

The presence of numerous low-amplitude F-waves instead of P-wave can be found in AF, jointly with asymmetrical R-R intervals.

### d) Lightweight CNN

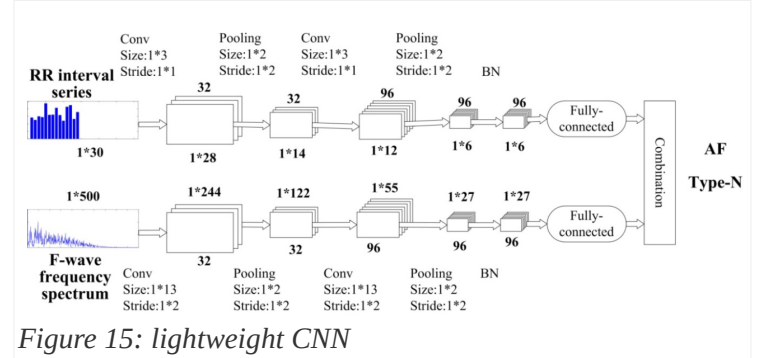


Figure 15: lightweight CNN

A lightweight CNN was used as a binary classifier (details are shown in Figure 15), then back-propagation was used to reduce the loss.

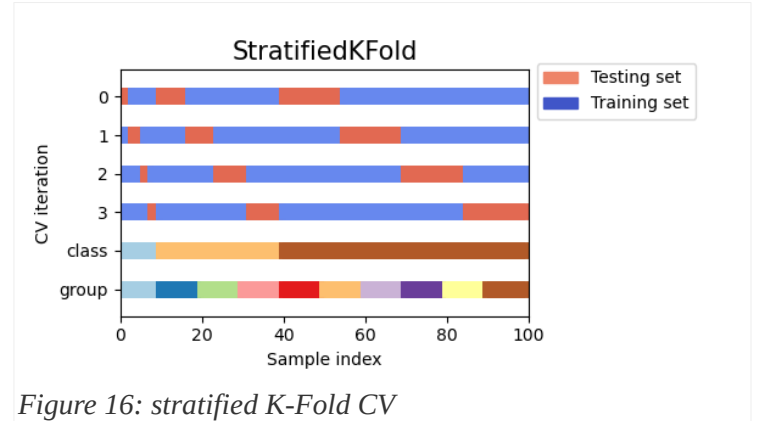


Figure 16: stratified K-Fold CV

A stratified 5-fold CV was used to tune both the model architecture and the hyper-parameters of the CNN and evaluate the model performance.

### e) Results

Three metrics were used to evaluate the two methodology:

1. R-R intervals + F-wave spectrum;
2. Raw data.

Table 2: classification performance

Low level features	Accuracy	Sensitivity	Specificity
RRI+FWS	97.5	97.8	97.2
Raw data	86.3	89.5	82.7

### 3.3. Multi-scale decomposition enhanced residual CNN

X. Cao, B. Yao and B. Chen<sup>[3]</sup> proposed a signal decomposition via derived wavelet frames and two different CNN models: MSResNet and FDResNet.

## a) Database

The database used from the PhysioNet Challenge 2017 is contributed by AliveCor (a manufacturer of single-channel ECG device) and is formed of 8528 single short ECG lead recordings, each of which is form individual customer of AliveCor, with a sample rate of 300Hz that are divided in 4 categories:

1. AF rhythm (A);
2. Normal rhythm (N);
3. Other rhythm (O);
4. Noisy recordings ( $\sim$ ).

## b) Preprocessing

In this step each recording is divided in segmented of 9s and derived wavelet frames (DWFs) is applied.

As you can see in Figure 17 implicit dual-tree complex wavelet packets (IWP) are constructed based on dyadic dual-tree CWP (DDCWPs).

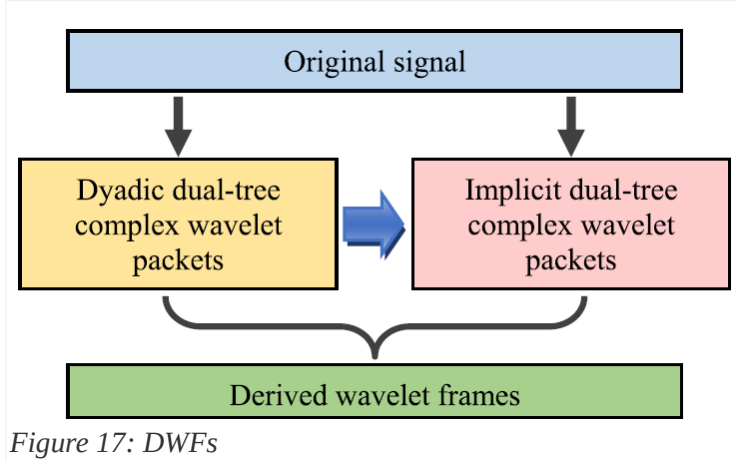


Figure 17: DWFs

To derive the IWPs, assuming  $x\{n\}$  is the ECG signal, then:

1. Perform a dual-tree wavelet packet decomposition, with  $k$  as the number of decomposition layers,  $j$  as the sequence number of the sub-signal.  $x\{n\}$  is transformed into a set of sub-signals:

$$D_k = \{D_k^j(n) : j=1, 2, \dots, 2^k\} \quad (21)$$

2. Rearrange the content of  $D_k$  according to the central frequency as  $R_k = \{R_k^j(n) : j=1, 2, \dots, 2^k\}$ , let:

$$j = \sum_{m=0}^{k-1} 2^m \cdot n_m + 1 \quad (22)$$

the binary coding of the index  $j$ , and construct a new index as follows

$$\hat{j} = \sum_{m=0}^{k-1} 2^m \cdot \hat{n}'_m + 1 \quad (23)$$

where the parameter  $\hat{n}'_m$  is defined as

$$\hat{n}'_m = \begin{cases} n_m, & m=k-1 \\ \text{mod}(n_m + n_{m+1}, 2) & m=0, 1, \dots, k-2 \end{cases} \quad (24)$$

3. Generate the implicit wavelet packet with the following equation:

$$iwp_k^j(n) = R_k^{2^j}(n) + R_k^{2^{j+1}}(n), \quad 1 \leq 2^{k-1} - 1 \quad (25)$$

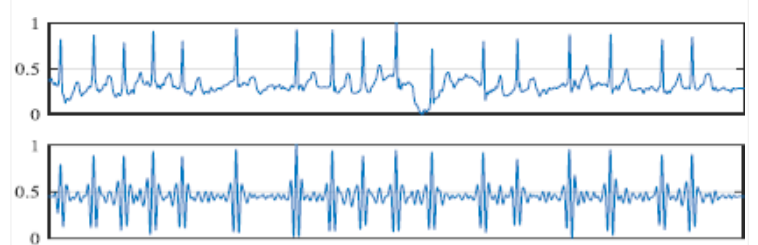


Figure 18: AF rhythm & the reconstructed sub-signal

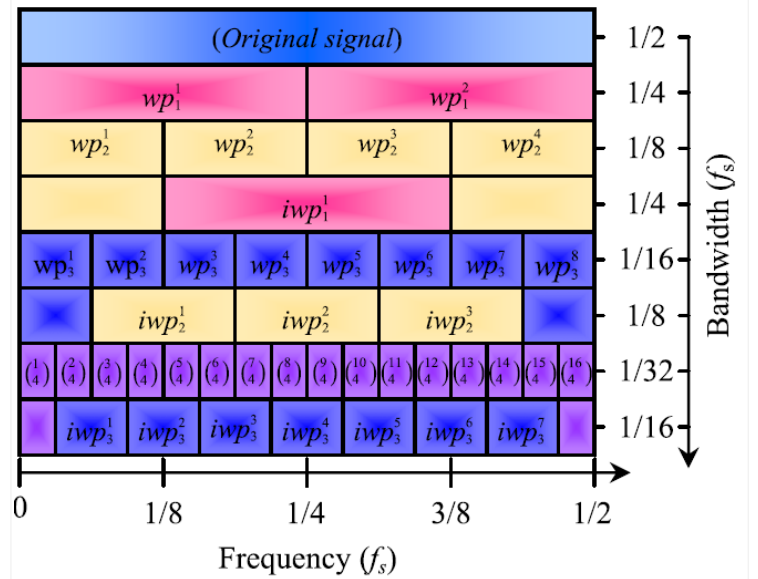


Figure 19: frequency-scale topology of the derived wavelet packet frame

In Figure 19 it can be seen that the center frequency of the derived wavelet packet is the band boundary of the traditional binary wavelet packet, thereby improving the ability of the algorithm to extract the information of the transition band. In Figure 18 the result of the DWFs on the AF rhythm signal.

## c) MSResNet

Multi-scale decomposition enhanced fast down-sampling residual CNN consists of three parallel FDResNet, same structure but independently trained by reconstructed samples of different scales (as you can see in Figure 20), each of them has learned differ-

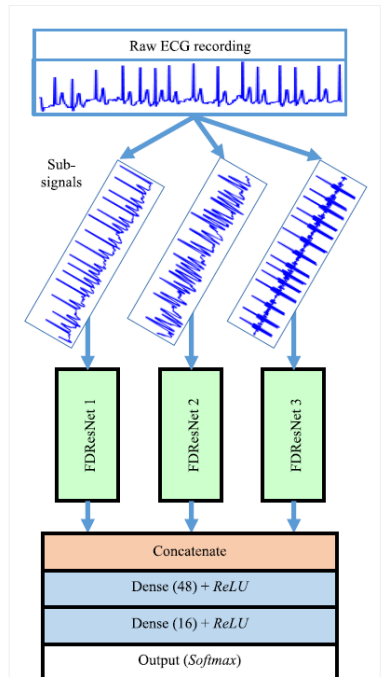


Figure 20: MSResNet



ent features, and the independent classification capabilities are different. The predictions are connected into a small neural network that learn the end-to-end characteristics of the three, and higher recognition accuracy can be obtained.

#### d) FDResNet

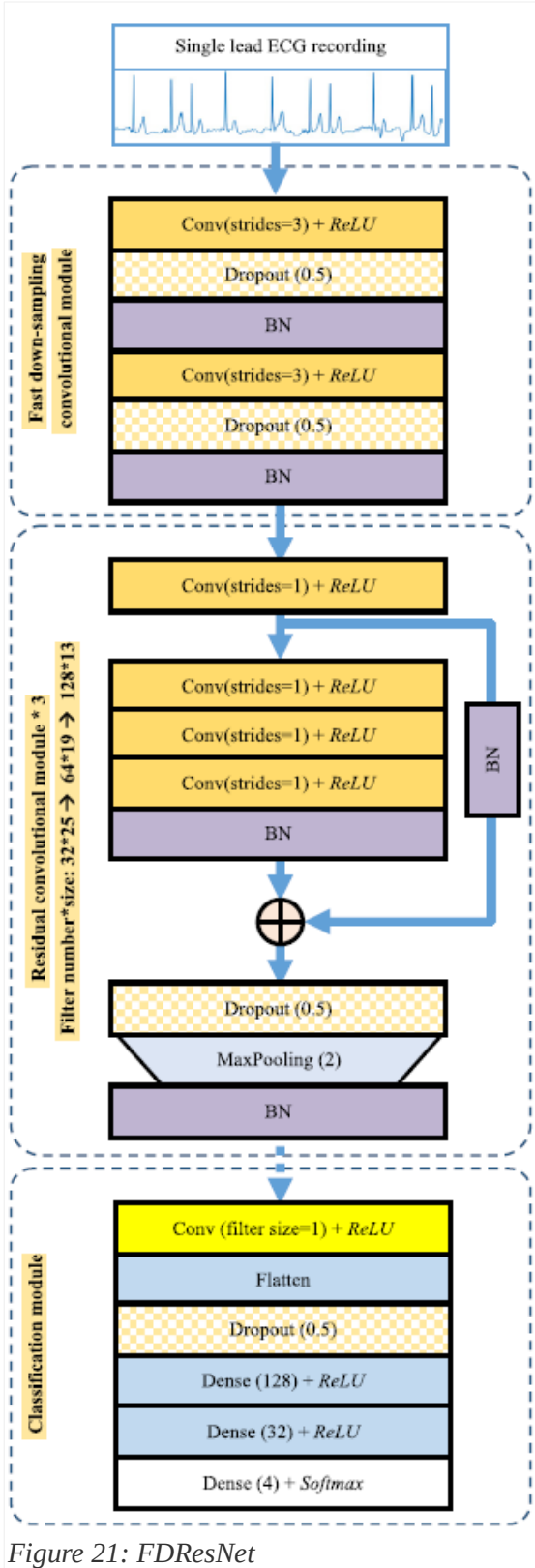


Figure 21: FDResNet

Fast down-sampling residual CNN is mainly composed of: a fast down-sampling module, a residual convolution module, and a classification module, as you can see in Figure 21 the full network structure is composed of 3 module:

1. Fast down-sampling convolutional module: is formed by  $2 \times$  convolutional layer (ReLU) with random dropout layer and a batch-normalization layer to enhance the generalization of the model. This module effectively reduces the calculation of subsequent DN, reduces data redundancy and facilitates model learning;
2.  $3 \times$  residual convolutional module: formed by convolutional layers in series and residual short circuit. The width of the 3 residual convolution modules is gradually increased. All of them use a max-pooling layer to down-sample the feature vectors;
3. Classification module: consist of a convolutional layer (to reduce the dimension of the feature vectors), a flatten layer, a random dropout layer (to prevent overfitting), 2 full connection layers (ReLU) and a softmax classifier.

#### e) Results

6-Fold CV was used to train the network. FDResNet can learn effective classification features from time domain ECG waveform. Based on different coupling strategies (concatenate layer in Figure 20) MSResNet gives the results shown in Table 3.

Table 3: FDResNet classification performance

Sub-band	Normal	AF	F1
Raw	0.8059	0.9816	0.8702
wp <sup>1</sup> <sub>3</sub>	0.7982	0.9757	0.8766
wp <sup>1</sup> <sub>4</sub>	0.8215	0.9751	0.8973

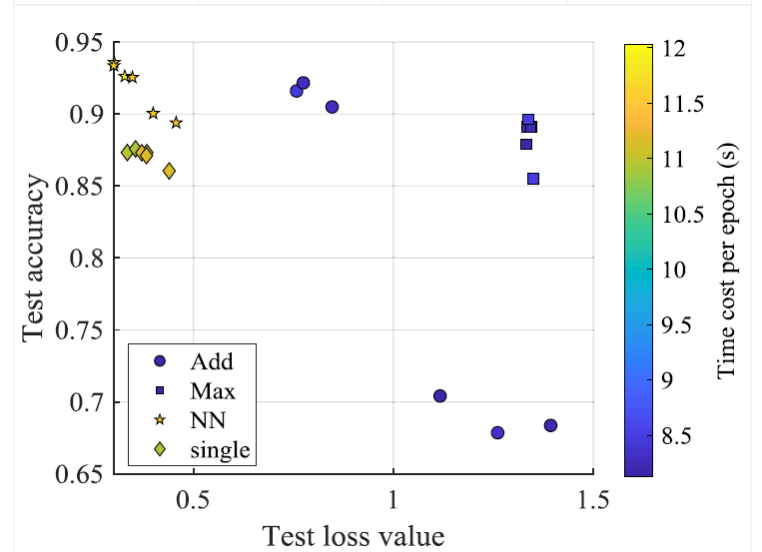


Figure 22: performance of different coupling methods

### 3.4. Dual heartbeat coupling based on CNN

X. Zhai and C. Tin<sup>[4]</sup> propose to transform the beats into a dual beat coupling matrix (2D) as input of the CNN.

## a) Database

The authors use the MIT-BIH arrhythmia database from PhysioNet, 48 records of 30min two-channel ECG signals. These were filtered with a band-pass filter (0.1Hz – 100Hz) and digitized at 360Hz.

## b) Preprocessing

The beat was segmented such that it was centered around the R peak (using a max interval of 20s) as in the left part of Figure 23. Because each segment can have different length, they were scaled into the same length  $M$ .

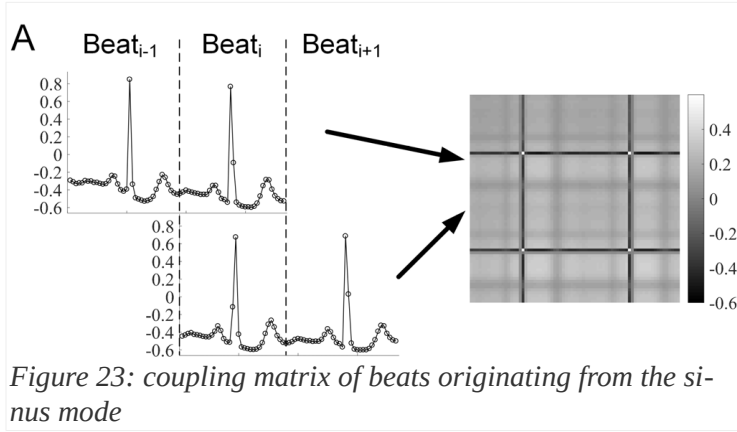


Figure 23: coupling matrix of beats originating from the sinus mode

A series of three adjacent beats is taken into account. The first pair is  $(Beat_{i-1}, Beat_i)$  denoted as  $DualBeat_{i-1,i}$ , the second pair is  $(Beat_i, Beat_{i+1})$  denoted as  $DualBeat_{i,i+1}$ , as shown in the left part of Figure 23.

Then a coupling matrix (CM) with size  $M \times M$  is computed as:

$$CM = [DualBeat_{i-1,i}[1], \dots, DualBeat_{i-1,i}[M]] \times [DualBeat_{i,i+1}[1], \dots, DualBeat_{i,i+1}[M]]^T \quad (26)$$

The result is shown in the right part of Figure 23 (an example on a supraventricular ectopic beat is shown in Figure 24).

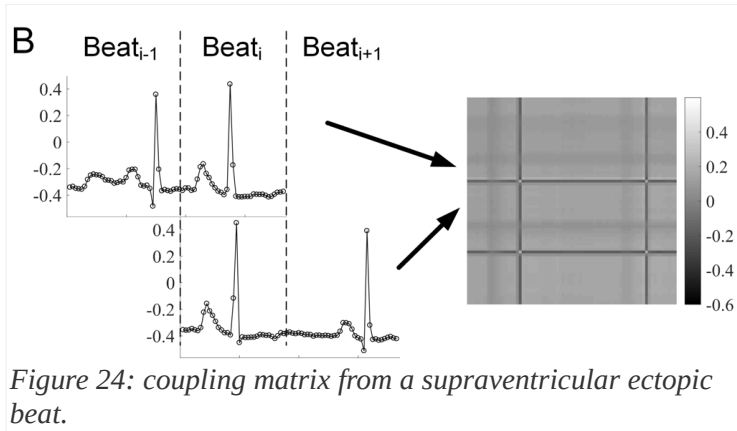


Figure 24: coupling matrix from a supraventricular ectopic beat.

## c) CNN classifier

The network (Figure 25) is formed by:

1. Input ( $73 \times 73$ );
2. Convolutional l. with ReLU ( $50 @ 66 \times 66$ )
3. Maximum sub-sampling l. ( $50 @ 33 \times 33$ )

4. Convolutional l. with ReLU ( $100 @ 24 \times 24$ )
5. Average sub-sampling l. ( $100 @ 8 \times 8$ )
6. Convolutional l. with ReLU ( $150 @ 4 \times 4$ )
7. Fully connected layer with dropout (150);
8. Softmax loss layer (5).

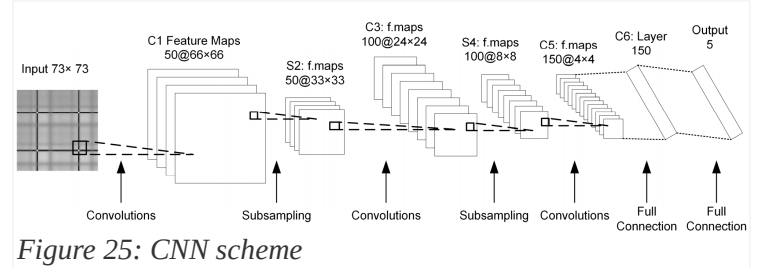


Figure 25: CNN scheme

## d) Results

Results show that the method perform better on VEB.

Table 4: classification performance

Beats	Accuracy	Sensitivity	Specificity	Precision
VEB <sup>2</sup>	98.6	93.8	99.2	92.4
SVEB <sup>3</sup>	97.5	76.8	98.7	74.0

## 3.5. CNN with SVM<sup>4</sup>

Z. Li, X. Feng, Z. Wu, C. Yang, B. Bai and Q. Yang<sup>[5]</sup> proposed the following architecture (as in Figure 26): original data will be calculated through convolution layer by feed-forward channel, then the result will be fed to a max-pooling layer and stretched as a one-dimensional vector through a flatten layer. Finally, the vector is trained into a SVM classifier.

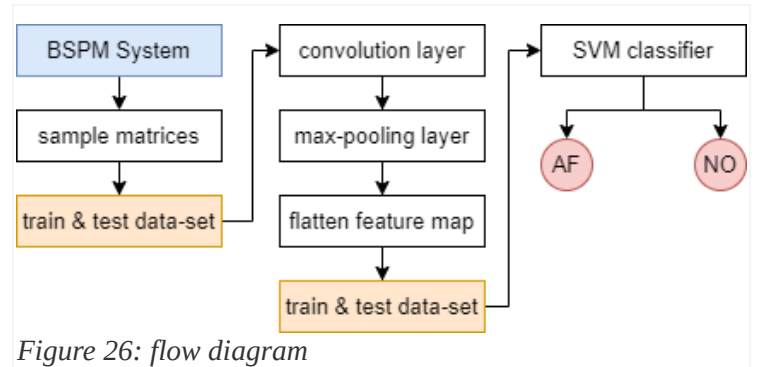


Figure 26: flow diagram

## a) Database

The data-set were provided and supported by West China Hospital<sup>5</sup> in collaboration with Sichuan University. It consist of ECG signals from 14 patients with AF, acquired via the preoperative body surface potential mapping (BSPM) and an integrated back-end acquisition system (NeuroScan). BSPM consist of 74 front electrode

- 2 Ventricular ectopic beats
- 3 Supraventricular ectopic beats
- 4 Support Vector Machine
- 5 <http://www.wchscu.cn>

points and 54 on the back, with a sampling frequency of 1kHz. The method is compatible with the standard 12-lead electrode coordinates, and contains more spatiotemporal information.

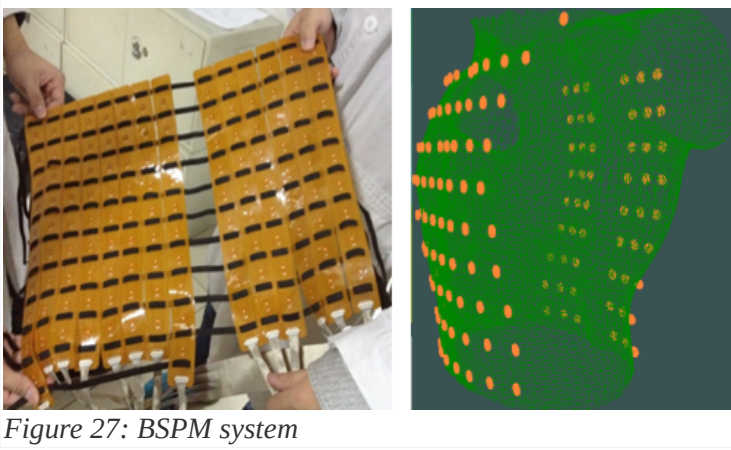


Figure 27: BSPM system

## b) Preprocessing

During the acquisition interference are removed via a 50Hz notch filter, then raw data were processed:

1. A zero-phase, third-order Butterworth filter (0.5-100Hz);
2. An absolute value filter;
3. A third-order Butterworth filter (20Hz cutoff).

Data are normalized (as in Eq. (17)), and then 160s are kept, from the center, of each acquisition. 10-fold cross validation is used.

## c) CNN

Convolution layers use ReLU as activation function. The authors choose Adam<sup>[f]</sup> (Adaptive Moment Estimation) as the algorithm for gradient descent with the following update rule:

$$w_{t+1} = w_t - \alpha \cdot \frac{\hat{m}_w}{\sqrt{\hat{v}_w + \epsilon}} \quad (27)$$

with gradient  $w$ , step-size  $\alpha$ , momentum factor  $\epsilon$ , and biased first moment estimate  $v_w$ , and second raw moment estimate  $m_w$ .

## d) SVM

The decision function with kernel used in this project is:

$$g(x) = \text{sign}\left(\sum_{i=1}^N \hat{\alpha}_i \cdot y_i \cdot K(\tau, x) + \hat{b}\right) \quad (28)$$

The authors tried three different kernels:

1. Linear kernel

$$K(\tau, x) = (\tau \cdot x + 1) \quad (29)$$

2. Polynomial kernel, where  $q$  is the degree.

$$K(\tau, x) = (\tau \cdot x + 1)^q \quad (30)$$

3. Gaussian kernel, where  $\delta$  is the width.

$$K(\tau, x) = \exp\left(-\frac{\|\tau - x\|^2}{2\delta^2}\right) \quad (31)$$

## e) Results

As in Figure 28 the RBF (Radial basis function) kernel give the best result:

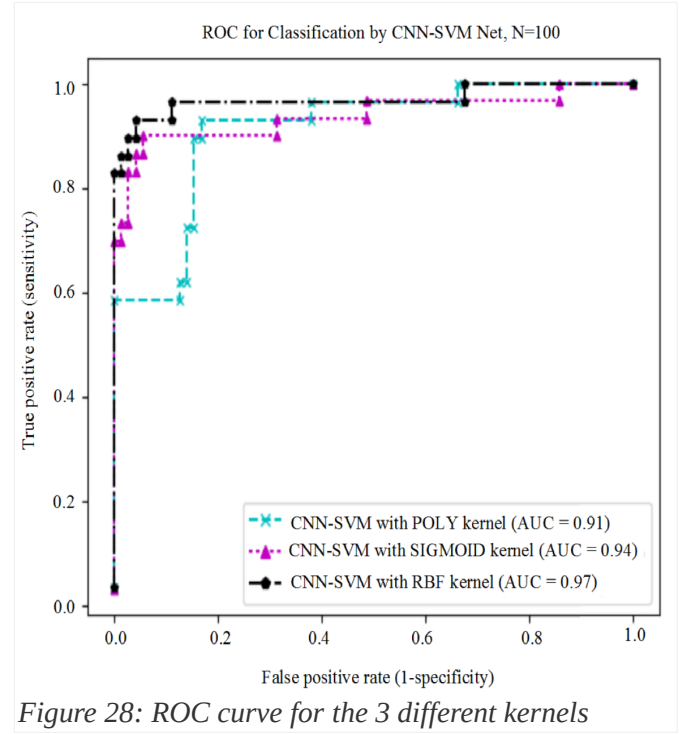


Figure 28: ROC curve for the 3 different kernels

Table 5: classification performance

Accuracy	Sensitivity	Specificity
96.06	88.48	96.29

## 3.6. Deep CNN

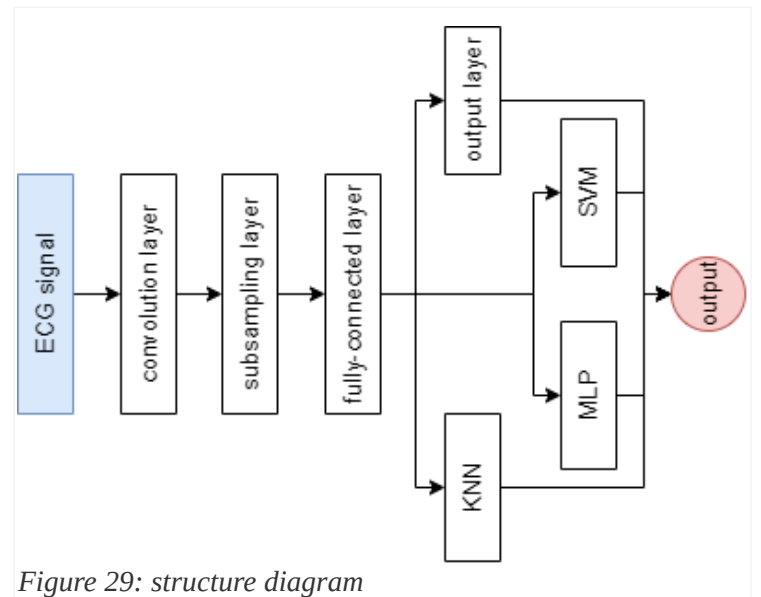


Figure 29: structure diagram

B. Pourbabaee, M. Roshtkhari and K. Khorasani<sup>[6]</sup> investigate the Paroxysmal atrial fibrillation (PAF) classification problem under two experiments:

1. An end-to-end CNN network is applied to extract the features and classify them, the first 3 layers (convolution layer, subsampling layer, fully-connected layer) in Figure 29 and the output layer;
2. The first 3 layers are used jointly with one of this classifiers: KNN<sup>6</sup>, SVM<sup>7</sup>, MLP<sup>8</sup>.

### a) Database

The PAF prediction challenge database, from PhysioNet, was used in this project, it consist of 150 two-channel ECG recordings each of 30min, half of the patients are healthy. The data-set was divided in train and test set (ratio 70/30).

### b) CNN

As mentioned before the CNN is composed by:

1. Convolution layer (size 32);
2. Subsampling layer (size 128);
3. Fully-connected layer (size 64).

The CNN use a stochastic gradient descent (SGD) algorithm, it's robust to distortions, displacements, translations and noise effects on the input. The loss function used was the log-likelihood (NLL):

$$NLL(\theta, D) = - \sum_{i=0}^{|D|} \log(P(Y = y_i | x_i, \theta)) \quad (32)$$

with parameters  $\theta$ , and training data  $D$ , in addition was used an  $L_2$  regularization mechanism where  $\lambda$  is a regularization parameter:

$$E(\theta, D) = NLL(\theta, D) + \lambda \|\theta\|_2 \quad (33)$$

The following classifiers receives from the fully-connected layer a feature vector of 64 elements.

### c) KNN

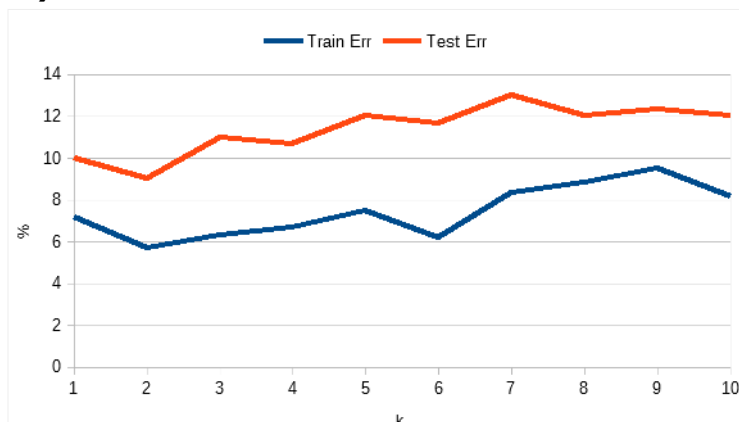


Figure 30: KNN learning curve

Is a type of instance-based learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed

from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

The optimal choice of the value  $k$  is highly data-dependent: in general a larger  $k$  suppresses the effects of noise, but makes the classification boundaries less distinct as shown in Figure 30. The best result is obtained with  $k=2$ .

### d) SVM

Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a hyperplane which separates the data into classes.

The authors tried two different kernels, a linear kernel

$$K(\tau, x) = (\tau \cdot x + 1) \quad (34)$$



Figure 31: linear kernel - learning curve

and a Gaussian kernel

$$K(\tau, x) = \exp\left(-\frac{\|\tau - x\|^2}{2\delta^2}\right) \quad (35)$$

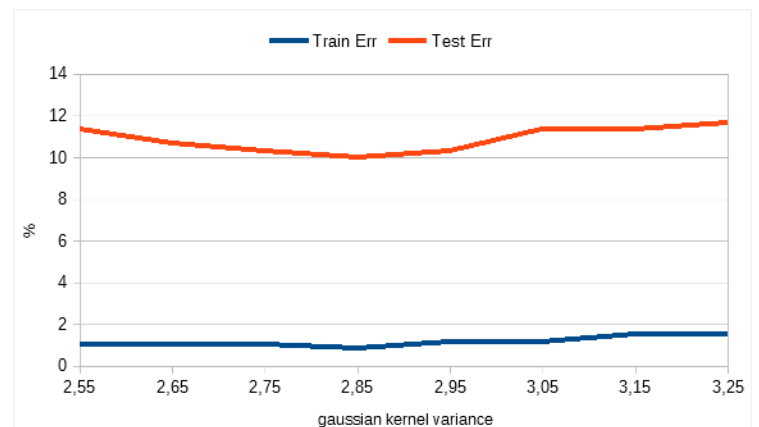


Figure 32: Gaussian kernel - learning curve

The linear kernel obtain his best result with  $\tau=25$  with a CCR of 87.67%. The Gaussian kernel obtain better result compared to the latter using a kernel variance of  $\delta=2.85$  and a  $\tau=11$  with a CCR of 90%.

6 K-nearest neighbors algorithm

7 Support-vector machines

8 Multilayer perceptron



Figure 33: Gaussian kernel - learning curve

**e) MLP**

Multi-layer perceptron is a type of network where multiple layers of a group of perceptron are stacked together to make a model. The best selection for the number of neurons in the hidden layer is obtained with 37.

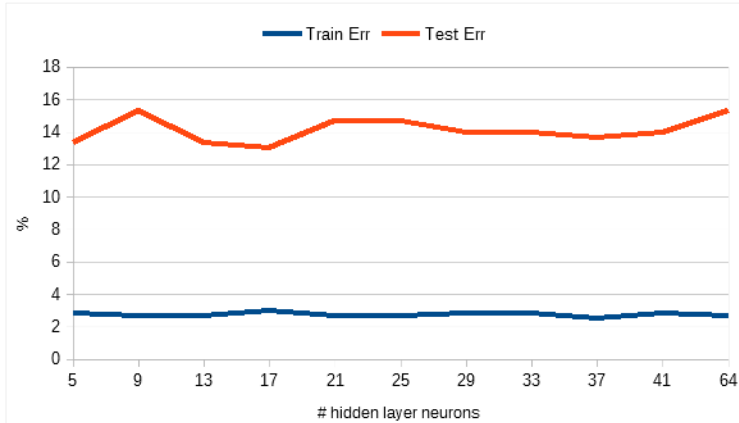


Figure 34: MLP learning curve

**f) Results**

Therefore, the KNN is capable of screening the PAF patients more accurately than the other classifier methods, and it can be proposed as the most suitable methodology. We should emphasize that the proposed CNN network is more appropriate to be utilized as a feature learning mechanism than a classifier, since the patient screening performance can be improved by combining the CNN with other conventional classifiers as compared to only an end-to-end CNN architecture.

Table 6: classification performance

Method	Sensitivity	Specificity	CCR
End-to-end CNN	0.7647	0.9456	0.8533
KNN	0.9020	0.9048	0.9100
Linear SVM	0.8758	0.8707	0.8767
Gaussian SVM	0.8627	0.9320	0.9000
MLP	0.8235	0.9116	0.8633

**3.7. DCN with focal loss and image generation**

M. Al Rahhal, Y. Bazi, H. Almubarak, N. Alajlan and M. Al Zuair<sup>[7]</sup> proposed a two-stage CNN for carrying classification. The first module aims to convert the ECG signal to an image using an opportune generative network (Figure 35). The second one called discriminative network mainly based on dense convolutional networks (DCN) takes the output of the generative module and carries out classification as in standard image classification paradigms (Figure 36).

Data are said to suffer the “Class Imbalance Problem” when the class distributions are highly imbalanced. In this context, many classification learning algorithms have low predictive accuracy for the infrequent class. The authors propose to exploit the focal loss (FL) to down-weight the loss for the well-classified ECG beats.

**a) Generative network**

The conversion of the ECG signal into an image is achieved through:

1. 2 fully-connected layers (size of 1024 and 1568);
2. A reshape layer: from (1568, 1) to (32, 7, 7) with:
 
$$X_t^r = \text{reshape}(\text{ReLU}(X_t))_7^7 \quad (36)$$
3. 2 up-sampling/convolution layer blocks, composed by:
  - i. A layer of up-sampling;
  - ii. A convolution;
  - iii. Batch normalization, that allows each layer of the network to learn by itself a little more independently of other layers;
  - iv. ReLU activation;
  - v. Dropout regularization layers, used to prevent overfitting during the training phase, by randomly dropping nodes from the hidden layers.
4. A convolution layer: from (64, 28, 28) to (3, 28, 28).

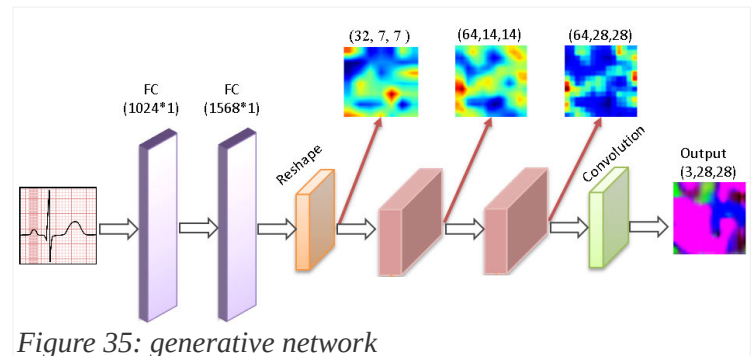


Figure 35: generative network

**b) Discriminative network**

It takes the images produced by the generative module as its input and classifies them into one of this four class:

1. N: Normal beat;
2. F: Fusion of ventricular and normal beat;
3. S: Supraventricular premature beat;
4. V: Premature ventricular contraction.

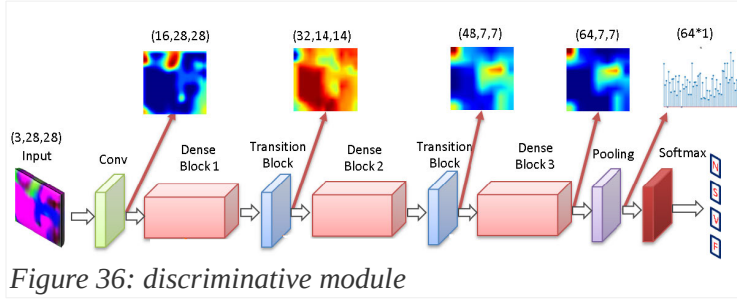


Figure 36: discriminative module

As shown in Figure 36 it consist mainly of dense block (Figure 37) to enhance the information sharing and information flow between layers, direct connections from any layer to its subsequent layers were introduced. Between every dense block there is a transition block that perform convolution and pooling to down-sample the data.

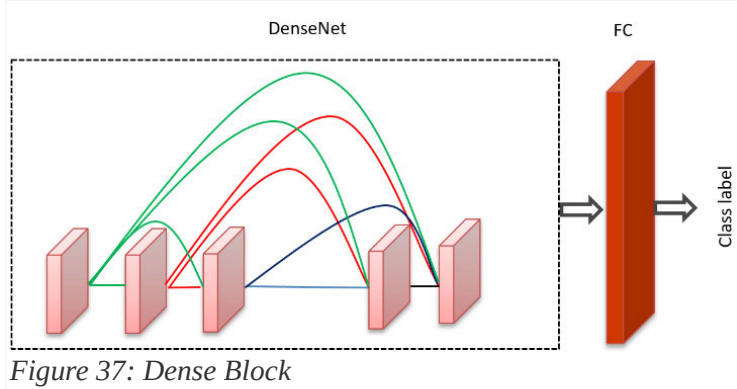


Figure 37: Dense Block

The output  $x_\ell$  of each layer of the DenseNet (Figure 37) is, with  $[x_0, x_1, \dots, x_{\ell-1}]$  the concatenation of the output of the previous layers and  $H_\ell$  a non-linear transformation (like ReLU, pooling, batch normalization, convolution, etc...):

$$x_\ell = H_\ell([x_0, x_1, \dots, x_{\ell-1}]) \quad (37)$$

### c) Focal loss

There are two popular ways to address data imbalance in ML: down-sampling dominant cases (or oversampling minority cases), changing weights in the loss function. The authors follows the second way and proposes a new loss function to address this problem, the focal loss. Focal loss is derived from cross-entropy loss:

$$CE(p, y) = \begin{cases} -\log(p), & y=1 \\ -\log(1-p), & \text{else} \end{cases} \quad (38)$$

$$p_t = \begin{cases} p, & y=1 \\ 1-p, & \text{else} \end{cases} \quad (39)$$

$$CE(p, y) = CE(p_t) = -\log(p_t) \quad (40)$$

in focal loss a modulating factor,  $(1-p)^y$ , is added to cross-entropy loss:

$$FL(p_t) = -(1-p_t)^y \log(p_t) \quad (41)$$

with the focusing parameter  $y \in [0, 5]$ .

### d) Database

The authors choose three database from PhysioNet:

1. MIT-BIH arrhythmia database: 48 records from 47 patients of 30min and 360Hz;
2. INCART (St. Petersburg Institute of Cardiological Technics): 75 records from 32 patients of 30min and 275Hz;
3. MIT-BIH supra-ventricular arrhythmia database: 78 records of 30min and 128Hz.

The training was done for 250 epochs with a batch size of 100. The algorithm for gradient descent used was Adam (learning rate 0.001).

### e) Results

The results of the three methods are shown in the following tables, FL perform slightly better.

Table 7: classification performance Ventricular ectopic beats

Method	Accuracy	Sensitivity	Specificity	Precision
FL	99.2	95.0	99.7	96.9
CE	99.1	93.4	99.8	97.5
CERE <sup>9</sup>	99.1	94.0	99.7	96.6

Table 8: classification performance Supraventricular ectopic beats

Method	Accuracy	Sensitivity	Specificity	Precision
FL	98.7	69.8	100.0	99.1
CE	98.9	79.4	99.8	94.1
CERE	98.6	69.7	99.9	97.9

## 3.8. STFT-based spectrogram and CNN

J. Huang, B. Chen, B. Yao and W. He<sup>[8]</sup> proposed to transform the ECG signals from the time domain into two-dimensional time-frequency ECG spectrograms by short-time Fourier transform. The resultant ECG spectrograms were used as input to a deep learning network.

### a) Database

The database used was MIT-BIH arrhythmia database from PhysioNet. Each recording was divided into segments of 10s and the sample rate was uniformly set to 360Hz.

## b) CNN

The network (Figure 38) is formed by 3 block of convolution (ReLU) and max-pooling which transform the input from (256×256×1) to (64×8×512), followed by a flatten layer, a dropout, and 2 dense activation (ReLU, softmax).

Each ECG data recording was transformed into an ECG spectrum image through STFT and used as input for the network.

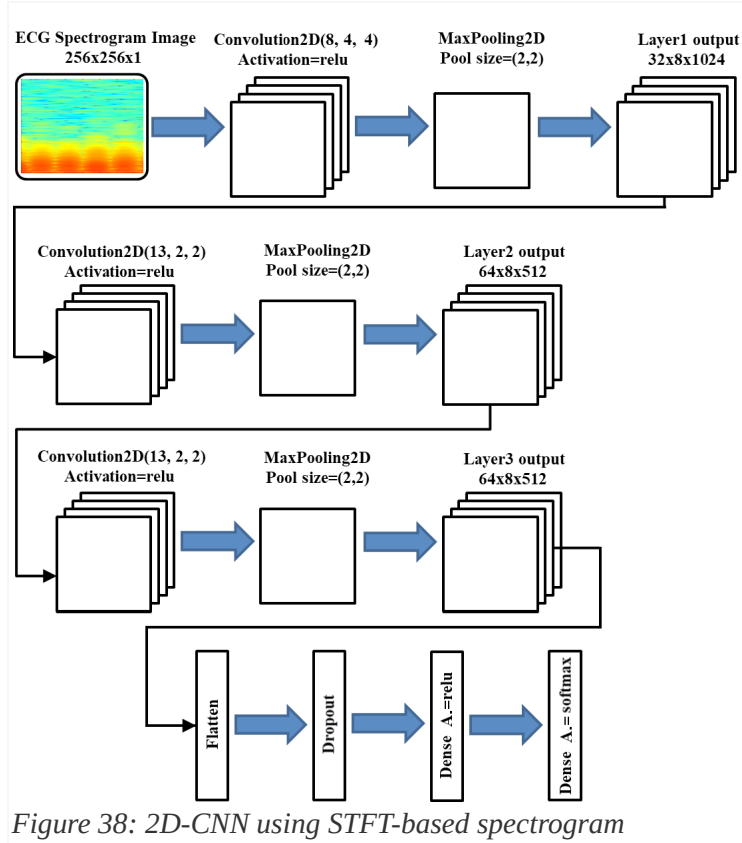


Figure 38: 2D-CNN using STFT-based spectrogram

## c) STFT

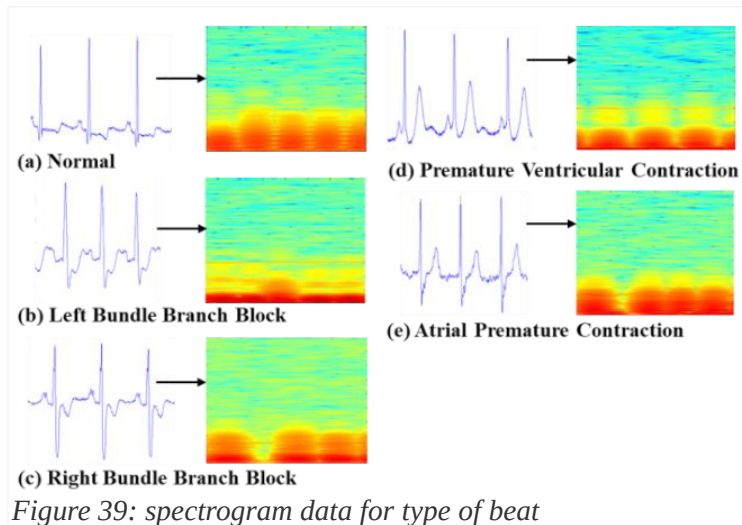


Figure 39: spectrogram data for type of beat

The short-time Fourier transform (STFT) is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes with respect to time. ECG signal record is transformed into an image (256×256) of time-frequency spectrogram by using the STFT:

$$STFT \{x[n]\} = X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (42)$$

where  $x[n]$  is the ECG signal, and  $w[n]$  is the Hanning window function:

$$w(n) = \begin{cases} 0.5 \left[ 1 - \cos\left(\frac{2\pi n}{M-1}\right) \right], & 0 \leq n \leq M-1 \\ 0, & \text{else} \end{cases} \quad (43)$$

In Figure 39 a sample application of STFT for each class's spectrogram.

## d) Results

The results are shown in the following table.

Table 9: classification performance

Model	Accuracy	Loss
2D-CNN	0.9900	0.0414

## 3.9. Gray-level co-occurrence matrix enhanced CNN

W. Sun, N. Zeng and Y. He<sup>[9]</sup> proposed to employ GLCM<sup>10</sup> for features vector description of ECG signals. CNN approach is utilized to automatically recognize the arrhythmia type from the generated 3D multi-scale GLCM.

### a) Database

In this research MIT-BIH arrhythmia database from PhysioNet was used. 7 class was used:

1. Normal beat (C1);
2. Left bundle branch block beat (C2);
3. Right bundle branch block beat (C3);
4. Premature ventricular contraction (C4);
5. Fusion of ventricular and normal beat (C5);
6. Atrial premature beat (C6);
7. Paced beat (C7).

The last 6 class are shown in Figure 41-a).

### b) Multi-scale GLCM

The GLCM is a matrix defined for a digital image. The image value refers to the gray value of the specific pixel. The value can be any, from a binary number to a 32-bit value for a color image. Note that a 32-bit image generates a  $2^{32} \times 2^{32}$  matrix. A co-occurrence matrix measures the texture mapping of the image.

For an image with  $p$  different pixel values, the  $p \times p$  co-occurrence matrix  $C$  is defined over an  $n \times m$  image  $I$ , the GLCM can be given by:

$$C_{\Delta x}(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1, & y(x) = i \wedge y(x + \Delta x) = j \\ 0, & \text{else} \end{cases} \quad (44)$$

with  $i, j$  are pixel values, and  $x$  is the spatial position in  $I$  and  $y$  the time domain signal, and  $\Delta x$  is the offset.

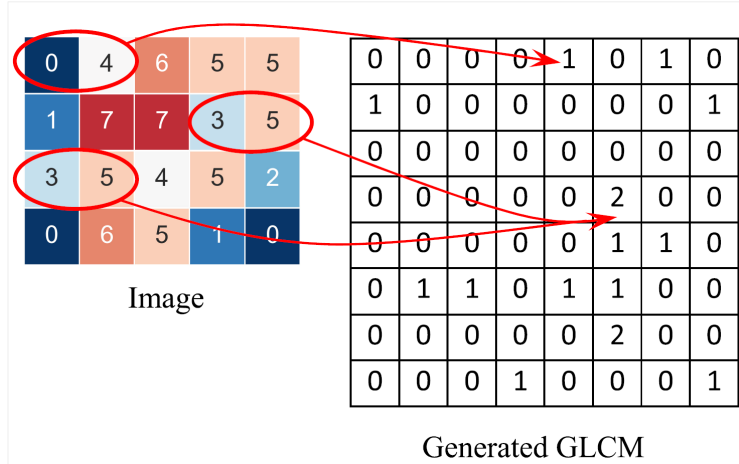


Figure 40: GLCM calculation process

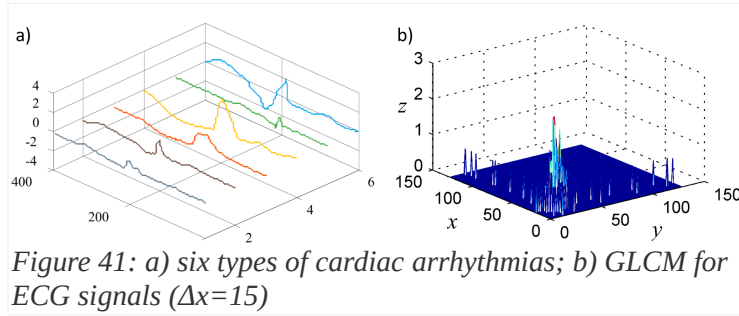


Figure 41: a) six types of cardiac arrhythmias; b) GLCM for ECG signals ( $\Delta x = 15$ )

### c) CNN

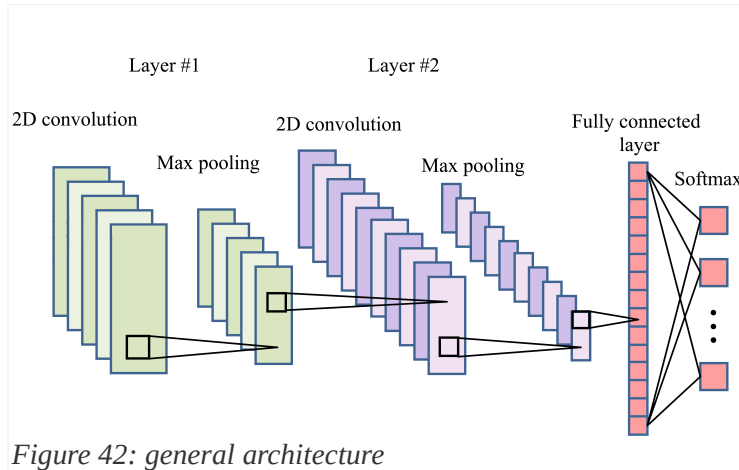


Figure 42: general architecture

For each of the 7 beats class the following steps are executed to generate the input for the network:

1. Normalize the ECG signals so that the samples can be digitized with 7-bit resolution (range from 0 to 127) with the following equation from signal  $X$ :

$$\hat{X} = \text{round} \left( 127 * \frac{X - X_{\min}}{X_{\max} - X_{\min}} \right) \quad (45)$$

2. Extract the multi-scale GLCM feature vector based of dimension  $128 \times 128 \times n$  (the authors suggest  $n=36$ );

3. Divide the feature vector in train and test set.

As shown in Figure 42 the general architecture of the network is composed by:

- A first convolution layer of 64 kernels ( $3 \times 3$ ) with ReLU followed by a max-pooling layer ( $2 \times 2$ );
- A second convolution layer of 32 kernels ( $2 \times 2$ ) with ReLU followed by a max-pooling layer;
- A softmax layer.

### d) Results

The results for all classes are in the following table.

Class	Precision	Sensitivity	F1
Normal beat	0.80	0.94	0.64
Left bundle branch block beat	1.00	1.00	1.00
Right bundle branch block beat	0.95	0.86	0.90
Premature ventricular contraction	0.90	0.95	0.92
Fusion of ventricular and normal beat	0.95	0.95	0.95
Atrial premature beat	0.85	0.85	0.85
Paced beat	1.00	0.91	0.95
Mean	0.92	0.92	0.92

## 3.10. Multi-scaled fusion of deep CNN

X. Fan, Q. Yao, Y. Cai, F. Miao, F. Sun and Y. Li<sup>[10]</sup> propose a multi-scaled fusion of deep convolutional neural network named MS-CNN.

### a) Database

The database used was Computing in Cardiology Challenge 2017 from PhysioNet, a collection of 8528 ECG records lasting from 9s to 61s, labeled with 4 classes: normal rhythm, AF rhythm (9%), other rhythm and noisy recordings. A cut-off frequency of 60Hz and a down-sampling to 120Hz was applied. All the data was normalized with Eq. (17). Because the data-set is relatively small, 10-fold cross validation was used.

### b) CNN

As in Figure 43 the network consists of two streams of 13-layer convolutional neural networks with different filter sizes in first four hidden layers and three fully connected layers after them. The four hidden layers are composed by 2 convolutional layers (with ReLU) of 64 filters, that increase by a factor of 2 after each max-pooling layer, until it reaches 512.





Method	Acc.	Sen.	Spe.	Prec.	F1
CNN with SVM					
	96.1	88.5	96.3		
Deep CNN					
End-to-end CNN		76.5	94.6		
KNN		90.2	90.5		
Linear SVM		87.6	87.1		
Gaussian SVM		86.3	93.2		
MLP		82.4	91.2		
DCN with focal loss and image generation					
FL	<b>99.0</b>	82.4	<b>99.9</b>	<b>98.0</b>	

Method	Acc.	Sen.	Spe.	Prec.	F1
CE	<b>99.0</b>	86.4	<b>99.8</b>	95.8	
CERE	<b>98.9</b>	81.85	<b>99.8</b>	97.3	
STFT-based spectrogram and CNN					
1D-CNN	90.9				
2D-CNN	<b>99.0</b>				
Gray-level co-occurrence matrix enhanced CNN					
		92.0		92.0	0.92
Multi-scaled fusion of deep CNN					
	98.1	93.8	98.8	91.8	

## 5. References

- 1) Ziqian Wu, Tianjie Lan, Cuiwei Yang, and Zhenning Nie, “A Novel Method to Detect Multiple Arrhythmias Based on Time-Frequency Analysis and Convolutional Neural Networks”, *IEEE Access*, vol. 7, pp. 170820-170830, December 2019
- 2) Dakun Lai, Xinshu Zhang, Yuxiang Bu, Ye Su, and Chang-sheng Ma, “An Automatic System for Real-Time Identifying Atrial Fibrillation by Using a Lightweight Convolutional Neural Network”, *IEEE Access*, vol. 7, pp. 130074-130084, December 2019
- 3) Xin-cheng Cao, Bin Yao, and Bin-qiang Chen, “Atrial Fibrillation Detection Using an Improved Multi-Scale Decomposition Enhanced Residual Convolutional Neural Network”, *IEEE Access*, vol. 7, pp. 89152-89161, December 2019
- 4) Xiaolong Zhai, and Chung Tin, “Automated ECG Classification Using Dual Heartbeat Coupling Based on Convolutional Neural Network”, *IEEE Access*, vol. 6, pp. 27465-27472, June 2018
- 5) Zhangjun Li, Xujian Feng, Ziqian Wu, Cuiwei Yang, Baodan Bai, and Qunqing Yang, “Classification of Atrial Fibrillation Recurrence Based on a Convolution Neural Network With SVM Architecture”, *IEEE Access*, vol. 7, pp. 77849-77856, December 2019
- 6) Bahareh Pourbabae, Mehrsan Javan Roshtkhari, and Khashayar Khorasani, “Deep Convolutional Neural Networks and Learning ECG Features for Screening Paroxysmal Atrial Fibrillation Patients”, *IEEE Transactions On Systems, Man, And Cybernetics: Systems*, vol. 48, n. 12, pp. 2095-2104, December 2018
- 7) Mohamad Mahmoud Al Rahhal, Yakoub Bazi, Haidar Almubarak, Naif Alajlan, and Mansour Al Zuair, “Dense Convolutional Networks With Focal Loss and Image Generation for Electrocardiogram Classification”, *IEEE Access*, vol. 7, pp. 182225-182237, December 2019
- 8) Jingshan Huang, Binqiang Chen, Bin Yao, and Wangpeng He, “ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Network”, *IEEE Access*, vol. 7, pp. 92871-92880, July 2019
- 9) Weifang Sun, Nianyin Zeng, and Yuchao He, “Morphological Arrhythmia Automated Diagnosis Method Using Gray-Level Co-Occurrence Matrix Enhanced Convolutional Neural Network”, *IEEE Access*, vol. 7, pp. 67123-67129, June 2019
- 10) Xiaomao Fan, Qihang Yao, Yunpeng Cai, Fen Miao, Fangmin Sun, and Ye Li, “Multiscaled Fusion of Deep Convolutional Neural Networks for Screening Atrial Fibrillation From Single Lead Short ECG Recordings”, *IEEE Journal Of Biomedical And Health Informatics*, vol. 22, n. 6, pp. 1744-1753, November 2018
- 11) Vincenzo Piuri, “Intelligent Systems”, lecture notes, Università degli Studi di Milano, 2003
- 12) Nicolò Cesa-Bianchi, “Statistical Methods for Machine Learning”, lecture notes, Università degli Studi di Milano, 2019
- 13) Nicola Carbonaro, “Natural and Artificial Senses”, lecture notes, Università di Pisa, 2019

## 6. Web references

- a) Wikipedia contributors, “Electrocardiography”, Wikipedia, The Free Encyclopedia, 2020, <http://en.wikipedia.org>
- b) Wikipedia contributors, “Cardiovascular disease”, Wikipedia, The Free Encyclopedia, 2020, <http://en.wikipedia.org>
- c) Wikipedia contributors, “Arrhythmia”, Wikipedia, The Free Encyclopedia, 2020, <http://en.wikipedia.org>
- d) Wikipedia contributors, “Atrial fibrillation”, Wikipedia, The Free Encyclopedia, 2020, <http://en.wikipedia.org>
- e) Wikipedia contributors, “Pan–Tompkins algorithm”, Wikipedia, The Free Encyclopedia, 2020, <http://en.wikipedia.org>
- f) Wikipedia contributors, “Stochastic gradient descent”, Wikipedia, The Free Encyclopedia, 2020, <http://en.wikipedia.org>
- g) MIT Learning, “MIT Deep Learning 6.S191”, 2020, <http://introtodeeplearning.com>
- h) Towards Data Science, “Gentle Dive Into Math Behind Convolutional Neural Networks”, 2020, <http://towardsdatascience.com>
- i) Scikit-learn documentation, “Scikit-learn: machine learning in Python”, 2020, <http://scikit-learn.org>