

Multi-kernel Convolutional Neural Network for Wrist Pulse Signal Classification

Xiaofei Chen^{1,2}, Hua Xu²✉, Peng Qian³✉, Yunfeng Xu¹✉, Fufeng Li³, Shengwang Li¹

¹School of Information Science and Engineering,

Hebei University of Science and Technology, Shijiazhuang 050018, China

²State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

³Shanghai University of Traditional Chinese Medicine, Shanghai 201203, China

Abstract—Wrist pulse is one kind of biomedical signals, it is affected not only by the heart beatings, but also by the conditions of nerves, organs, muscles, skin, etc. Therefore, wrist pulse signals can reflect a person's physical state and it has been widely used in health status analysis. However, previous works mainly use traditional machine learning methods to analyze wrist pulse signal. Because wrist pulse signal is high-dimensional and complex, it is difficult for traditional machine learning methods to learn effective information from them. This study aims to explore the utilizing of deep learning methods on wrist pulse signal analysis. We propose a novel multi-kernel Convolutional Neural Network for wrist pulse signal classification. Our model can handle multiple kinds of input features and each of them will pass through a convolutional neural network that has three different sizes of convolution kernel to capture multi-scale information in different time steps. We compare our method with traditional machine learning methods on two tasks: Coronary Atherosclerotic Heart Disease Classification and Traditional Chinese Medicine Constitution yin deficiency and yang deficiency Classification. Besides, we also research the influence of different input features and different channels on wrist pulse signal analysis. The results show that our model significantly improves the performance on the two tasks, which proves the deep learning method is more suitable to deal with complex wrist pulse data.

I. INTRODUCTION

Pulse diagnosis is one of the four major diagnostic methods in Traditional Chinese Medicine (TCM). TCM sphygmopalpation (TCMS) is a combination of human arterial pulse sensing and diagnosis [1]. It plays an indispensable role in the diagnosis of TCM, because it is convenient, low cost and non-invasive. But the wrist pulse signal is easily affected by factors such as the environment and equipment during the collection process. After denoising and baseline drift removal, the information contained in the signal data is still very complex and difficult to obtain. Data-driven deep learning is more effective for processing such complex signal data due to its powerful learning ability, adaptability, and portability. In this paper, based on deep learning methods, we propose a Multi-kernel Convolutional Neural Network (MkCNN) model to classify pulse signal data. We compare our model with traditional machine learning methods in two tasks: Coronary Atherosclerotic

Heart Disease (CAHD) Classification and TCM Constitution yin deficiency and yang deficiency Classification.

The typical computational pulse signal analysis mainly includes four stages: data collection, signal preprocessing, feature extraction and physiology or pathology medical problem classification. There are three types of collector, i.e., pressure, photoelectric, and ultrasonic sensors, which are widely used in data collection; Signal preprocessing generally includes denoise, baseline drift removal and period division; Feature extraction usually extracts time domain features, frequency domain features and wavelet features; Physiology or pathology medical problem classification methods usually use SVM, LDA, etc. The pulse signal can be regarded as approximately periodic change data. Traditional computational pulse signal analysis is often difficult to capture the information of complex pulse signals. Deep learning methods have no additional feature engineering requirements, and it can learn useful information directly from the original signal and get better classification results. In this paper, we focus on feature extraction and physiology or pathology medical problem classification, and propose a pulse signal classification framework based on MkCNN.

The traditional feature extraction methods based on feature engineering: At present, the feature extraction method based on traditional feature engineering mainly obtains features in different domains such as time domain, frequency domain, etc. By using transform-based and non-transformable-based methods. Common transform-based feature extraction methods mainly include Hilbert-Huang Transform (HHT), Approximate Entropy (ApEn), Wavelet Packet Transform (WPT), Wavelet Transform (WT). [2] calculate the extreme point according to the single-period shape feature, and calculate the periodic interval according to the continuous multi-period pulse waveform. Then they introduce the wavelet transform to measure the energy. Finally, they combine the obtained features for subsequent classification. [3], [4] normalize the single-period pulse waveform of each subject, extract the amplitude features, frequency features and principal component features of the pulse signal. Then they sort the different features according to the contribution degree. In recent years, Sample Entropy (SampEn) has been very successful in Electro

✉ xuhua@tsinghua.edu.cn(Hua Xu), qqpp2000439@126.com(Peng Qian), hbkd_xyf@hebust.edu.cn(Yunfeng Xu)

Cardio Graph (ECG) signal analysis and CAHD diagnosis. [5] quantify the complexity of pulse signal by calculating SampEn, and prove that SampEn is also suitable for pulse signal analysis. Common non-transformable feature extraction methods mainly include pulse waveform reference points, autoregressive models and editing distances. [6] train an autoregressive model on the training dataset to calculate the data residuals on the test dataset, and use the mean and variance of the data residuals as features to combine with doppler ultrasound diagnostic parameters for disease classification. [7] use Edit Distance with Real Penalty to calculate the similarity of different pulse waveforms, and classify them according to the similarity. [8] extract six spatial features by locating the reference point of the pulse signal and combine them with Empirical Mode Decomposition (EMD)-based spectral features as the final feature. [9] use a gaussian function to fit a single-period pulse waveform and extract key points in the fitted curve as preliminary extracted features, then they calculate the similarity between features and removes redundant features separately. Finally, they use a statistical differences method to select disease-sensitive features. Based on the Discrete Fourier Series (DFS) curve fitting method, [10] determine the number of DFS items by fitting the residuals of the signal and the original signal, and use the coefficients selected from each item as features. In order to completely extract the pulse features from the entire signal, [11] redefine the wrist pulse is a combination of periodic stationary time series and aperiodic distribution, and decompose the wrist pulse signal into two-dimensional independent components. The above traditional feature extraction methods are mostly based on single-period pulse signal extraction, which ignores information between periods. Due to the patient's irregular pulse rate and other physiological factors. The change between periods can also reflect a person's physical health, which plays a non-negligible role in pulse diagnosis.

The feature representation methods based on deep learning: The physiological signal data is a kind of unstructured data, which has inconsistency in length, and the discrete spectral lines are not distributed at equal intervals. Deep learning methods can directly obtain high-level feature representations containing changes within and between periods from this data. Based on a multi-path deep neural network, [12] extract local and global feature representation of signal data. Based on a bi-directional long short-term memory network, [13] extract time series feature representation of different lengths. Based on a bi-gated recurrent unit network, [14] extract the hierarchical feature representation of signal data at a different level for emotion classification. Pulse data is a kind of quasi-periodic physiological signal data. But, deep learning methods are not widely used in solving pulse signal classification problems at present. Generally, deep learning methods can effectively obtain more abundant representation when processing complex unstructured data.

The classification of physiological or pathological medical problems: Traditional machine learning classification methods need the help of feature engineering, while deep

learning methods can automatically extract features on prediction. [2], [3], [5] use LDA and SVMs methods to classify pulse signals. [15] apply multiple kernels learning on classifying. [7] classify basing on Edit Distance with Real Penalty (ERP) with the Difference Weighted KNN (DFWKNN) classifier. Due to pulse have characteristics like the appearance (deep or floating), rate (rapid or slow), intensity (forceful or forceless), rhythm (tidy or not, whether there is a pause) and morphology, it is an important pattern identification in TCM. The traditional machine learning classifier is difficult to implement in large-scale samples due to the limit of feature engineering, and it is sensitive to the selection of parameters and kernel functions. Deep learning technology has been widely used in ECG signals which are similar to pulse signals. [16] introduce the DNN to the arrhythmia classification experiment. The results show that the DNN achieves the best accuracy, sensitivity and specificity. [17] design 1 Dimension-CNN for arrhythmia classification, and it achieves an average accuracy of 91.33% in the classification experiments of 17 arrhythmia diseases. The success of deep learning on ECG signals also brings many possibilities for pulse signal analysis. Deep learning methods can effectively obtain complex and indistinguishable pulse information, and improve the accuracy of pulse signal analysis in consequence.

Traditional methods are ineffective in capturing the local and global period information, and deep learning methods trend to building model without considering TCM theory. To solve these problems, we propose the MkCNN which can effectively learn the local and global information of pulse signal and integrate TCM theory into extracted features. These efforts make us achieve better results in the pulse signal classification task. Detailed contents will be introduced in the fourth section.

We summarize our contributions as follows:

- We innovatively propose a multi-kernel convolutional network for pulse signal analysis basing on deep learning knowledge, which has achieved significantly improved on the Coronary Atherosclerotic Heart Disease Classification and Traditional Chinese Medicine Constitution yin deficiency and yang deficiency Classification. The experiment results prove that our model has good robustness and portability.
- We compare the classification performance of different features (features extracted by feature engineering and preprocessing), different classification models (SVM, LDA, XGBoost, MkCNN) and different pulse channels (pulse signals collected from cun, guan, chi on left and right hand according to TCM theory), which make our work very useful for reference.
- Pulse diagnosis is one of the main diagnostic methods in the clinical diagnosis of TCM. Our model achieves the accuracy of 91.9% in CAHD classification task and 78.3% in TCM constitution yin deficiency and yang deficiency classification task. The results indicate MkCNN has important practical significance in modern TCM auxiliary diagnosis decision.

The rest of the paper has been divided into five basic sections. "Section II" introduces our specific reference work on feature engineering and deep learning methods, "Section III" introduces the acquisition and preprocessing process of pulse signal data, and "Section IV" describes feature extraction methods and physiology and pathology medicine. The classification method of the problem, "Section V" introduces the experimental settings and experimental results, and analyzes the experimental results meticulously. Finally, "Section VI" concludes the proposed work and gives some hints on its scope in the near future.

II. RELATED WORK

In this section, we mainly introduce the approaches most relevant to this paper, including the feature engineering method and Deep learning.

A. Feature Engineering

It is a common practice to use feature engineering to extract features from pulse signal data. [8] propose a spatial feature extraction method for blood flow velocity signal. This method calculates the time and amplitude feature of the signal key points, and extracts each period signal spatial features by calculating the ratio. Compared with the traditional pulse signal period features in [11], this method is more intuitive and it is difficult to locate key points. [3] use the PCA method to compress data, and obtain good results. The blood flow velocity signal of the wrist is highly correlated with the pulse signal, and the pulse signal data is quasi-periodic, and the periods are highly similar. Inspired by this, we use the PCA method to compress the pulse signal data and remove redundant features while extracting the spatial features.

B. Deep Learning

In recent years, deep learning algorithms have been proved to have great potential in the field of medical image. [18] propose a residual neural network for the cochlear detection model. The model is composed of three residual CNNs and is trained on three images of different sizes. The experimental results show that the network can accurately detect the cochlea. [19] propose a single-view 2D CNNs for lung nodule detection. Unlike previous 3D CNN-based frameworks, this model uses single-view 2D CNNs to improve computational efficiency, which achieves the most advanced performance while reducing the complexity of the calculation. [20] propose a CNN network model, which uses a multi-channel architecture to ensure that the learned vectors do not deviate too far from the original value to prevent overfitting. Inspired by the simple CNN with little hyperparameter tuning and static vectors [20], we use the features extracted by existing feature engineering methods. Through fine-tuning to learn task-specific vectors and further gains in performance. Considering that the pulse signals are multi-channel time series data, we combine the features of the pulse signal data to construct a multi-kernel CNN network structure. By setting different size convolution kernels, the model learns more comprehensive information from the pulse signals.

III. PULSE SIGNAL COLLECTION AND PRE-PROCESSING

In this section, we mainly introduce the collection device information, data collection process, and data preprocessing method.

A. Data Collection

We use the ZM-IIIIC intelligent pulse signal collector, which is composed of MH-IIA single-probe pulse transducer, ZM-IIIIC pulse collector (including A/D converter) and serial communication cable. The transducer can be placed in cun, guan or chi according to the need to detect pulse information. ZM-IIIIC intelligent pulse signal collector has two output modes: digital signal and analog signal. The analog signal can be connected to the printer to print out various pulse waveform graph, and can also be connected to the display instrument or other recording instruments; the digital signal is connected to the user computer. In addition, it can save the collected signal into files which is convenient for further research. In this work, we use the device to collect six channels of pulse signals of cun, guan, chi on the left and right hand respectively. We show these data collected positions in Fig. 1, and collect 40 seconds of data in each channel with the sampling rate is 1000Hz.

B. Pulse Signal Preprocessing

Because the sampleable available area of the pulse signal on the wrist is very small, the inaccurate placement of the sampling device will cause serious distortion of the data, which are shown in Fig. 2. Hence, we select the data manually after sampling to ensure that the data used is valid.

The data after manual selecting still has noise and drift. Therefore, we further process the data to solve these problems. Because removing data drift needs to extract the key points of the data, and the noise will affect the detecting of key points. We use the wavelet transform to denoise the signal in the first step. Then, we use the cubic spline method to fit the drift curve, thus obtain the drift removal signal. The details are explained in the following subsections.

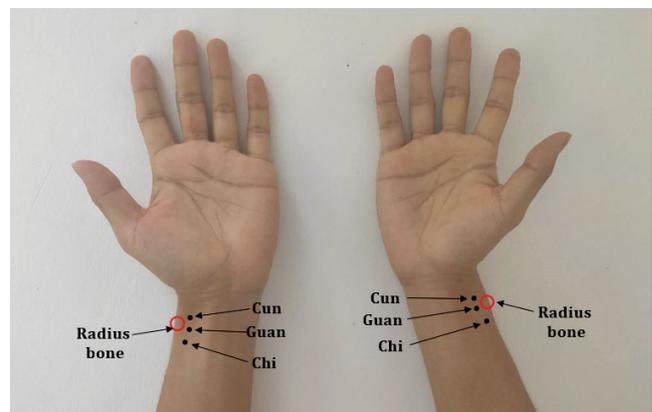


Fig. 1. Position of cun, guan, chi

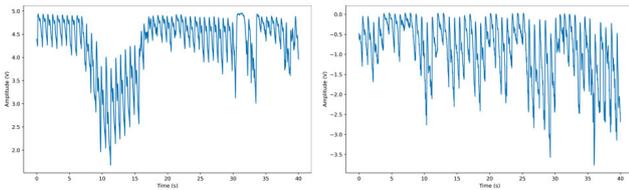


Fig. 2. Distorted waveform samples

1) *Pulse Denoising*: The causes of pulse signal noise are power frequency signal interference, vibrates of the sampling position, etc. The frequency of these noises is much greater than the pulse signal (the frequency of the effective data in the pulse signal is lower than 20Hz [21]). So the noise can be removed according to the characteristics in the data.

The traditional low-pass filter can effectively remove the interference of the power frequency signal, but due to the error between the actual low-pass filter and the ideal low-pass filter. The wavelet transform scheme to remove noise does not have such a problem. Therefore, as shown in Fig. 3, we use discrete Meyer wavelet transform to decompose the data. After the original signal is decomposed once, cD1 (detail coefficients, high frequency) and cA1 (approximation coefficients, low frequency) are obtained, and cA1 is further decomposed to obtain cA2, cD2. Similarly, cA3 and cD3 are obtained. The noise in the pulse signal exists in cD1, cD2 and cD3 [4], these three components are set to zero, and the reconstructed signal is noiseless. The visualization of this method to remove noise is shown in Fig.4.

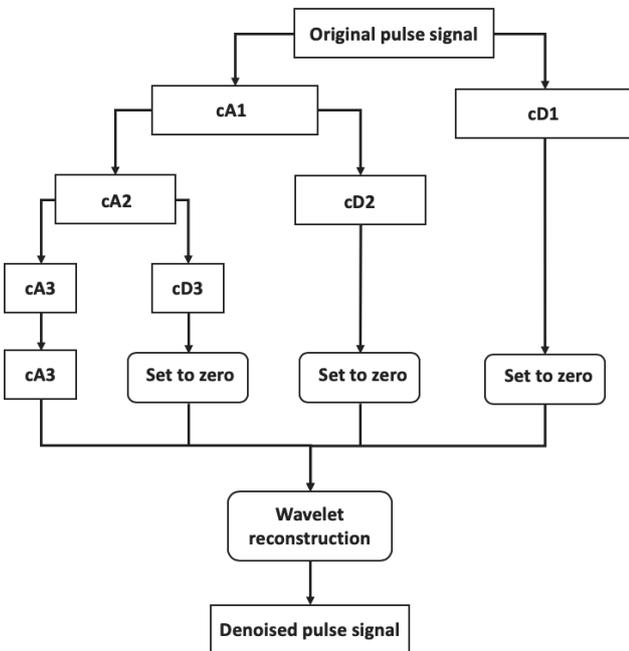


Fig. 3. Wavelet transform based pulse signal denoising

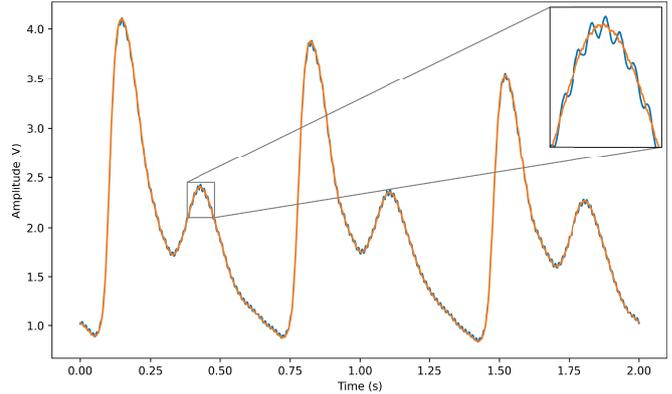


Fig. 4. The visualization of removing noise. Blue line is the Original Pulse Signal and orange line is the Denoised Pulse Signal

2) *Baseline Drift Removal*: The collector applies inconsistent pressure will cause the baseline drift. In order to perform accurate pulse signal analysis, this interference needs to be eliminated. Based on the Previous works [9], [22], and [23], we propose a drift-removal scheme that combines the data frequency and length characteristics. The flow diagram is shown in Fig. 5). Firstly, we detect the peak points and valley points of the data after noise denoising (as shown in B and C of Fig. 5), the detail steps are as follows:

- Step 1: Perform discrete fourier transform on the sampled data to find the fundamental frequency f of the pulse signal. The period length of T of a single pulse signal can be estimated by $T = 1/f$. According to T , the total number of periods $N = t/T$ within the sampling time t can be estimated.
- Step 2: Find the peak point in $[0, T]$ sampled data, marked as P_1 , the corresponding time is t_1 , P_1 is the first peak point of all sampled data.
- Step 3: Taking P_1 as the starting point, the next peak point is detected within the range of $[t_1 + 0.5 \times T, t_1 + 1.5 \times T]$, marked as P_2 , and the corresponding time is t_2 .
- Step 4: Repeat the previous step until all the peak points in the sampled data are detected.
- Step 5: The minimum point between two adjacent peak points is the valley point.

Then we set the valley point as the starting point of each pulse period. Performing cubic spline interpolation on these starting points (as indicated by D in Fig. 5) to obtain an estimate of baseline drift, then subtract this estimate. The overall waveform moves to a unified baseline (as indicated by E in Fig. 5).

Finally, the drift removal data is obtained, and the visualization is shown as F in Fig. 5. Comparing with the methods in [9] and [23], our scheme uses an improved onsets detection method to solve the case of missing detection at the start and end of the valley point. It achieves similar performance to the adaptive cascade filter with less calculation and more robust.

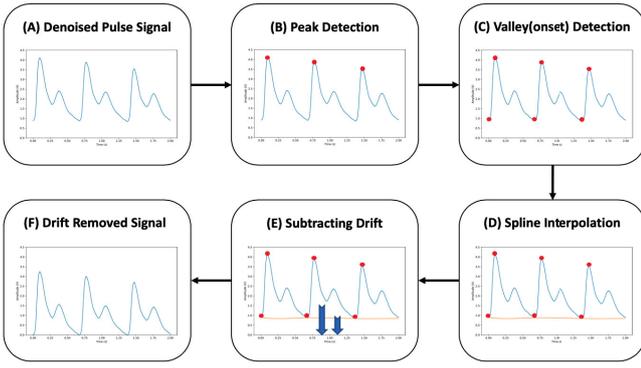


Fig. 5. Remove the drift process flow diagram. The arrow in the figure E indicates the baseline drift of the waveform image

IV. METHOD

In this section, we mainly introduce feature engineering methods and Multi-kernel Convolutional Neural Networks (MkCNN). The overall framework of the method is shown in Fig. 6. The framework mainly includes three modules. The Data Preprocessing and Feature Engineering module provide the input feature for MkCNN with Single-feature module and MkCNN with Multi-feature module. Among them, Data Preprocessing has been introduced in section III-B. The rest modules will be expanded in this section.

A. Feature Engineering

In this section, we divide the single period of the pulse signal, and then extract the spatial features and principal component analysis features.

1) *period division*: In section III-B2, we use the proposed onsets detection method to locate the starting point of all periods accurately, and then we can divide the data into single periods. In order to calculate the average value of the split single-period signals, we pad all the single-period signals to the same length. After zero-padding operation [10], the all single-period waveforms are shown in Fig. 7(a). We divide the summation of data by the number of periods to get the average period as shown in Fig. 7(b); We divide the summation of data by the maximum value of periods to get the normalized period as shown in Fig. 7(c). The single-channel pulse data can be divided into several periods $p_i, i = 1, \dots, n$, and the summed period data is P .

$$P = \sum_{i=1}^n p_i \quad (1)$$

The formula for calculating the average period \bar{p} and normalized period \hat{p} of single-channel data is as follows:

$$\begin{aligned} \bar{p} &= \frac{1}{n} P \\ \hat{p} &= \frac{\bar{p}}{\max(P)} \end{aligned} \quad (2)$$

TABLE I. FIDUCIAL POINTS OF BLOOD VELOCITY SIGNAL

Points	Feature Meaning
a	Onset of one period
b	Peak point of primary wave
c	Dicrotic notch
d	Peak point of secondary wave
a'	Onset of the next period

2) *Spatial Feature*: Spatial features can show the amplitude and duration of key points in a single periodic signal [8]. Taking the average period as an example, the key points in a waveform is shown in Fig. 8. There are five key points. Table I describes the meaning of each key point.

After the single-period signal is divided, the key point detection method is as follows:

- Step 1: Using average period signal data to extract features, the starting (valley) point of the data is point a , the time is t_a , the peak point is point b , the corresponding time is t_b , the amplitude is h_b , and the total length of the data is T .
- Step 2: Search for the peak of the secondary wave (*step* is the empirical value) between $[t_b + 0.5 \times \text{step}, t_b + 1.5 \times \text{step}]$, and record it as point d , the corresponding time is t_d , the amplitude is h_d . Then searching for the minimum value between the points b and d , and record it as point c . The corresponding time is t_c and the amplitude is h_c .
- Step 3: The formula for calculating the features of the average period signal is as follows:

$$\begin{aligned} \frac{T_{ba}}{T} &= \frac{t_b - t_a}{T} \\ \frac{T_{cb}}{T} &= \frac{t_c - t_b}{T} \\ \frac{T_{dc}}{T} &= \frac{t_d - t_c}{T} \\ \frac{T_{a'b}}{T_{ba}} &= \frac{T - t_b}{t_b - t_a} \end{aligned} \quad (3)$$

3) *Principal Component Analysis Feature*: To remove redundant data from the pulse data as much as possible, we performed principal component analysis (PCA) for each channel separately. The purpose of PCA is to find out the data the orthogonal direction of strong variability. Given the single-channel pulse data x_i of d dimension, where $i = 0, 1, \dots, n$, the orthogonal projection calculation method is as follows:

$$y_i = A^T (x_i - \mu) \quad (4)$$

Where y_i is the converted data, and μ is the average value of the pulse signal data. The covariance matrix M of pulse data is defined as:

$$M = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \quad (5)$$

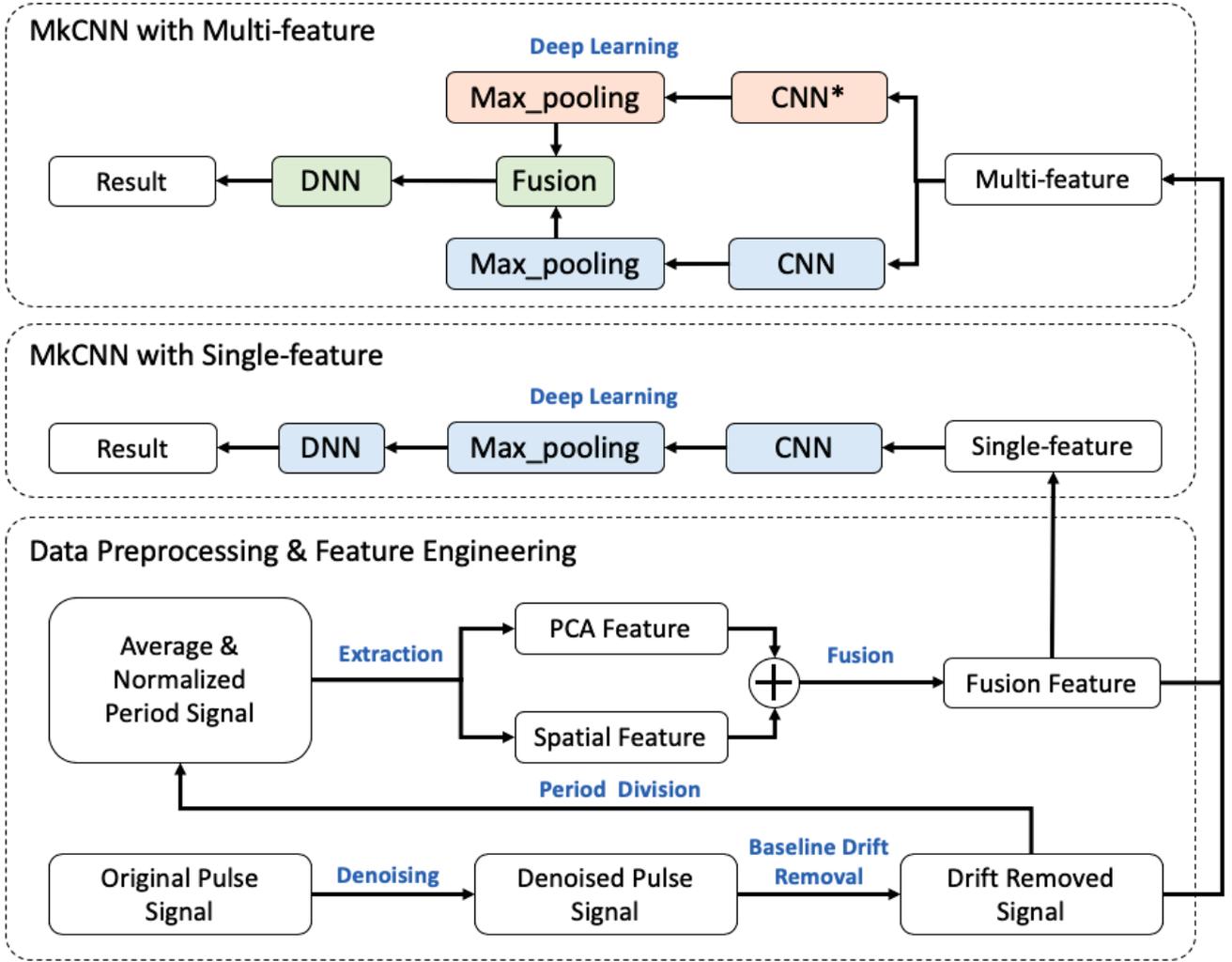


Fig. 6. The overall framework of wrist pulse signal classification. CNN* and CNN have different convolution kernel sizes

After the covariance matrix M is obtained, the eigenvalue A of the matrix M can be obtained by eigenvalue decomposition:

$$M = AVA^T \quad (6)$$

In order to achieve optimal linear dimensionality reduction, the p principal component of the least square reconstruction error is retained, and the reconstruction error is located as:

$$E_i = |x - \mu - yA_p|^2 \quad (7)$$

A_p represents the mapping from input data to ideal space [24]. Finally, we reduce the data dimension from 624 to 20 dimensions, and use it as the PCA feature for subsequent classification experiment.

B. Deep Learning Model

The overall structure diagram of our proposed Multi-kernel Convolutional Neural Network (MkCNN) is shown in Fig. 9. Our method has two variants of pulse data with a different

number of features. MkCNN with single-feature (MkCNN-sf) and MkCNN with multi-feature fusion (MkCNN-mf).

1) *MkCNN-sf*: *MkCNN with single-feature*: The model structure of MkCNN-sf takes the lower part of Fig. 9 as an example. The first is the feature fusion of the input level. Spatial feature D_{spa} and PCA feature D_{pca} will be concatenated to obtain the fusion feature D_{FPS} .

$$D_{FPS} = \text{Concat} ([D_{spa}, D_{pca}]) \quad (8)$$

We use the fusion feature as a single feature in the Input layer. Using the multi-kernel convolutional neural network to learn the high-order feature d_{FPS} from D_{FPS} .

In Fig. 9, the different colored dotted boxes represent the convolutional kernel of different sizes, which is used to capture pulse data information at different time steps. First, we set three different sizes of convolution kernels (f_1, f_2, f_3). It's 1, 5, 10. Perform convolution and pooling operations on D_{FPS} . After each convolution, use the Rectified Linear Units

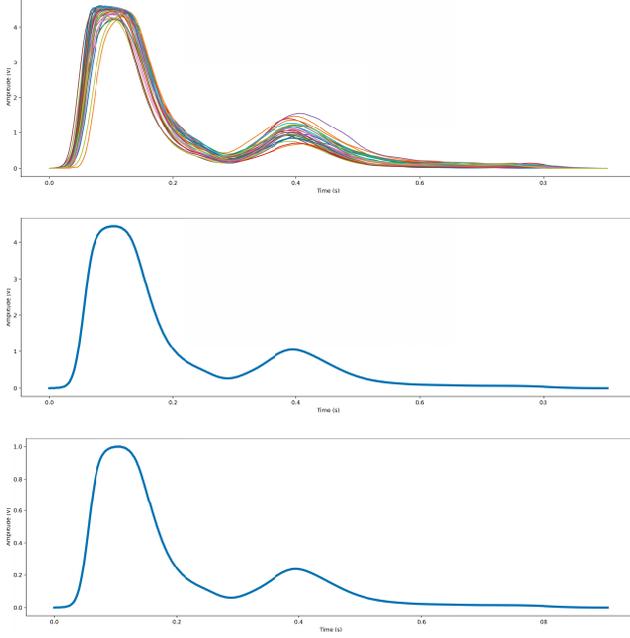


Fig. 7. Period division.(a) denotes All-periods; (b) denotes Average Period; (c) denotes Normalized Period

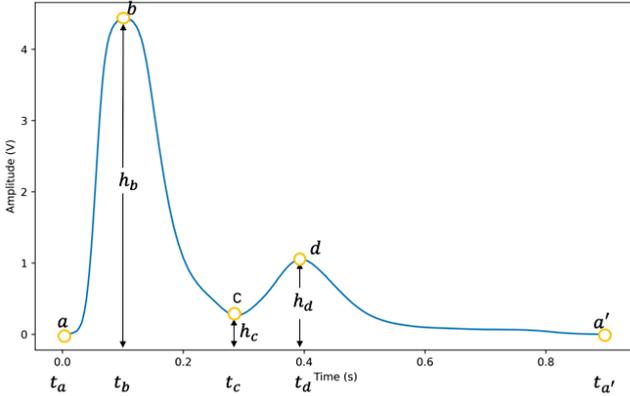


Fig. 8. Key points in single period waveform

(ReLU) activation function to keep the gradient from decay. The specific formula is as follows:

$$\begin{aligned} d_{FPS}^1 &= \text{Maxpool}(\text{ReLU}(\text{Conv 1D}(D_{FPS}, f_1))) \\ d_{FPS}^2 &= \text{Maxpool}(\text{ReLU}(\text{Conv 1D}(D_{FPS}, f_2))) \\ d_{FPS}^3 &= \text{Maxpool}(\text{ReLU}(\text{Conv 1D}(D_{FPS}, f_3))) \end{aligned} \quad (9)$$

Then, the high-order features d_{FPS}^1 , d_{FPS}^2 and d_{FPS}^3 convolved by three different convolution kernels are concatenated. The specific formula is as follows:

$$d_{FPS} = \text{Concat}([d_{FPS}^1, d_{FPS}^2, d_{FPS}^3]) \quad (10)$$

Then we through a fully connected network to learn its final predicted value P_{FPS} , the specific formula is as follows,

where W_{FPS} and b_{FPS} are the weights and biases of the single-feature D_{FPS} fully connected layer, respectively.

$$P_{DRS} = W_{FPS} \times d_{FPS} + b_{FPS} \quad (11)$$

2) *MkCNN-mf*: *MkCNN with multi-feature fusion*: The model structure of MkCNN-mf is shown as the whole in Fig. 9. We performed model level fusion for different features. The size of three convolution kernels (g_1, g_2, g_3) is set to 10, 100, 1000, respectively. Concatenating after convolution and pooling operations. As follows:

$$\begin{aligned} d_{DRS}^1 &= \text{Maxpool}(\text{ReLU}(\text{Conv 1D}(D_{DRS}, g_1))) \\ d_{DRS}^2 &= \text{Maxpool}(\text{ReLU}(\text{Conv 1D}(D_{DRS}, g_2))) \\ d_{DRS}^3 &= \text{Maxpool}(\text{ReLU}(\text{Conv 1D}(D_{DRS}, g_3))) \\ d_{fps} &= \text{Concat}([d_{fps}^1, d_{fps}^2, d_{fps}^3]) \end{aligned} \quad (12)$$

Then we perform the model layer fusion of the high-order features d_{DRS} and d_{FPS} .

Finally, we use the fully connected layer for classification prediction. Meanwhile, W_f and b_f are the weights and offsets of fully connected layers using multi-input features D_{DRS} and D_{FPS} , respectively.

$$P = W_f \times \text{Concat}([d_{DRS}, d_{FPS}]) + b_f \quad (13)$$

3) *Model Training*: We use the L1-Loss as the objective function. Besides, we create a criterion that measures the Mean Absolute Error (MAE) between each element in the input x and target y . x and y are tensors of arbitrary shapes with a total of n elements each. The unreduced loss can be described as:

$$\begin{aligned} l(x, y) = L &= \text{mean}(\{l_1, \dots, l_N\}^1) \\ l_n &= |x_n - y_n| \end{aligned} \quad (14)$$

Where N is the batch size. The sum operation still operates over all the elements, and divides by n .

TABLE II. DATASET STATISTICS

Dataset	CAHD-D		TCMC-D	
	CAHD	Healty	yin	yang
Train	77	86	53	47
Test	27	27	19	21
All	104	113	72	68

TABLE III. INPUT FEATURES DETAIL

Input Feature	Abbreviation	Data length
Original Pulse Signal	OPS	40,000
Denoised Pulse Signal	DPS	40,000
Drift Removed Signal	DRS	40,000
Average Period Signal	APS	624
Normalized Period Signal	NSS	624
PCA Feature	PF	20
Spatial Feature	SF	6
Fusion Feature	FF	26

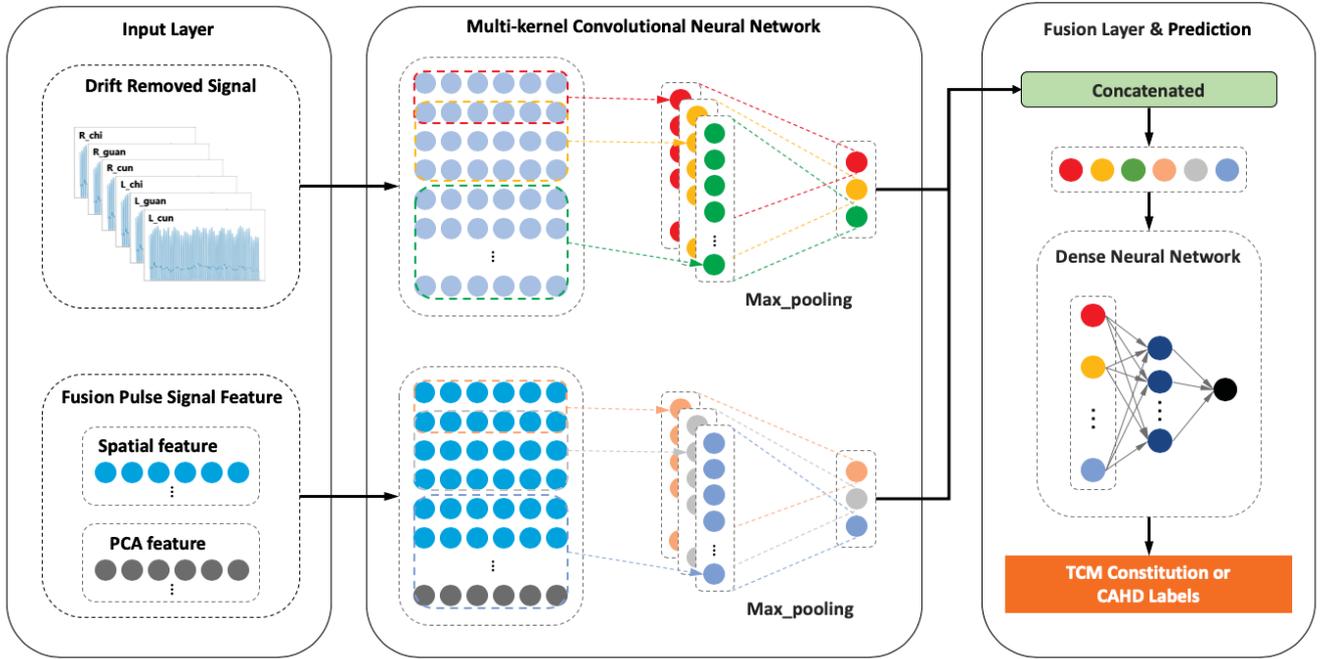


Fig. 9. Multi-kernel Convolutional Neural Network. Using *L* and *R* to represent the left and right hands, for example, *L – cun* represents the cun position pulse data of the left wrist, and *R – guan* represents the guan position pulse data of the right wrist

V. EXPERIMENTS

In this section, we introduce the experiment setting, and we show sufficient experimental results and analyze the experimental results in detail. The results demonstrate the effectiveness of our work.

A. Experiment Setting

In this section, we present the dataset, the important hyperparameter settings during the deep learning model training process, evaluation metrics, and the traditional machine learning models we compared.

1) *Datasets*: Using the volunteer’s pulse signal data collected from different regions of China, we construct two datasets. One is the Coronary Atherosclerotic Heart Disease Pulse Dataset (CAHD-D), which collect a single-channel signal at the guan position of the left hand. The other is the TCM Constitution yin deficiency and yang deficiency Pulse Dataset (TCMC-D) which collect on the six positions of cun, guan, chi on the left and right hand respectively. TCM experts label the constitution dataset in combination with TCM standards. The statistical information of the two datasets is shown in Table II:

2) *Training Details and Parameters Setting*: We employ the PyTorch framework to implement the proposed MkCNN-sf and MkCNN-mf. We set three different sizes of convolution kernels for different feature inputs. After model level fusion, our set dropout rate to 0.5. The initial learning rate is set to 0.01. When the accuracy on the test set has not improved for fifty consecutive epochs, the training is terminated. All of our results are the average of 5 round experiments.

3) *Evaluation Metrics*: We use five cross-validation as the final experimental evaluation result. We use the four evaluation metrics: Accuracy, F1-score, Sensitivity, and Specificity to evaluate the performance of our method. Accuracy shows the ratio of the number of samples of the model prediction pair to the total number of samples.

According to the Fig. 10. The four evaluation metrics are defined as follows:

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \\
 \text{F1-score} &= \frac{2TP}{2TP+FP+FN} \\
 \text{Sensitivity} &= \frac{TP}{TP+FN} \\
 \text{Specificity} &= \frac{TN}{FP+TN}
 \end{aligned}
 \tag{15}$$

Confusion Matrix		Predictive Value	
		Positive (yin/Healthy)	Negative (yang/CAHD)
Actual Value	Positive (yin/Healthy)	TP	FN
	Negative (yang/CAHD)	FP	TN

Fig. 10. Confusion Matrix

4) *Compared Method: Linear Discriminant Analysis (LDA)*: [25] LDA is a linear classification method. Taking binary classification as example, given dataset $D = \{(x_i, y_i)\}_{i=1}^m$, $y_i \in \{0, 1\}$, the mean vector and covariance matrix of samples $i \in 0, 1$ type are represented by μ_i, ω_i , respectively. The distance between the two categories after the projection can be expressed as the distance between the center points of the category $\|\omega^T \mu_0 - \omega^T \mu_1\|_2^2$, the distance between the sample points within the two categories is $\omega^T \Sigma_0 \omega + \omega^T \Sigma_1 \omega$. In order to maximize the distance between categories and minimize the distance of sample points within the category after projection, the optimal projection direction can be obtained by solving this problem. The optimization function of LDA is:

$$\max_{\omega} \frac{\|\omega^T \mu_0 - \omega^T \mu_1\|_2^2}{\omega^T \Sigma_0 \omega + \omega^T \Sigma_1 \omega} \quad (16)$$

Linear Support Vector Machine (SVM-linear): [26] SVM-linear is a linear classifier defined in the feature space. The learning strategy is to maximize the interval of the categories. The optimization of linear separable SVM can be expressed as the following convex quadratic problem:

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i (w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N \end{aligned} \quad (17)$$

ω is the normal vector on the hyperplane where the sample points (x_i, y_i) , $i = 1, 2, \dots, N$ locate, and b is the intercept. Given a linearly separable train set, the separation hyperplane obtained by solving the above convex quadratic problem:

$$\omega^* x + b^* = 0 \quad (18)$$

The corresponding classification decision function as follows:

$$f(x) = \text{sign}(\omega^* x + b^*) \quad (19)$$

Support Vector Machine with Radial Basis Function kernel (SVM-rbf): The basic model SVM is a linear classifier, but the introduction of kernel trick makes it a substantially nonlinear classifier. Commonly used kernel functions are radial basis function, polynomial kernel function, etc. The expression of the radial basis function is as follows:

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right) \quad (20)$$

After using this kernel function, the classification decision function of SVM is as follows:

$$f(x) = \text{sign}\left(\sum_{i=1}^{N_s} a_i^* y_i \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) + b^*\right) \quad (21)$$

σ is a free parameter and a^* , b^* are the parameters learned after model training.

Extreme Gradient Boosting (XGBoost) XGBoost is an improved algorithm of Gradient Boosting Decision Tree (GBDT), and is the integration of several Classification And

Regression Tree (CART). XGBoost's objective function is as follows:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (22)$$

Compared with GBDT, the objective function of XGBoost has more regular items, making the learning model more difficult to overfit. The regular term function is as follows:

γ and λ are coefficient terms.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \quad (23)$$

T is the number of leaf nodes used to control the complexity of the tree. The weights ω is the leaf nodes of each tree, and the other part is the quadratic sum of ω . γ and λ are coefficient terms.

B. Experimental Results

In this section, we conduct classification experiments from three perspectives (input features, classification models, and pulse signal collect positions.) on two datasets.

To explore the performance of different input features, we utilize eight kinds of features, among which the fusion feature means concatenated the PCA feature and Spatial feature. The details are shown in Table III.

1) *Compare Different Input Features*: On the CAHD dataset, we use MkCNN model to compare the performance on 8 different single-features and a multi-feature which use Drift Removed Signal and Fusion Feature as input features simultaneously. The experimental results is shown in Table 4.

Because of deep learning can automatically extract features from the original data, we explored the performance of single features in the preprocessing stage. The Drift Removed Signal after preprocessing achieved the best performance. This shows that preprocessing is very necessary.

At the stage of feature extraction, the Fusion Feature performance is better than PCA Feature or Spatial Feature. This shows that it is meaningful to combine the different types of features. Table V shows the parameters and the average time of 5 epochs during training MkCNN.

So, we choose the Drift Removed Signal and Fusion Feature as the input features of MkCNN-mf. As we expected, our proposed MkCNN-mf achieved the best results on the three evaluation metrics. Compared to the Fusion Feature, the multi-feature accuracy rate has been improved by 5.1%, F1 and sensitivity have achieved huge improvements of 15.58% and 19.62% respectively. The classification results using the multi-feature have achieved a more balanced effect from the confusion matrix (as shown in Fig. 11).

2) *Compare Different Classification Models*: Based on the above experiments, we obtain the pulse signal data with the best performance in different preprocessing stages. Therefore, we use Drift Removal Signal, Fusion Feature as input features, and compare the effectiveness of the five models (XGBoost, SVM-linear, SVM-bf, LDA and MkCNN) on CAHD dataset. The experimental results are shown in Table VI:

Our MkCNN model achieves the best performance in all evaluation metrics, and it improves 8.07% in accuracy and

TABLE IV. THE COMPARISON EXPERIMENT RESULTS OF DIFFERENT INPUT FEATURES

Input Feature	Model	ACC	CAHD-D		
			F1	Sensitivity	Specificity
OPS	MkCNN-sf	0.8705	0.8154	0.7751	0.9601
DPS	MkCNN-sf	0.8751	0.8529	0.8124	0.9410
DRS	MkCNN-sf	0.8776	0.7660	0.7393	0.9767
APS	MkCNN-sf	0.8395	0.7521	0.6780	0.9733
NPS	MkCNN-sf	0.8656	0.8184	0.7592	0.9397
PF	MkCNN-sf	0.8656	0.8201	0.7486	0.9509
SF	MkCNN-sf	0.5951	0.4174	0.5310	0.5455
FF	MkCNN-sf	0.8684	0.7606	0.6920	1
DRS + FF	MkCNN-mf	0.9194	0.9164	0.8882	0.9398

TABLE V. PARAMETERS AND AVERAGE TIME OF EACH EPOCH BY USING FUSION FEATURE AND NORMALIZED PERIOD SIGNAL

Input feature	Parameter	Average time(ms)
Fusion Feature	24051	6.14
Normalized Period Signal	43751	6.90

TABLE VI. THE COMPARISON EXPERIMENT RESULTS OF DIFFERENT CLASSIFICATION MODELS

Model	Input Feature	CAHD-D			
		Accuracy	F1-score	Sensitivity	Specificity
SVM-rbf	DRS	0.7317	0.5934	0.5568	0.8839
SVM-rbf	FF	0.6327	0.5131	0.5117	0.7000
SVM-rbf	DRS + FF	0.7363	0.5967	0.5568	0.8916
LDA	DRS	0.4427	0.3453	0.3174	0.5515
LDA	FF	0.7632	0.7325	0.7137	0.7955
LDA	DRS + FF	0.5094	0.4795	0.4826	0.5363
SVM-linear	DRS	0.4898	0.4434	0.4421	0.5582
SVM-linear	FF	0.8387	0.7907	0.7261	0.9285
SVM-linear	DRS + FF	0.6212	0.5882	0.5850	0.6562
XGBoost	DRS	0.5248	0.4616	0.4302	0.6051
XGBoost	FF	0.7649	0.7901	0.8530	0.6883
XGBoost	DRS + FF	0.7454	0.7720	0.8530	0.6519
MkCNN-sf	DRS	0.8612	0.7710	0.7222	0.9813
MkCNN-sf	FF	0.8684	0.7606	0.692	1
MkCNN-mf	DRS + FF	0.9194	0.9164	0.8882	0.9398

12.57% in F1-score compared with the best results of the four traditional machine learning methods. From Fig. 12, we can see that MkCNN has achieved the best results in accuracy and specificity on the three types of inputs.

3) *Compare Different Pulse Channels*: Based on the experiments in the previous sections, we directly select the Drift Removed Signal and Fusion Feature multi-features as input features. Then we conduct different pulse position experiments on the TCMC dataset. We regard pulse data collected at different positions as different channels. Among them, L-cun, L-guan, L-chi, R-cun, R-guan, R-chi correspond to channel 0#, 1#, 2#, 3#, 4#, 5#. We not only use the single channel, but also using the combined channels. The experimental results are shown in Table VII.

In the experiment of 63 different channel combinations, we ranked the experimental results according to the performance of the accuracy. The combination with the highest accuracy

TABLE VII. THE COMPARISON EXPERIMENT RESULTS OF THE TCMC CLASSIFICATION TASK CHANNEL COMBINATION

Channel	TCMC-D			
	Accuracy	F1-score	Sensitivity	Specificity
3#	0.5333	0.5781	0.7460	0.2982
1#, 3#	0.5500	0.6143	0.7302	0.3509
1#, 2#, 3#	0.5583	0.7014	0.9841	0.0877
1#, 3#, 5#	0.5667	0.4514	0.3492	0.8070
1#, 3#, 4#	0.5667	0.6548	0.8095	0.2982
0#	0.5667	0.6744	0.8571	0.2456
0#, 3#	0.5667	0.6402	0.7619	0.3509
1#, 2#, 3#, 5#	0.5750	0.5649	0.5397	0.6140
5#	0.5750	0.4941	0.4444	0.7193
0#, 4#	0.5750	0.5324	0.5397	0.6140
2#, 4#	0.5750	0.6688	0.8254	0.2982
4#	0.5833	0.6477	0.7937	0.3509
1#, 2#, 4#, 5#	0.5833	0.6247	0.6984	0.4561
3#, 4#	0.5833	0.5452	0.4921	0.6842
1#, 2#, 5#	0.5833	0.4523	0.3333	0.8596
1#, 2#	0.5833	0.6192	0.6667	0.4912
1#, 2#, 4#	0.5833	0.4975	0.4762	0.7018
1#, 4#	0.5833	0.5509	0.5079	0.6667
0#, 1#, 3#	0.5917	0.5229	0.4603	0.7368
0#, 1#, 2#, 4#	0.5917	0.6874	0.8730	0.2807
2#, 3#	0.5917	0.6251	0.6508	0.5263
2#, 5#	0.5917	0.6635	0.7937	0.3684
0#, 1#, 3#, 4#	0.5917	0.6236	0.6825	0.4912
0#, 1#, 2#, 3#	0.6000	0.5347	0.5397	0.6667
1#, 2#, 3#, 4#	0.6000	0.6261	0.6667	0.5263
3#, 5#	0.6083	0.5505	0.5079	0.7193
2#	0.6083	0.5017	0.3810	0.8596
2#, 3#, 5#	0.6083	0.6422	0.6825	0.5263
3#, 4#, 5#	0.6083	0.5337	0.4286	0.8070
2#, 3#, 4#, 5#	0.6167	0.4885	0.3810	0.8772
0#, 2#, 3#	0.6167	0.5841	0.5397	0.7018
0#, 1#, 2#, 3#, 5#	0.6167	0.5665	0.5079	0.7368
1#, 2#, 3#, 4#, 5#	0.6167	0.5635	0.5079	0.7368
1#	0.6167	0.6675	0.7778	0.4386
0#, 2#	0.6250	0.5423	0.4286	0.8421
2#, 4#, 5#	0.6250	0.4816	0.3333	0.9474
0#, 1#, 2#	0.6250	0.5186	0.3968	0.8772
1#, 3#, 4#, 5#	0.6250	0.6022	0.5397	0.7193
0#, 1#, 2#, 3#, 4#	0.6250	0.6763	0.7460	0.4912
2#, 3#, 4#	0.6333	0.6230	0.6190	0.6491
0#, 2#, 3#, 4#	0.6333	0.6659	0.6984	0.5614
0#, 3#, 4#	0.6333	0.6585	0.6984	0.5614
0#, 2#, 4#	0.6417	0.6421	0.6508	0.6316
0#, 1#	0.6417	0.5826	0.4762	0.8246
0#, 5#	0.6417	0.6740	0.7143	0.5614
0#, 1#, 4#	0.6417	0.6227	0.5873	0.7018
0#, 3#, 5#	0.6417	0.6879	0.7619	0.5088
1#, 5#	0.6500	0.6663	0.6667	0.6316
0#, 1#, 5#	0.6583	0.6426	0.5873	0.7368
1#, 4#, 5#	0.6667	0.6587	0.6190	0.7193
0#, 2#, 3#, 5#	0.6667	0.6760	0.6667	0.6667
4#, 5#	0.6750	0.6051	0.4762	0.8947
0#, 1#, 3#, 5#	0.6833	0.6895	0.6984	0.6667
0#, 1#, 2#, 5#	0.6917	0.6716	0.6190	0.7719
0#, 2#, 5#	0.7000	0.7028	0.6825	0.7193
0#, 3#, 4#, 5#	0.7000	0.7444	0.8413	0.5439
0#, 2#, 3#, 4#, 5#	0.7083	0.7482	0.8254	0.5789
0#, 1#, 3#, 4#, 5#	0.7417	0.7607	0.7937	0.6842
0#, 1#, 4#, 5#	0.7417	0.7624	0.7937	0.6842
0#, 1#, 2#, 4#, 5#	0.7500	0.7695	0.7937	0.7018
0#, 4#, 5#	0.7583	0.7973	0.9048	0.5965
0#, 1#, 2#, 3#, 4#, 5#	0.7667	0.7878	0.8254	0.7018
0#, 2#, 4#, 5#	0.7833	0.8056	0.8571	0.7018

TABLE VIII. THE TOP 20 ACCURACY ON THE TCM CONSTITUTION CLASSIFICATION TASK

Input Feature	Model	Channel	ACC	F1	sensitivity	specificity
DRS + FF	MkCNN-mf	0#, 2#, 4#, 5#	0.7833	0.8056	0.8571	0.7018
FF	XGBoost	5#	0.775	0.7805	0.7619	0.7895
FF	MkCNN-sf	0#, 2#, 4#, 5#	0.7667	0.7737	0.7619	0.7719
DRS + FF	MkCNN-mf	0#, 1#, 2#, 3#, 4#, 5#	0.7667	0.7878	0.8254	0.7018
DRS + FF	MkCNN-mf	0#, 4#, 5#	0.7583	0.7973	0.9048	0.5965
DRS + FF	MkCNN-mf	0#, 1#, 2#, 4#, 5#	0.75	0.7695	0.7937	0.7018
FF	MkCNN-sf	0#, 4#, 5#	0.7417	0.7514	0.746	0.7368
FF	MkCNN-sf	0#, 1#, 4#, 5#	0.7417	0.7576	0.7778	0.7018
DRS + FF	MkCNN-mf	0#, 1#, 3#, 4#, 5#	0.7417	0.7607	0.7937	0.6842
DRS + FF	MkCNN-mf	0#, 1#, 4#, 5#	0.7417	0.7624	0.7937	0.6842
FF	MkCNN-sf	0#, 1#, 2#, 4#, 5#	0.7333	0.7541	0.7778	0.6842
FF	MkCNN-sf	0#, 1#, 3#, 4#, 5#	0.7333	0.7703	0.8571	0.5965
OPS	LDA	2#	0.725	0.7179	0.6667	0.7895
DRS + FF	SVM-linear	3#	0.725	0.7179	0.6667	0.7895
DRS	SVM-linear	3#	0.725	0.7317	0.7143	0.7368
PF	MkCNN-mf	0#, 3#, 4#, 5#	0.7167	0.7137	0.6825	0.7544
PF	MkCNN-mf	0#, 1#, 3#, 4#, 5#	0.7167	0.7201	0.6984	0.7368
DRS	MkCNN-mf	0#, 1#, 2#, 3#, 5#	0.7167	0.7256	0.7302	0.7018
FF	MkCNN-mf	0#, 3#, 4#, 5#	0.7167	0.7306	0.7302	0.7018
DRS	MkCNN-sf	2#, 3#	0.7167	0.7539	0.8254	0.5965

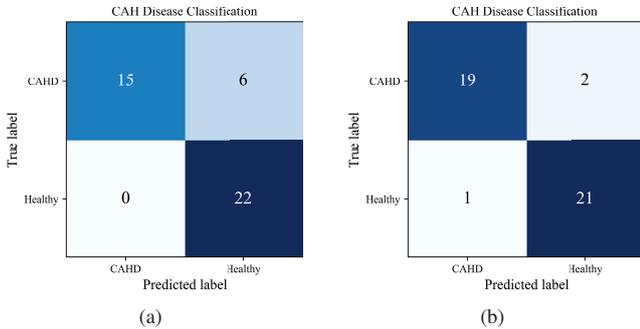


Fig. 11. CAH Disease classification confusion matrix. (a) means using FF single-feature; (b) means using DRS and FF multi-feature

and F1 is 0#, 2#, 4#, 5# channels whose accuracy is 16.7% higher than the best single-channel (1#). This proves that channel combination can greatly improve the performance of pulse signal classification.

In addition, based on the Drift Removed Signal and Fusion Feature multi-feature as input, we test six single-channel and best-performing two-channel 4#, 5# (b₂), three-channel 0#, 4#, 5#, (b₃), four-channels 0#, 2#, 4#, 5# (b₄), five-channels 0#, 1#, 2#, 3#, 4#, 5# (b₅) and six-channels (all) results on five different classification models. As shown in Fig. 13, the four methods such as LDA, SVM-linear, SVM-rbf, and XG-Boost have darker color blocks on a single-channel than multi-

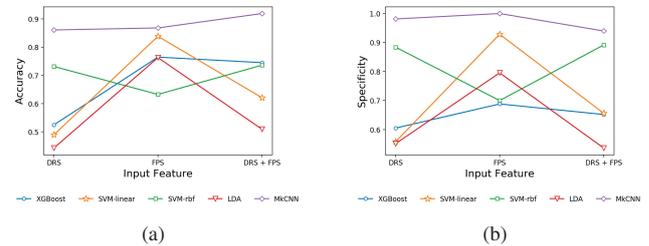


Fig. 12. Comparing the performance of different classification models. (a) is Accuracy performance; (b) is Specificity performance.

channel combinations. On the contrary, MkCNN achieves better results on all multi-channel combinations. By comparing the distribution of dark blocks in different models, it also proved that the traditional machine learning classifier is more suitable for processing simpler data. However, MkCNN can effectively learn complementary information between multi-channel data, and achieve better classification performance.

Comparing the best performance of different input features, different classification models, and different pulse channel combinations: Inspired by the above experiments, we conduct 2853 sets of experiments which consist of 8 input features, 5 classification models, and 63 channel combination methods. Table VIII shows the experimental results and different settings of the TOP 20 accuracy performance.

We achieve the best accuracy, F1-score and Sensitivity using

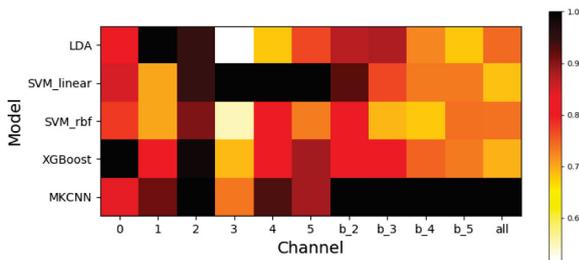


Fig. 13. Performance visualization of the single-channel and multi-channel pulse signals under different classification models. The darker the color, the higher the accuracy.

Drift Removed Signal and Fusion Feature on MkCNN-mf when channel combination is 0#, 2#, 4#, 5#. What's more, our model also achieves better performance than traditional computational pulse signal analysis method on the TCMC dataset, which proves the effectiveness of research ideas in this paper. There are 16 experiments using the combined channel in Table VIII. This shows that the use of pulse signals from multiple positions on the wrist is helpful for TCM constitution analysis.

VI. CONCLUSION

In this paper, we innovatively integrate deep learning methods into pulse signals analysis. We propose the Multi-kernel Convolutional Neural Network for wrist pulse analysis, and use it to explore the combined experiments of different features and channels. Compared with the traditional computational pulse signal analysis methods, our method achieves the best performance on Coronary Atherosclerotic Heart Disease Classification task and TCM Constitution Classification task. The results indicate our work is meaningful in wrist pulse analysis. In the future, we will introduce image data from tongue diagnosis and face diagnosis to our model, and explore how to effectively integrate pulse signals with medical images to better analyze diseases and TCM constitution.

VII. ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 62173195); and the National Key R&D Program Projects of China (Grant No: 2018YFC1707605).

REFERENCES

- [1] K. Kong, H. Chan, Q. Huang, C. Lee, Y. L. Leung, B. Guan *et al.*, "Sphygmopalpation using tactile robotic fingers reveals fundamental arterial pulse patterns," *IEEE Access*, vol. 10, pp. 12 252–12 261, 2022.
- [2] D. Wang, D. Zhang, and G. Lu, "A novel multichannel wrist pulse system with different sensor arrays," *IEEE transactions on instrumentation and measurement*, vol. 64, no. 7, pp. 2020–2034, 2015.
- [3] D. Wang, D. Zhang, and G. Lu, "An optimal pulse system design by multichannel sensors fusion," *IEEE journal of biomedical and health informatics*, vol. 20, no. 2, pp. 450–459, 2015.
- [4] Z. Jiang, D. Zhang, and G. Lu, "A robust wrist pulse acquisition system based on multisensor collaboration and signal quality assessment," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 12, pp. 4807–4816, 2019.
- [5] P. Wang, W. Zuo, and D. Zhang, "A compound pressure signal acquisition system for multichannel wrist pulse signal analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 6, pp. 1556–1565, 2014.
- [6] Y. Chen, L. Zhang, D. Zhang, and D. Zhang, "Computerized wrist pulse signal diagnosis using modified auto-regressive models," *Journal of Medical Systems*, vol. 35, no. 3, pp. 321–328, 2011.
- [7] D. Zhang, W. Zuo, D. Zhang, H. Zhang, and N. Li, "Classification of pulse waveforms using edit distance with real penalty," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 303140, 2010.
- [8] D.-Y. Zhang, W.-M. Zuo, D. Zhang, H.-Z. Zhang, and N.-M. Li, "Wrist blood flow signal-based computerized pulse diagnosis using spatial and spectrum features," *Journal of Biomedical Science and Engineering*, vol. 3, no. 04, p. 361, 2010.
- [9] Y. Chen, L. Zhang, D. Zhang, and D. Zhang, "Wrist pulse signal diagnosis using modified gaussian models and fuzzy c-means classification," *Medical engineering & physics*, vol. 31, no. 10, pp. 1283–1289, 2009.
- [10] Z. Jiang, D. Zhang, and G. Lu, "Radial artery pulse waveform analysis based on curve fitting using discrete fourier series," *Computer methods and programs in biomedicine*, vol. 174, pp. 25–31, 2019.
- [11] D. Zhang, W. Zuo, and P. Wang, "Generalized feature extraction for wrist pulse analysis: From 1-d time series to 2-d matrix," in *Computational Pulse Signal Analysis*. Springer, 2018, pp. 169–189.
- [12] S. Zhou, J. Jia, Q. Wang, Y. Dong, Y. Yin, and K. Lei, "Inferring emotion from conversational voice data: A semi-supervised multi-path generative neural network approach," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [13] G. Aguilar, V. Rozgić, W. Wang, and C. Wang, "Multimodal and multi-view models for emotion recognition," *arXiv preprint arXiv:1906.10198*, 2019.
- [14] Y. Xu, H. Xu, and J. Zou, "Hgmfm: A hierarchical grained and feature model for acoustic emotion recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6499–6503.
- [15] L. Liu, W. Zuo, D. Zhang, N. Li, and H. Zhang, "Combination of heterogeneous features for wrist pulse blood flow signal diagnosis via multiple kernel learning," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 598–606, 2012.
- [16] G. Sannino and G. De Pietro, "A deep learning approach for ecg-based heartbeat classification for arrhythmia detection," *Future Generation Computer Systems*, vol. 86, pp. 446–455, 2018.
- [17] Ö. Yildırım, P. Pławiak, R.-S. Tan, and U. R. Acharya, "Arrhythmia detection using deep convolutional neural network with long duration ecg signals," *Computers in biology and medicine*, vol. 102, pp. 411–420, 2018.
- [18] F. Heutink, V. Koch, B. Verbist, W. J. van der Woude, E. Mylanus, W. Huinck, I. Sechopoulos, and M. Caballo, "Multi-scale deep learning framework for cochlea localization, segmentation and analysis on clinical ultra-high-resolution ct images," *Computer Methods and Programs in Biomedicine*, vol. 191, p. 105387, 2020.
- [19] H. Eun, D. Kim, C. Jung, and C. Kim, "Single-view 2d cnns with fully automatic non-nodule categorization for false positive reduction in pulmonary nodule detection," *Computer methods and programs in biomedicine*, vol. 165, pp. 215–224, 2018.
- [20] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.
- [21] D. Wang, D. Zhang, and G. Lu, "A robust signal preprocessing framework for wrist pulse analysis," *Biomedical signal processing and control*, vol. 23, pp. 62–75, 2016.
- [22] L. Xu, D. Zhang, K. Wang, N. Li, and X. Wang, "Baseline wander correction in pulse waveforms using wavelet-based cascaded adaptive filter," *Computers in Biology and Medicine*, vol. 37, no. 5, pp. 716–731, 2007.
- [23] L. Xu, D. Zhang, and K. Wang, "Wavelet-based cascaded adaptive filter for removing baseline drift in pulse waveforms," *IEEE transactions on biomedical engineering*, vol. 52, no. 11, pp. 1973–1975, 2005.
- [24] I. E. Kaya, A. c. Pehlivanlı, E. G. Sekizkardeş, and T. İbrikli, "Pca based clustering for brain tumor segmentation of t1w mri images," *Computer methods and programs in biomedicine*, vol. 140, pp. 19–28, 2017.
- [25] R. Duda, "Hart. p., e., and stork, dg," pattern classification," *Nueva York: John Wiley and Sons*, 2001.
- [26] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.