Identify cardiac condition through arrhythmia detection with two-dimensional (2D) convolutional neural network (CNN) model

by Andrew Yong Zheng Dao[1]

[1]Department of Computing, University of Wollongong Malaysia

Heart sound signal classification using machine learning models is an exploratory and iterative process as it is a form of cardiac auscultation to understand heart sound. In this paper, an artificial intelligence (AI) based experiment and a desktop application software with the use of machine learning model are designed and developed to enable the diagnosis of heart sound through either audio file recording or AD8232 electrocardiogram (ECG) sensor. Two-dimensional (2D) convolutional neural network (CNN) machine learning model is used to classify the ECG signals which include normal heart sound, murmur heart sound, extrasystole heart sound, and extra heart sound. The process involves segmentation by using Shannon energy and features extraction by calculating two time-domains and five frequency domains from the segmented ECG signal, before passing to the model for training and testing. The classification results of the 2D CNN are at the accuracy rate of 81.48%, after applying data normalization and data augmentation. The classified heart sound is used to identify the heart condition to provide appropriate potential cardiovascular disease (CVD) based on the predicted result of 2D CNN model.

Keywords Electrocardiogram · convolutional neural network · segmentation · feature extraction · cardiovascular disease

1.0 Introduction

Cardiovascular diseases are the conditions that influence the heart function and the main contributor to mortality and morbidity globally. In the current real-life setting, adulthood classic risk factors, such as blood pressure, cholesterol levels, smoking, and adulthood stress are capable to trigger cardiovascular diseases[1]. According to World Health Organization[2], there are about 17.9 million of people die each year and is estimated about 31% of all deaths are caused by cardiovascular diseases and more than 75% deaths take place in developing countries. In Europe, cardiovascular disease accounts for 3.9 million of death cases and over 1.8 million of death cases in European Union[3]. Indeed, the prevalence of cardiovascular disease in the Asia-Pacific region was recorded at 21.1% in year 2011 and are in escalating trend[4].

As the cardiovascular diseases are in the precarious situation, certain techniques and strategies are evaluated and implemented to evaluate the heart condition. Electrocardiograms (ECG) is an ordinary and effective method to evaluate the heart condition based on the heart sound. ECG will capture the arrhythmias information in the form of electrical signal and the signal represents a series of events for the result of polarization and depolarization of cardiac issues[5].

Machine can understand the prospect of human-like sound in various real environment likes security system, smartphones, and autonomous robots that can provide a wide range of applications, such as acoustic surveillance, search in audio archives to retrieve and classify information, as well as acoustic information for controlling intelligence machine[6]. Sound represents in natural environment have substantial diversity and span wide range of frequencies. Supervised learning can be applied for classifying the natural sound as well as heart sound. Supervised learning normally applied for classification and regression in which the supervised learning model is trained and learned from many samples which also known as training dataset[7].

In the recent years, the medical field has enhanced to deploy computerized analysis and diagnosis in signal processing through the use of bio-signals, such as ECG which provides large difference in the shape and the pattern between normal heart sound and abnormal heart sound, based on the respective signal varies with respective to time, amplitude, frequency content, and intensity[8, 9].

The organization of this paper is in the following manner. Section 2 introduces the heart sound dataset, methods, the proposed two-dimensional (2D) convolutional neural network (CNN) model, hardware setup, and software design. The result obtained from the performance of the proposed 2D CNN model are given in Section 3. Finally, the paper is concluded in Section 4.

2.0 Methodology

In the section below, the preparation of data, implementation of software and hardware tools, and the design of two-dimensional (2D) convolutional neural network (CNN) model that used for classifying heart sounds in the electrocardiograms (ECGs) are explained and designed. The method involves pre-processing of data, model's structure, hardware design, and software design. The pre-processing of the sample data involves the process of segmentation and feature extraction.

2.1 Heart sound dataset description

The heart sound audio recording dataset is collected from PASCAL Classifying Heart Sounds Challenges, which is published by Bentley, *et al.*[10]. The dataset consists of dataset A and dataset B. Dataset A contains the audio data for four types of heart sound, whereas dataset B contains three types of heart sound. Dataset A consists of normal heart sound, murmur heart sound, extra heart sound, and artifact heart sound, whereas dataset B contains normal heart sound, murmur heart sound, and extrasystole heart sound. Likewise, each of the heart sound audio clips are recorded at the varying length in between one second to 30 seconds. According to Chakir, *et al.*[11], heart sound data are collected from two sources in which dataset A is retrieved via iStethoscope Pro iPhone app, however dataset B is retrieved via digital stethoscope DigiScope. Apart from artifact heart sound, the other heart sound is grouped and categorized into four files separately. Apart from this, there are three samples of the extra heart sound are collected from Advanced Physical Diagnosis Learning and Teaching which is published by University of Washington[12].

2.2 Segmentation process

In this paper, segmentation process that proposed by Beyramienanlou and Lotfivand[13] and Chen and Zhang[14] are referred. There are only some processes are implemented which includes normalization of the signal data, Shannon energy computation, averaging the Shannon energy, forming the Shannon energy envelope, and applying threshold to extract the signal. Before segmenting the heart sound, each heart sound signal is passed for denoising process using fast Fourier transform (FFT). FFT represents the frequency spectrum of the sound signal[15] which is normally used in removing the background noise and the ambient noise in sound analysis. The description and the flow of segmentation process is described as below and shown in Fig.1. The segmentation process starts with denoising followed by normalization, calculate Shannon energy and Shannon energy envelope, and finally extract the threshold value.

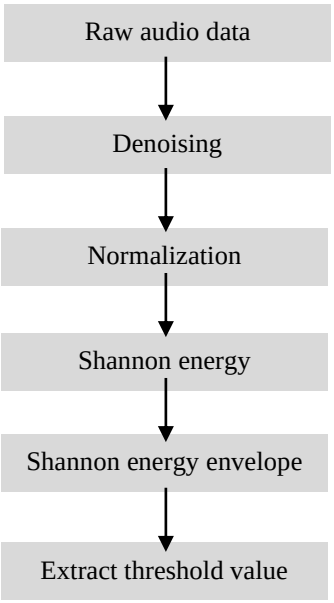


Fig. 1 The flow of segmentation process.

2.2.1 Denoising

The open source of Python library, Noisereduce is used to denoise the signal. Fast Fourier transform (FFT) is calculated over the noise audio clip and signal audio clip to transform the signal into its frequency domain. Statistics are calculated using the transformed frequency domain of noise audio clip and followed by computing the threshold based on the statistics on the noise audio clip. A mask is identified after comparing the FFT with the threshold value. The mask is then smoothened with a filter over frequency and time before applying to the FFT of the signal audio clip. Its inverse transform is then calculated to convert the frequency back to its original signal.

2.2.2 Normalization

The denoised signal is then normalized with the maximum value of the signal. The formula for normalizing the signal is shown in Equation (1) below.

$$a[n] = \frac{|s[n]|}{max_{i=1}^{N}|s[n]|} \quad (1)$$

where $a[n]$ is a normalized amplitude and $s[n]$ represents the signal after applying mask and inverse fast Fourier transform, and N represents the number of samples.

2.2.3 Shannon energy

Shannon energy is then calculated over the normalized signal. This energy is a square of the input signal because signal square is proximity to signal energy[13]. Likewise,

Shannon energy calculates the average of signal energy which discount high component into low component. Shannon energy computes the energy of local spectrum for each sample in the signal with the use of Equation (2) as followed.

$$s[n] = -a^2[n] \log (a^2[n]) \quad (2)$$

where $s[n]$ is the Shannon energy, $a^2[n]$ represents the squared signal at n^{th} signal.

2.2.4 Average Shannon energy

The vector of Shannon energies is then passed for averaging. Shannon energies are averaged in continuous signal with 0.02 seconds intervals. The average Shannon energy is computed using the Equation (3) below.

$$E_s(t) = \frac{1}{N} \sum_{i=1}^{N} s[n] \quad (3)$$

where N represents the number of the sample in 0.02 intervals[16], and $E_s(t)$ represents the value of average Shannon energy of the sample after averaging the total of each Shannon energy at the time t with N within 0.02 seconds intervals.

2.2.5 Shannon energy envelope

The average Shannon energy is then normalized to convert into energy package, which is also known as Shannon energy envelope using its mean and standard deviation. The envelope decreases the signal base and placing the signal below the baseline. Equation (4) shown the calculation of the Shannon energy[14].

$$P[t] = \frac{E_s[t] - \mu(E_s[t])}{\sigma(E_s[t])} \quad (4)$$

where $\mu(E_s[t])$ is the mean for the random variable vector of $E_s[t]$,$\sigma(E_s[t])$represents the standard deviation of $E_s[t]$, and $P[t]$ is the Shannon energy envelope at the time t. Likewise, $P[t]$ also known as normalized average Shannon energy.

2.2.6 Threshold value

A threshold value is the definition to determine the peaks (QRS complex location) with the fact that sample with greater amplitude than the threshold is chosen as output. The threshold value is defined using the mean, standard deviation, and a constant. The formula is defined in Equation (5) below.

$$threshold = |k\mu(1-\sigma^2)| \; if \; \sigma < \mu, \quad threshold = |k\sigma(1-\mu^2)| \; if \; \mu < \sigma \quad (5)$$

where k refers to a constant. In this case, the constant value is defined as 0.001. The extracted signal is then passed to extract the features as described in Section 2.3 below.

2.3 Features extraction

A total of seven features are extracted from different domains. The features are processed using "librosa" and "pyAudioAnalysis" libraries. The extracted features included zero-crossing rate, mel-frequency cepstral coefficients (MFCCs), spectral centroid, spectral roll-off, spectral flux, frequency, and energy entropy. The zero-crossing rate and the energy entropy are the time domain features, whereas the other five features belong to the frequency domain features[17]. The definition of each of the sample is defined in Table 1.

Table 1. Definition for the feature[16, 18, 17, 19, 20].

Feature name	Definition
Zero-crossing rate	The ratio of the sign change of the spectrum of an audio signal, that can be interpreted as a change in a signal from a positive number to negative number, vice versa.
Mel-frequency cepstral coefficients	The distribution of energy for a signal in the frequency domain and refers to a perceived frequency.
Spectral centroid	The center of gravity for the spectrum.
Spectral roll-off	The k^{th} percentile of the total power spectral distribution in audio signal in which k is either 85% or 95% spectral roll-off of the signal.
Spectral flux	The independent of total power and the phase consideration that measures the rate of change in spectral shape that calculated as the frame-to-frame magnitude spectral difference.
Frequency	Fast Fourier transforms of the signal.
Energy entropy	The measurement of abrupt changes, the impurity of normalized energies for the sub-frames.

Each feature in Table 1 is calculated using the Numpy which is Python open-source library. Each feature is extracted as a single value but the feature MFCCs and frequency are simplified into its statistical value. MFCCs and frequency features are represented by mean and standard deviation value, respectively. The data is then saved to a comma-separated value (CSV) file. Likewise,

the labelling is provided to each row of the entries. The labelling represents by *sound type* column which represents the label of the heart sound. The numerical values 0, 1, 2, and 3 are used where each represent normal, murmur, extrasystole, and extra heart sound, respectively.

2.4 The proposed two-dimensional convolutional neural network model

The structure of the two-dimensional (2D) convolutional neural network (CNN) model resembles a multi-layer perceptron (MLP) and each neuron in the MLP is associated with an activation function that maps the weighted inputs to the output[21]. There are some basic layers in CNN model, which are convolutional layer, max-pooling layer, dropout, batch normalization, and fully connected layer or a dense layer, with a rectified linear unit (ReLU) activation function in CNN architecture[22]. The batch normalization layer is added to normalize the output of the previous layer and allow every CNN layer to learn efficiently and independently. Also, the batch normalization layer is utilized to avoid overfitting of the CNN model through regularization. Table 2 summarizes the structure of the proposed 2D CNN model.

Table 2. The information about the layers and the parameters utilized in the proposed 2D CNN model.

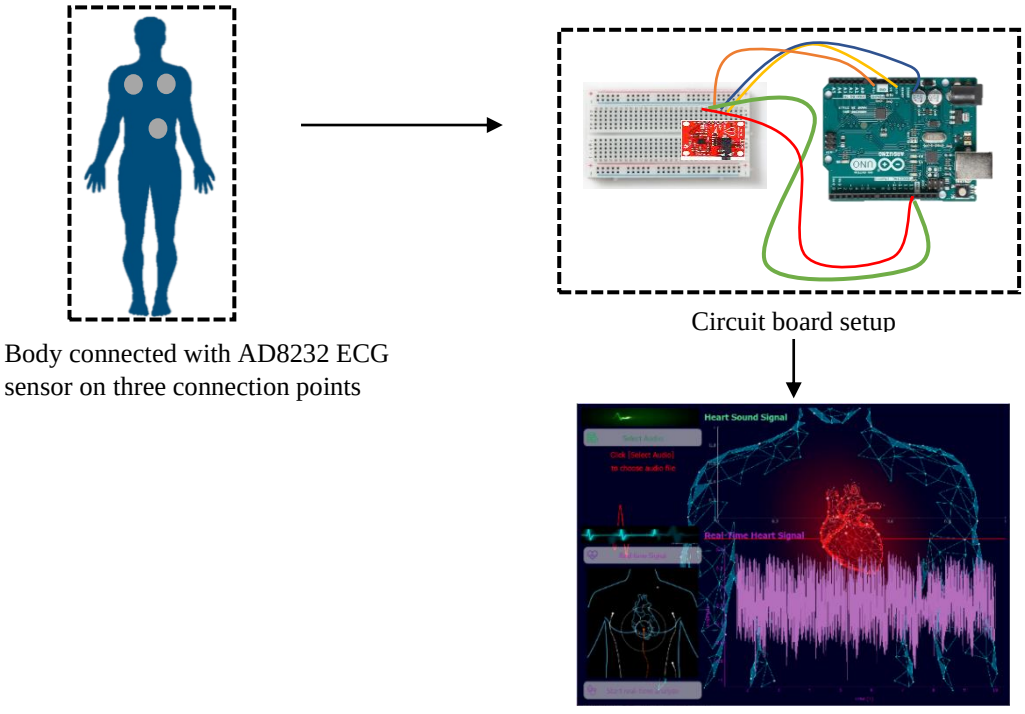
Layer to layer	Type	Layer parameters
0-1	Convolution 2D	Output channels = 16, kernel = (1,1), activation function = ReLU
1-2	Batch Normalization	-
2-3	Max-Pooling 2D	Pooling size = (1, 1)
3-4	Dropout	Rate = 0.2
4-5	Convolution 2D	Output channels = 32, kernel = (1,1), activation function = ReLU
5-6	Batch Normalization	-
6-7	Max-Pooling 2D	Pooling size = (1, 1)
7-8	Dropout	Rate = 0.2
8-9	Convolution 2D	Output channels = 32, kernel = (1,1), activation function = ReLU
9-10	Batch Normalization	-
10-11	Max-Pooling 2D	Pooling size = (1, 1)
11-12	Dropout	Rate = 0.2
12-13	Convolution 2D	Output channels = 32, kernel = (1,1), activation function = ReLU
13-14	Batch Normalization	-
14-15	Max-Pooling 2D	Pooling size = (1, 1)
15-16	Dropout	Rate = 0.2
16-17	Average-Pooling2D	-
17-18	Flatten	-
18-19	Dense	Output channels = 32, activation function = ReLU
19-20	Batch Normalization	-
20-21	Dropout	Rate = 0.2
21-22	Dense	Output channels = 32, activation function = ReLU
22-23	Batch Normalization	-
23-24	Dropout	Rate = 0.2
24-25	Dense	Output channels = 16, activation function = ReLU
25-26	Batch Normalization	-
26-27	Dropout	Rate = 0.2
27-28	Dense	Output channels = 4, activation function = Softmax

The hidden layer of the CNN model is differed from the normal neural network model. The max-pooling layer is located in between the normal hidden layer or known as convolutional layer and a layer after the input layer in

CNN. This is because max-pooling layer provides the output which has maximum value of all its respective convolutional output layer[24]. This is the downsampling process to the detection of features. Besides, the flatten layer and the dense layer are only added before the classification result. The flatten layer transforms the three-dimensional (3D) output into one-dimensional which are the vectors that can be processed by dense layer[22]. The last layer, softmax layer will perform the classification process. The rectified linear unit (ReLU) activation function is implemented to each convolutional layer in CNN model. ReLU activation function is more efficient as it avoids all the neurons to activate simultaneously[24]. ReLU activation function is formulated as $f(x) = \max(0, x)$. Softmax activation function normally considers for the multi-class classification instead of the binary classes[25, 26]. The activation function is required because it allows a more dynamic neural network and capable to extract complex information from data and achieve the non-linear mapping from input to output[24].

2.5 Electrocardiogram (ECG) sensor

After training the two-dimensional (2D) convolutional neural network (CNN) model, the trained weight of the model is saved in the H5 file which is a data file saved in the hierarchical data format 5 (HDF5). This file will be used to predict the input signal from audio file or electrocardiogram (ECG) sensor. The ECG sensor is implemented to conduct the performance testing of the trained 2D CNN towards the real-life heart sound prediction. Therefore, AD8232 ECG is selected as the sensor to receive the analogue signal of the heart sound. Likewise, the analogue to digital converter is applied to convert the analogue heart sound into its digital format before passing to 2D CNN model. Although the AD8232 sensor is different from the iStethoscope Pro iPhone app and digital stethoscope DigiScope, but all these hardware record the heart sound in the ECG format. Indeed, Arduino UNO board is chosen as the microcontroller to interface with AD8232 ECG sensor. Moreover, the breadboard is used as a platform to connect the microcontroller and the sensor. Fig. 2 shows the abstract view of the overall self-designed architecture of the flow of hardware sensor to the interface of the software system. that illustrates the signal that received from the sensor.



Body connected with AD8232 ECG sensor on three connection points

Circuit board setup

Graphical user interface with ECG signal from sensor

Fig. 2 Abstract view of interaction between hardware and software.

2.6 System architecture design

A software system is developed to interact with the hardware design which shown in Fig. 2. by using Python framework and PyQt library.

The architecture of the design for the software is divided into four modules, which are front-end module, back-end module, machine learning module, and database module.

The design of the front-end module involves the input selection, illustration of heart sound signals and the result of diagnosis. The software provides the input selection options which either retrieve signal from recorded audio file in *.WAV* format or real-time signal retrieval through AD8232 sensor. The signal of the heart sound (raw signal) from the input source is then illustrated on the graphical user interface (GUI) screen for navigation. Additionally, each processed signal in each stage of segmentation process that start from denoising process as shown in Fig. 1, the extracted feature value, and the diagnosed result of the heart sound with its corresponding potential heart diseases are also displayed on the GUI screen after the diagnosis process is completed.

The machine learning framework is partially independence module to the system because the training and the testing processes of the machine learning model do not involve in the diagnosis process. The machine learning framework is designed to train and test the two-dimensional (2D) convolutional neural network (CNN) model and save the weight of the model as HDF5 format when the model reaches its excepted accuracy. In the machine learning module, the segmentation process and feature extraction process are included to process and generate the data required to train the model. The structured data is saved into CSV file, passing the file as an input to train and test the model. Likewise, the labelling of the type of the heart sound is given to each record in the CSV file. Indeed, the saved weight is involved in the system while it is used to perform diagnosis process. In the diagnosis process, the raw input of the signal from the front-end will pass to machine learning frame to undergo segmentation and feature extraction process, in order to generate the same data before implementing the prediction to the trained weight to identify the heart sound.

The database module is used to record the user's information and the result after the diagnosis process. The data to be recorded includes username, password, age, gender, weight, height, status which is the identified type of heart sound, type of diagnosis which is either through audio file or sensor, and the date and time of diagnosis. Google's firebase is used to design the database. Firebase is the real-time database which enable the real-time update of the data after each diagnosis and provide responsive experience to the user.

The back-end module of the software is used to control all functionalities and execute the logic flows of the system, starting from signal input to the signal illustration, followed by segmentation process, extraction of feature value, prediction of machine learning model, result and record illustration, and read and write the record from and to Google's firebase.

The overview of the flow for self-designed system architecture is illustrated in Fig. 3.

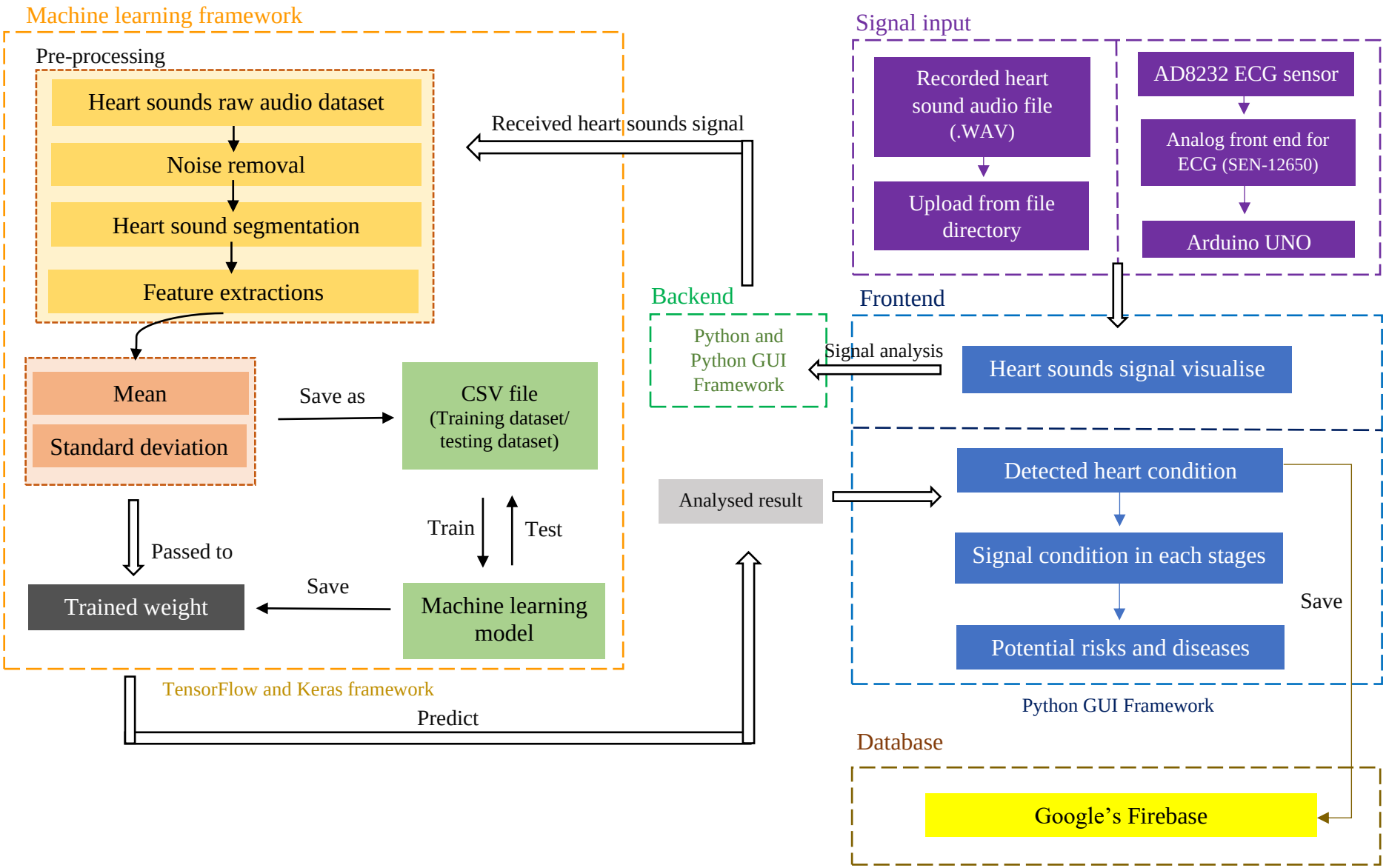


Fig. 3 The abstract view of the overall system architecture design.

3.0 Results

In this section, the results of each process of heart sound signal processing, and the trained result and tested result of 2D CNN are shown. Also, the confusion matrix and the receiver operating characteristic (ROC) curve are computed and plotted to evaluate the overall classification performance of the proposed 2D CNN model, respectively.

3.1 Signal visualization

The clean signal and the background noise are separated before the application of the segmentation. The clean heart sound signals and the background noise is illustrated using the red and blue colours, respectively. The signal in red colour is passed for segmentation and feature extractions which illustrated in Fig. 4. In addition, the graph for the Shannon energy envelope, $P(t)$ with its respectively signal to detect the "lub" sound and the "dub" sound is shown in Fig. 5. Generally, the "lub" sound is known as $S1$, whereas "dub" sound is known as $S2$.

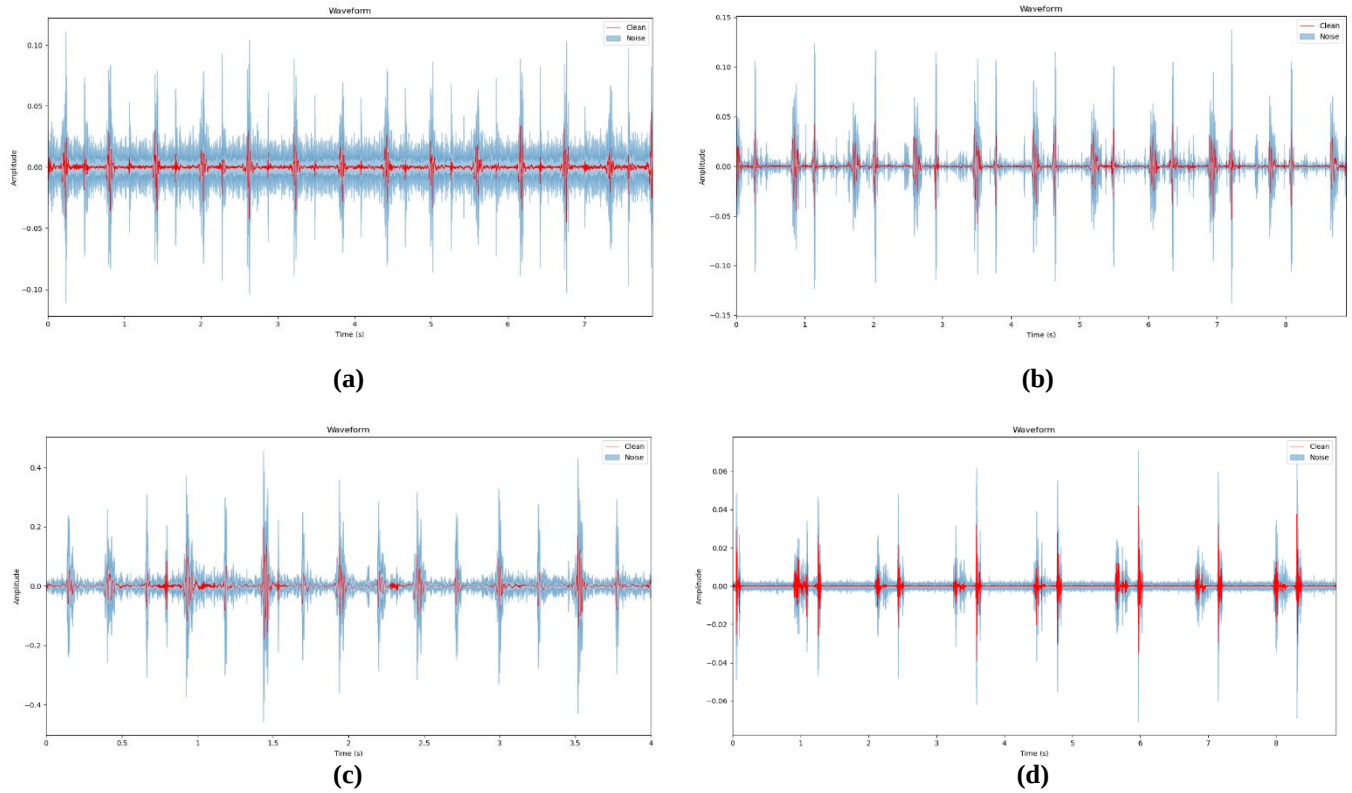


Fig. 4 The visualization of the clean signals in time domain spectrum. (a) Normal heart sound. (b) Murmur heart sound. (c) Extrasystole heart sound. (d) Extra heart sound.

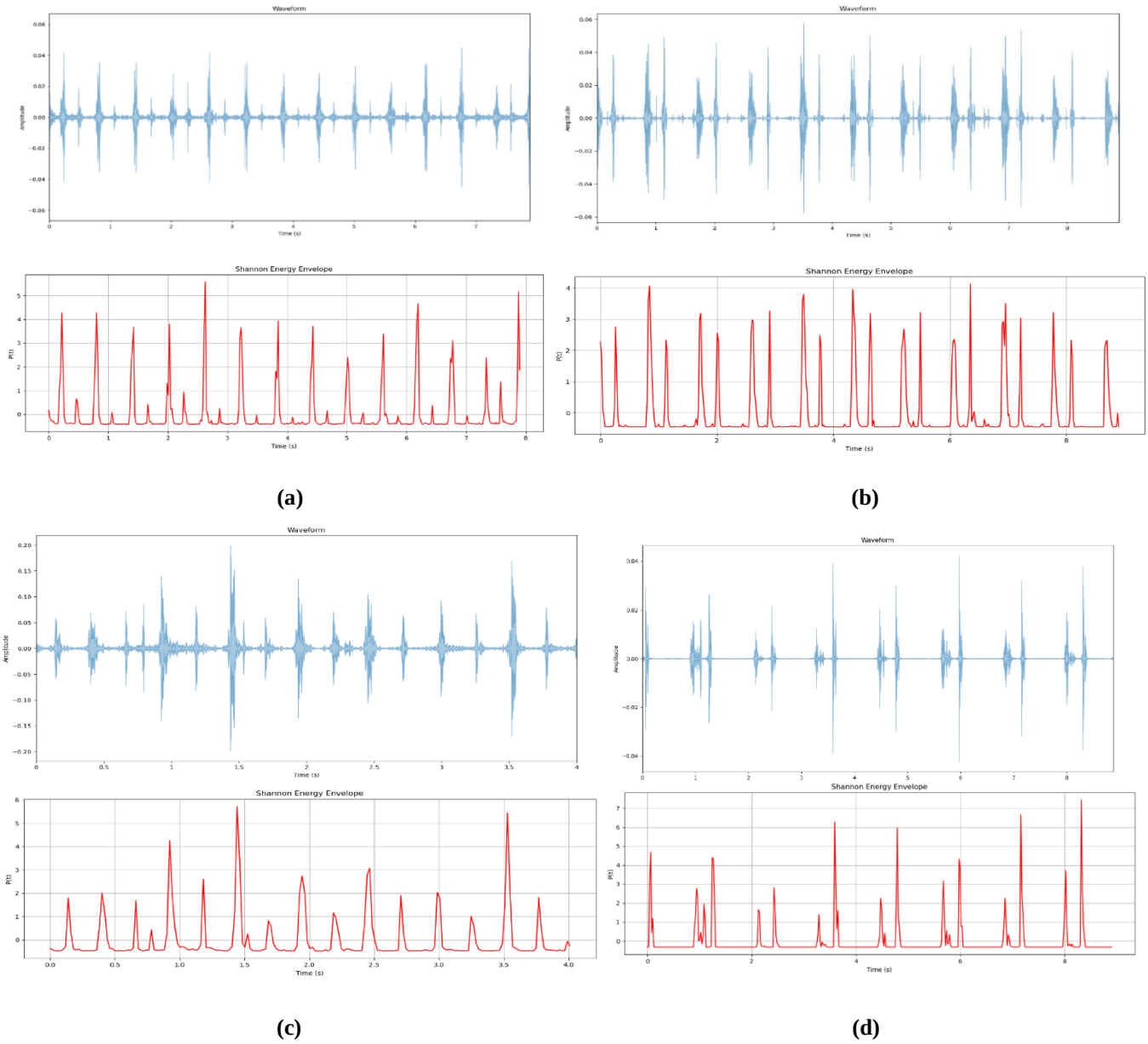


Fig. 5 The graphs that shown the matching of the S_1 and S_2 heart sound with the corresponding Shannon energy envelope. (a) Normal heart sound. (b) Murmur heart sound. (c) Extrasystole heart sound. (d) Extra heart sound.

3.2 Model accuracy

The proposed two-dimensional (2D) convolutional neural network (CNN) model is trained and tested with the dataset in comma-separated value (CSV) format. The dataset is then split into 80% of training set, 10% of validation set, and 10% of testing set.

The training of the model which with the structure shown in Table 2 has undergone several trials. In each trial, the number of the file samples used for each type of heart sound is identical, but the number of iteration, epoch, and batch size are adjusted accordingly. The experiment is to test on the effect of the epoch size and the batch size to the accuracy of the 2D CNN model in binary classification without changing the number of file samples. In addition, the epoch is the number of times that the 2D CNN algorithm works through the training dataset, whereas the batch size is the number of samples processed before the weights of the 2D CNN model are updated.

After the first trial training of the model, the model shows high bias towards normal heart sound and murmur heart sound. This is because the number of file samples used

for the normal and murmur heart sound is greater than the extrasystole and extra heart sound.

In the second trial 2.0, the number of the file samples for the normal and the murmur heart sound is reduced to equalize the file samples among four types of heart sound. However, the training accuracy of the model is reduced. Therefore, the threshold value which is derived while forming the Shannon energy envelope is included as one of the features to train the model in the second trial 2.1. However, the second trial does not show any improvement in the training accuracy.

In the third trial, the file samples for the normal and the murmur heart sound is then increased to 46 to equalize the number of the file samples with the file samples of the extrasystole heart sound. The file samples of the extra heart sound are kept as 19 due to the lack of data. As there is no improvement toward the training accuracy in the third trial, the augmentation and normalization techniques are applied by executing the fourth trial.

In the fourth trial, the data augmentation technique has implemented to increase the heart sound samples of the extrasystole heart sound and extra heart sound. The data augmentation is applied using the open source "Audiomentations" library. After the testing with the audiomentation technique, there are some improvements on the training accuracy of the model. To improve the performance, each row of the signal is then normalized with the minimum value and the maximum value of the respective column to avoid the distortion different in the ranges of feature values. After the data normalization, the kernel size of the 2D CNN is adjusted from 1 × 1 to 2 × 2 to filter the signal input to improve the accuracy of the model. The result of each trials of the 2D CNN training is shown in Table 3.

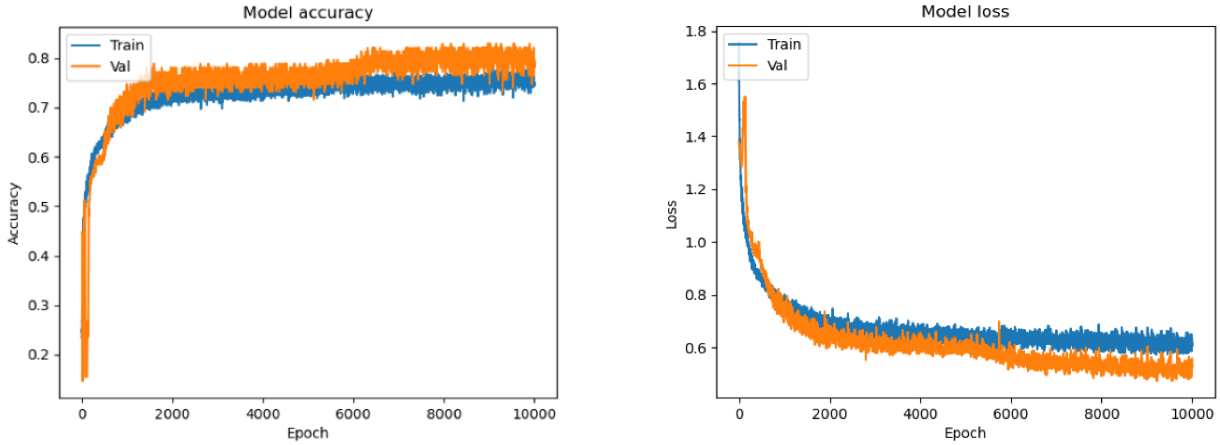
Table 3. The result of the training accuracy of each trials for the 2D CNN.

First trial				
Types of heart sound	**Normal heart sound**	**Murmur heart sound**	**Extrasystole heart sound**	**Extra heart sound**
File samples	350	127	46	19
Epoch	**Batch size**	**Training accuracy (%)**	**Loss**	
50	10	64.39	0.92	
100	10	64.58	0.94	
100	20	64.58	0.92	
100	50	64.39	0.92	
200	10	64.21	0.92	
1000	10	65.18	0.88	
1000	20	66.61	0.84	
10000	10	64.58	0.89	
Second trial 2.0				
Types of heart sound	**Normal heart sound**	**Murmur heart sound**	**Extrasystole heart sound**	**Extra heart sound**
File samples	31	34	46	19
Epoch	**Batch size**	**Training accuracy (%)**	**Loss**	
500	10	52.31	1.04	
3000	10	47.69	1.05	
10000	10	35.38	1.34	
Second trial 2.0 with threshold value				
Epoch	**Batch size**	**Threshold value**	**Training accuracy (%)**	**Loss**
1000	10	Exclude	63.08	0.86
1000	10	Include	61.54	0.80
2000	10	Exclude	50.77	1.04
2000	10	Include	42.31	1.24
5000	10	Exclude	56.92	1.00
5000	10	Include	35.38	1.34
Third trial				

Types of heart sound	Normal heart sound	Murmur heart sound	Extrasystole heart sound	Extra heart sound
File samples	46	46	46	22
Epoch	**Batch size**	**Threshold value**	**Training accuracy (%)**	**Loss**
1500	10	Include	51.25	1.04
2000	10	Include	53.13	1.04
Fourth trial				
Types of heart sound	**Normal heart sound**	**Murmur heart sound**	**Extrasystole heart sound**	**Extra heart sound**
File samples	350	127	598	286
Fourth trial with audio augmentation				
Epoch	**Batch size**	**Threshold value**	**Training accuracy (%)**	**Loss**
1000	100	Include	68.80	0.71
Fourth trial with data normalization				
Epoch	**Batch size**	**Threshold value**	**Training accuracy (%)**	**Loss**
4000	500	Include	71.67	0.65
4000	600	Include	74.69	0.62
Fourth trial with (2 × 2) kernel size				
Epoch	**Batch size**	**Threshold**	**Training accuracy (%)**	**Loss**
6000	600	Include	77.74	0.57
8000	600	Include	79.72	0.51
10000	600	Include	81.48	0.46

As shown in Table 3, the best result for the accuracy is 81.48%. In this case, the epoch and batch size are increased steadily until the model reach at a better accuracy. Also, the model shows a better performance after data augmentation and data normalization.

The well defined batch size is obtained at 600 after training on the normalized data. This batch size is kept for the next training. The highest accuracy of the training is obtained with the epoch defined at 10,000 and after refining the kernel size of the 2D CNN model to 2 × 2 from the earlier designed model structure as shown in Table 2. The results proved that the model requires large samples and training times to achieve a higher accuracy. The result of the training accuracy of the 2D CNN model is shown in Fig. 6



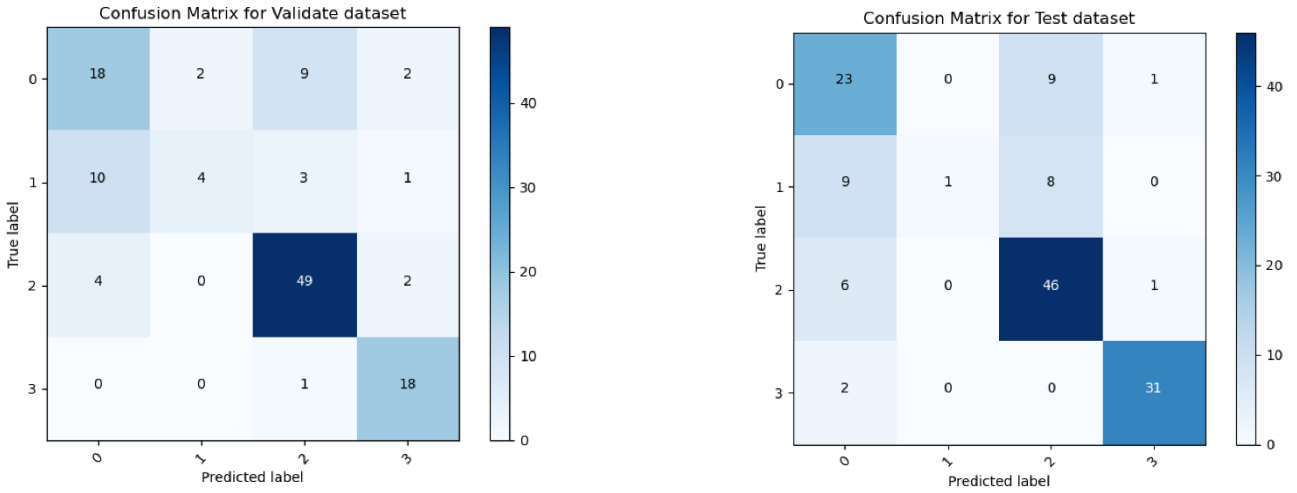
(a) **(b)**

Fig. 6 Performance graphs during the training phase with 2 × 2 kernel size. (a) The accuracy of the proposed model. (b) The loss of the proposed model.

3.2.1 Confusion matrix

The validate set and the test set are passed for model evaluation. The accuracies for the validate set and test set are 72.36% and 73.72%, respectively. While the losses of the validate set and test set are 0.76 and 0.76,

respectively. The confusion matrix in Fig. 7 is used to further evaluate the performance of the model.



(a) **(b)**

Fig. 7 The confusion matrix for the proposed 2D CNN model. (a) Validate dataset. (b) Test dataset.

The accuracy, specificity, sensitivity, recall, and precision are calculated after the value of the elements is determined. According to Krstinić, *et al.* [27] and Narváez, *et al.*[28], the method to calculate the accuracy and specificity, and sensitivity are shown in Equations (6) to (11) below,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (6)$$

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN} \times 100 \quad (7)$$

$$Specificity = \frac{TN}{FP + TN} \times 100 \quad (8)$$

$$Sensitivity = \frac{TP}{FN + TP} \times 100 \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (11)$$

where TP, TN, FP, and FN each represents true positive, true negative, false positive, and false negative, respectively. The results of the calculation for validate and testing datasets are shown in Tables 4 and 5, respectively.

Table 4. The performance matrix of validate dataset for each class.

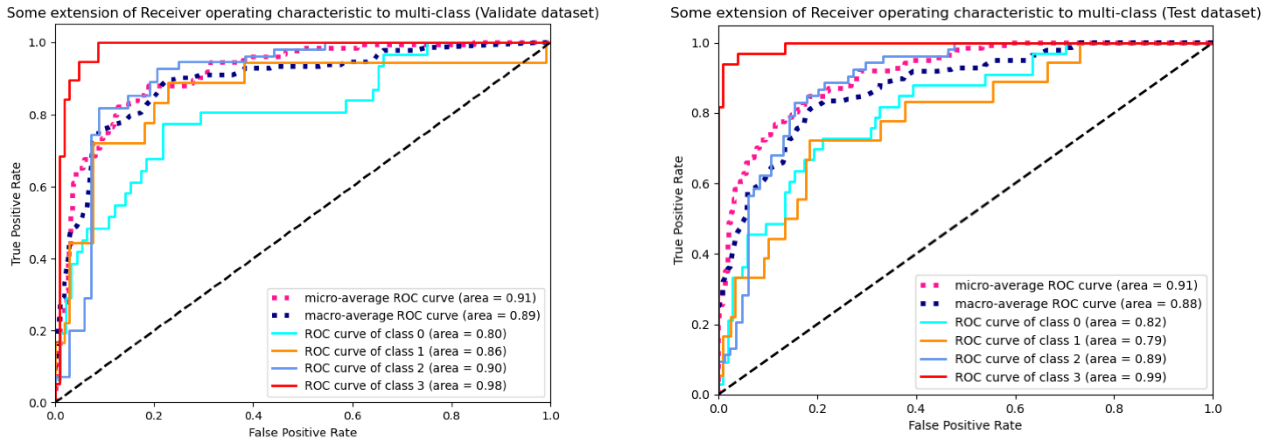
Classes	Accuracy (%)	Error rate (%)	Specificity (%)	Sensitivity (%)	Recall (%)	Precision (%)
Normal	78.05	21.95	84.78	58.07	58.07	56.25
Murmur	86.99	13.01	98.10	22.22	22.22	66.67
Extrasystole	84.55	15.45	80.88	89.09	89.09	79.03
Extra	95.12	4.88	95.19	94.74	94.74	78.26

Table 5. The performance matrices of test dataset for each class.

Classes	Accuracy (%)	Error rate (%)	Specificity (%)	Sensitivity (%)	Recall (%)	Precision (%)
Normal	80.29	19.71	83.65	69.70	69.70	57.50
Murmur	87.59	12.41	100.00	5.56	5.56	100.00
Extrasystole	82.48	17.52	79.76	86.79	86.79	73.02
Extra	97.08	2.92	98.08	93.94	93.94	93.94

3.2.2 Receiver operating characteristic (ROC) curve

Alternatively, the receiver operating characteristic (ROC) curve is employed to evaluate the overall classification performance. The ROC is a graphical curve that plots the relationship between cost, which is the false positive rate (FPR) and benefit, which is the true positive rate (TPR) as the decision threshold varies in the proposed 2D CNN[29]. The ROC curves for both validate dataset and test dataset are shown in Fig. 8.



(a) **(b)**

Fig. 8 ROC curve for the proposed 2D CNN model. (a) Validate dataset. (b) Test dataset.

In the ROC curve, the TPR is plotted against the FPR. The curves shown in the graph are the probability curves. The area under curve (AUC) is calculated to represent the degree or separability measurement. In other word, AUC tells the capabilities of the model in distinguish between classes. Fayzrakhmanov and Kulikov [29] stated that AUC is a metric to evaluate both precision and recall. The micro-average and macro-average are calculated for multi-class ROC curve by using the weight. The weight of micro-average ROC is based on the number of samples in each class, whereas the weight of macro-average is the same for all classes[30]. A good model gives an AUC near to one which is 100% as it shows a good measure of separability. The more separation results in larger area between the ROC curve and the diagonal (dotted line), the higher the AUC value[31]. The AUC value obtained from the ROC curve is tabulated in the Table 6.

Table 6. ROC based performance metrics for 2D CNN heart sound classification.

Experiment / dataset	Micro-average curve area	Macro-average curve area	ROC curve area for class 0 (Normal)	ROC curve area for class 1 (Murmur)	ROC curve area for class 2 (Extrasystole)	ROC curve area for class 3 (Extra)
Validate	0.91	0.89	0.80	0.86	0.90	0.98
Testing	0.91	0.88	0.82	0.79	0.89	0.99

Based on the result shown in Tables 4 to 6, the proposed model performs well in classifying the extrasystole and extra heart sound, followed by normal heart sound and murmur heart sound. The model shows poor performance in classifying the murmur heart sound due to its low sensitivity value and lower AUC value in testing set. This could cause by small amount of data compare to other classes. On the other hand, the model presents a good performance in classifying the extrasystole and the extra heart sound because of the data augmentation. The trained model is then saved in a hierarchical data format 5 (HDF5) file and loaded this model to predict the electrocardiograms (ECG) signal from the real-time AD8232 transmission. The experiment is repeated ten times with a normal user body and ended up with 50% of predictive accuracy. In short, five out of ten trials are predicted correctly by the model. Apart from this, the prediction of the ECG signal is highly relying on the data received from the input signal. Any fault or flaw from the signal could degrade the accuracy of the proposed model.

3.3 System software platform

The software for this application is developed as a cross-platform desktop application software which can be used for both real-time detection and pre-recorded detection. There are two main GUI screen for the desktop application. The first screen is the audio acquisition page which allow user to input the ECG signal based on their preference and illustrates the signal input after receiving the signal. The second page illustrates the analysis result of the signal.

3.3.1 Audio acquisition page

The audio acquisition page is divided into two sections. Fig. 9(a) shows the selection of the pre-recorded audio file, while Fig. 9(b) shows the real-time signal transmission from the AD8232 sensor. Indeed, both sections are capable to illustrate the ECG signal for visualization as shown in Fig. 9.

3.3.2 Result page

The result page is further divided into three subpages which include ECG signal processing subpage, feature extraction subpage, classification subpage, and historical record subpage.

3.3.2.1 ECG signal processing page

This page illustrates all the signal in five processing stages. The stages include denoising stage, normalization stage, forming of Shannon energy envelope, the location of 'lub' and 'dub' signal, and finally the clean signal which without noise. The screenshot of this screen is illustrated as in Fig. 10.

3.3.2.2 Feature extraction page

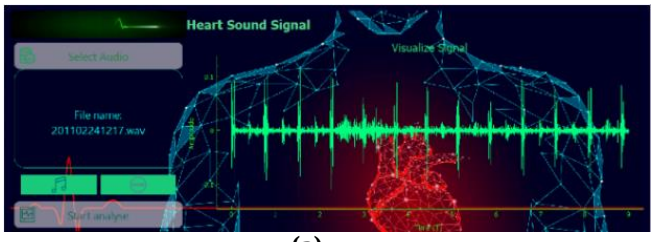
All the value of the feature as stated in Table 1 will be shown in this page. The features value will be extracted from the clean signal after the end of signal processing stage. These feature values are the attribute that used by model to predict the input of ECG signal. The screenshot of this page is shown in Fig. 11.

3.3.2.3 Classification page

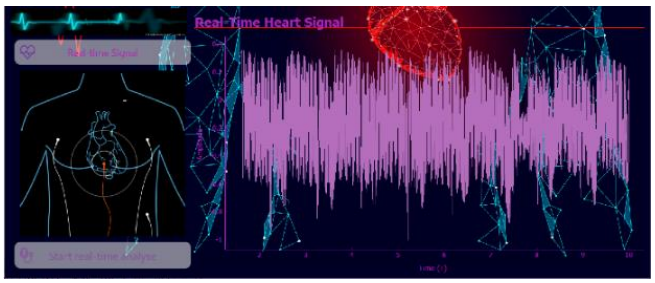
This page shows the result of the model prediction. The information includes the probability value (in term of percentage, %) that the model predicted among different classes, and the list of potential diseases which based on predicted result. In addition, the class with the highest probability value will be selected as the diagnosed result. The screenshot of this page can be viewed in Fig. 12.

3.3.2.4 Historical record page

This page shows the records of the user's details and user's diagnosis details after reading from Google's firebase. The header part is showing the user's details such as username, age, gender, weight, height whereas the line-item part shows the status, type of diagnosis, and the date and time of diagnosis. The screenshot of this page is shown in Fig. 13.



(a)



(b)

Fig. 9 The screenshot of the audio acquisition page. (a) Audio file selection and (b) Transmission from AD8232 sensor.

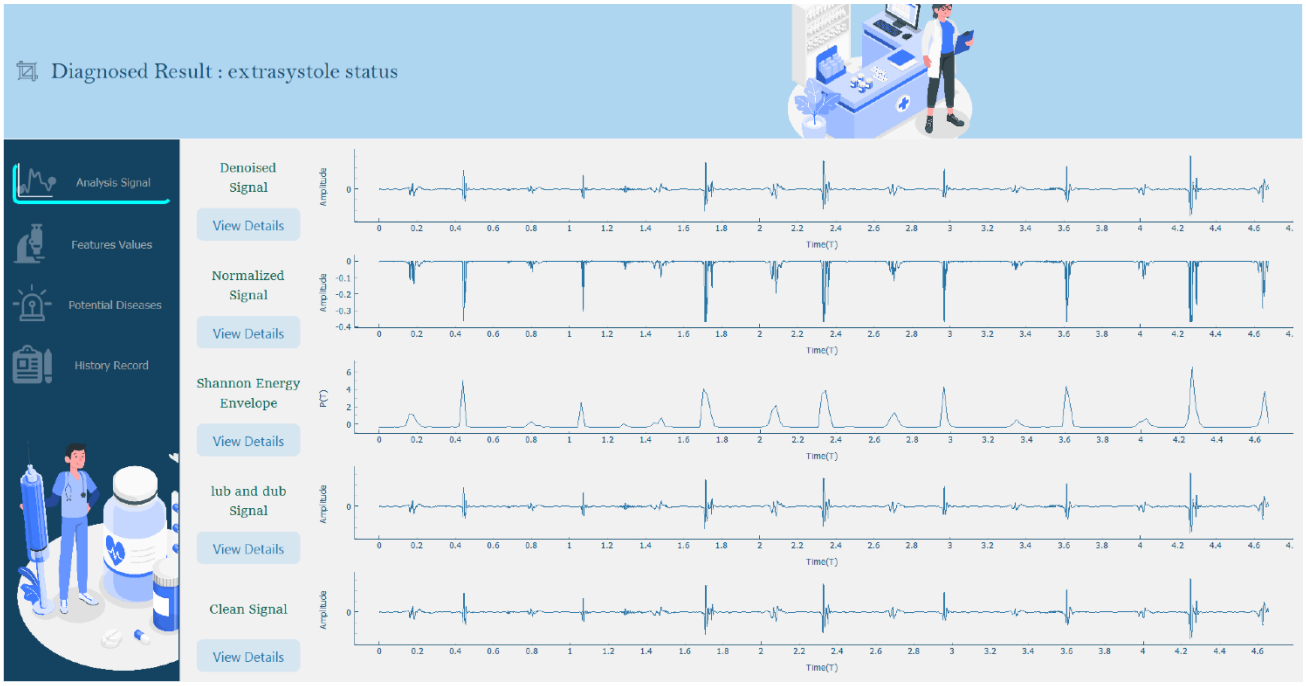


Fig. 10 The screenshot for signal processing stage

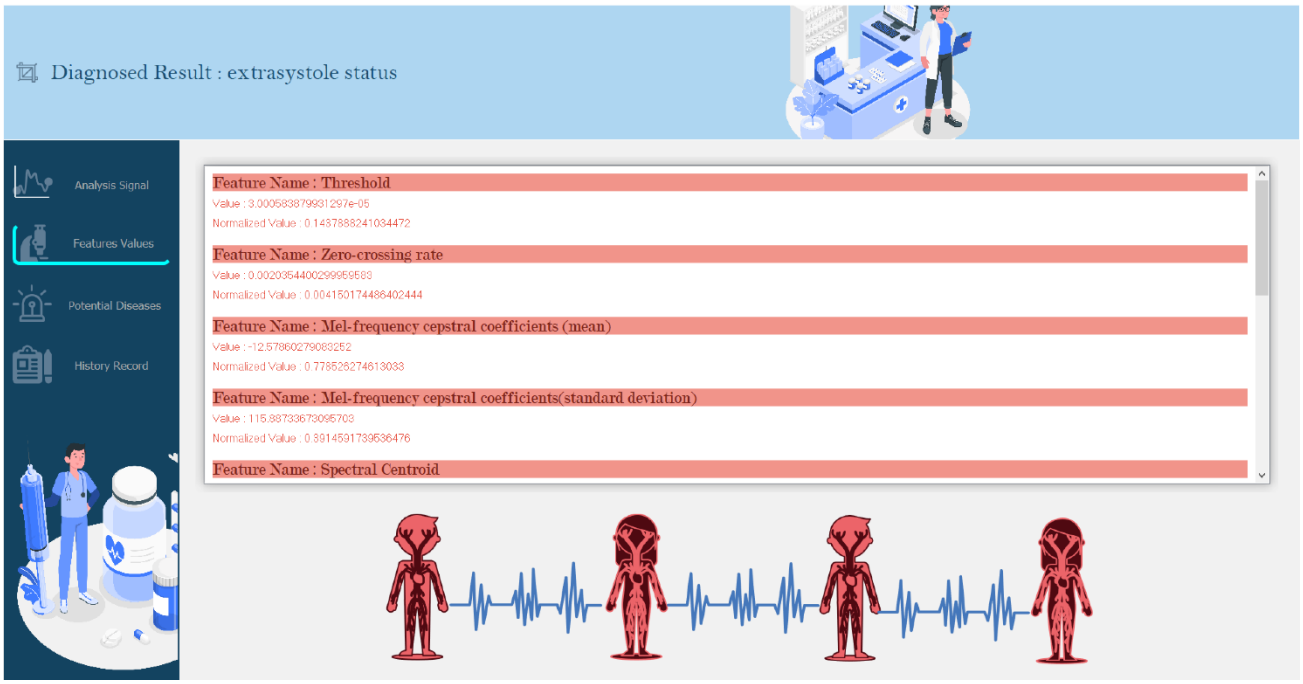


Fig. 11 The screenshot for feature extraction page

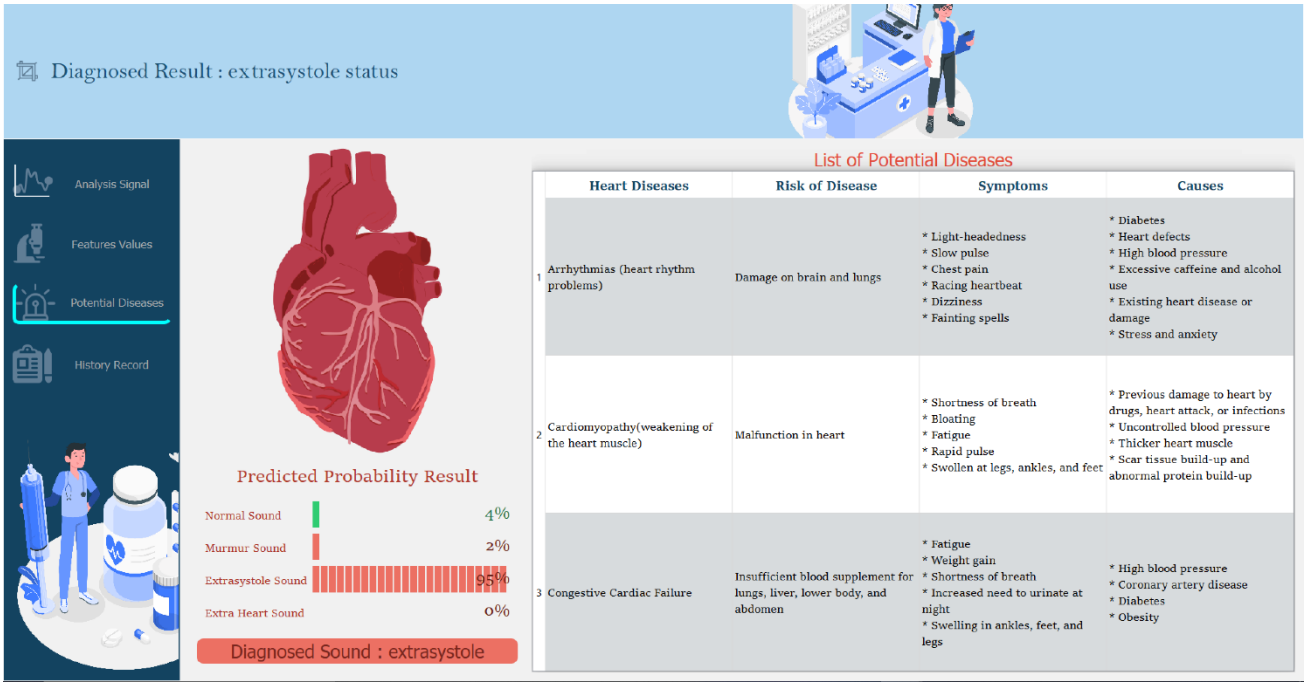


Fig. 12 The screenshot for classification page

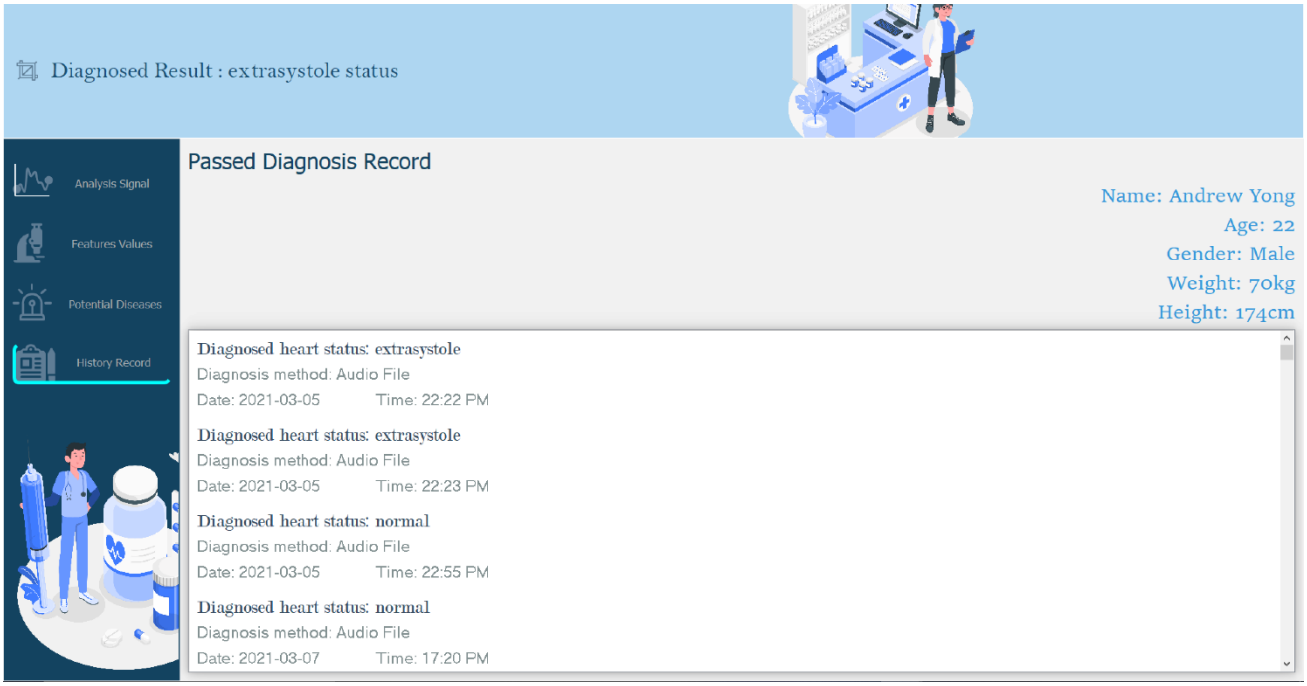


Fig. 13 The screenshot for historical record page

3.4 Discussion

In this experiment, two-dimensional (2D) convolutional neural network (CNN) is designed to classify the electrocardiograms (ECG) biomedical signals. ECG signal is divided into normal and the abnormal heart sound. Indeed, the abnormal heart sound is further categorized into murmur heart sound, extrasystole heart sound, and extra heart sound.

Shannon energy-based algorithm is utilised in detecting the peaks of the ECG signal because it is practicable for

R-peak detection. Suboh, *et al.*[32] had achieved the high sensitivity and accuracy which both were recorded above 99% in detecting the peak location by applying Shannon energy envelope. Also, the experiment by Zhu, *et al.*[33], and Beyramienanlou, *et al.*[13] also proved that Shannon energy had achieved high sensitivity and positive productivity of 99.92% in R-peak detection.

The application of the 2D CNN model is inspired by Yıldırım, *et al.*[22]. Initially, the CNN model was designed for image classification which is to solve the problem of computer vision. Nevertheless, some experiments were done to demonstrate the potential of the CNN model in classifying the ECG signal. In the experimental setup[22], implemented deep one-dimensional (1D) CNN model to recognise normal and the abnormal signal automatically without any feature extraction. The accuracy obtained for detecting the abnormal signal is 79.34%, precision rate of 79.64%, and sensitivity of 78.71%. However, the abnormal ECG detection in this experiment is divided into three distinct classes each with a higher accuracy rate as shown in Tables 4 and 5. This aids in specifying the abnormal heart sound, corresponding to its potential diseases. Apart from this, Lin, *et al.*[34] also applied 1D CNN based algorithm to extract feature and classify arrhythmia to detect the cardiac disease. Nevertheless, the experiment in this study was carried out by utilising 2D CNN in classifying the heart sound. Correspondingly, 2D CNN achieved the accuracy of 81.48% in classifying the features of four types of heart sound.

In this study, the feature is extracted from the signal to reduce the training complexity, time complexity, and improve the accuracy of classification of the model on the ECG signal. At the same time, labelling was provided to all extracted features as CNN model is the supervised machine learning model. These features are widely used in the signal and audio classification as the features can represent the signal in differentiating the types of signal.

4.0 Conclusion

In conclusion, the project has produced an application system that predicts the heart sound through the recorded heart sound audio file and AD8232 electrocardiogram (ECG) sensor. Indeed, the proposed 2D CNN model has achieved the performance at the accuracy of 81.48% in classifying the ECG signal for four types of heart sound. This shows that 2D CNN model could outperform in heart sound diagnosis. Also, the Shannon energy envelope and the threshold value are adopted in segmentation process to detect the "lub" and the "dub" sound of ECG signal. Likewise, the experiment has proven that the threshold value which included as one of the feature values do contribute to the classification of the heart sound. In addition, the hyperparameters of the CNN and kernel size can be adjusted to achieve a higher accuracy.

References

1. Kivimäki, M. & Steptoe, A. Effects of stress on the development and progression of cardiovascular disease. Nature Reviews Cardiology 15(4), 1-15 (2018). https://doi.org/10.1038/nrcardio.2017.189
2. World Health Organization. WHO, (2020). https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1
3. Wilkins, E., Wilson, L., Wickramasinghe, K., Bhatnagar, P., Leal, J., LuengoFernandez, R., Burns, R., Rayner, M. & Townsend, N. European cardiovascular disease statistics 2017. (European Heart Network, 2017).
4. Mohammadnezhad, M., Mangum, T., May, W., Lucas, J.J. & Ailson, S. Common modifiable and non-modifiable risk factors of cardiovascular disease (CVD) among Pacific countries. World Journal of Cardiovascular Surgery 6(11), 153-170 (2016).
5. Marinho, L.B., de MM Nascimento, N., Souza, J.W.M., Gurgel, M.V., Rebouças Filho, P.P. & de Albuquerque, V.H.C. A novel electrocardiogram feature extraction approach for cardiac arrhythmia classification. Future Generation Computer Systems 97, 564-577 (2019).https://doi.org/10.1016/j.future.2019.03.025
6. Li, J., Dai, W., Metze, F., Qu, S. & Das, S. A comparison of deep learning methods for environmental sound detection. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 126-130 (2017). 10.1109/ICASSP.2017.7952131
7. Zhou, Z.H. A brief introduction to weakly supervised learning. National Science Review 5(1), 44-53 (2018). https://doi.org/10.1093/nsr/ nwx106
8. Nabih-Ali, M., El-Dahshan, E.S.A. & Yahia, A.S. A review of intelligent systems for heart sound signal analysis. Journal of Medical Engineering & Technology 41(7), 553-563 (2017).https://doi.org/10.1080/03091902.2017.1382584

9. Yadav, A., Dutta, M.K., Travieso, C.M. & Alonso, J.B. Automatic Classification of Normal and Abnormal PCG Recording Heart Sound Recording Using Fourier Transform. IEEE, (2018). 10.1109/IWOBI.2018.8464131
10. Bentley, P., Nordehn, G., Coimbra, M., Mannor, S. & Getz, R. Classifying Heart Sounds Challenge. PASCAL, (2011).
11. Chakir, F., Jilbab, A., Nacir, C. & Hammouch, A. Phonocardiogram signals classification into normal heart sounds and heart murmur sounds. IEEE, 1-4 (2016). 10.1109/SITA.2016.7772311
12. Advance Physical Diagnosis Learning and Teaching at the Bedside. Demonstrations: Heart Sounds & Murmurs. University of Washington, (2021).https://depts.washington.edu/physdx/heart/demo.html
13. Beyramienanlou, H. & Lotfivand, N. Shannon's energy-based algorithm in ECG signal processing. Computational and mathematical methods in medicine 2017, 1-16 (2017). https://doi.org/10.1155/2017/8081361
14. Chen, P. & Zhang, Q. Classification of heart sounds using discrete time frequency energy feature based on S transform and the wavelet threshold denoising. Biomedical Signal Processing and Control 57, 1-11 (2019). https://doi.org/10.1016/j.bspc.2019.101684
15. Roy, J.K. & Roy, T.S. A Simple technique for heart sound detection and real time analysis. IEEE, 1-7 (2017). 10.1109/ICSensT.2017.8304 502
16. Chen, G.F. & Wu Y.D. Audio Feature Analysis for Trombone. IEEE, 71-74 (2019). 10.1109/ICECE48499.2019.9058540
17. Giannakopoulos, T. pyaudioanalysis: An open-source python library for audio signal analysis. PloS one 10(12), 1-17 (2015). https://doi.org/10.1371/journal.pone.0144610
18. Darji, M.C. Audio Signal Processing: A Review of Audio Signal Classification Features. International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2(3), 227-230 (2017).
19. Kostuchenko, E., Novokhrestova, D., Pekarskikh, S., Shelupanov, A., Nemirovich-Danchenko, M., Choynzonov, E. & Balatskaya, L. Assessment of Syllable Intelligibility Based on Convolutional Neural Networks for Speech Rehabilitation After Speech Organs Surgical Interventions. In: Salah AA, Karpov A (ed) International Conference on Speech and Computer. (Springer, 2019). https://doi.org/10.1007/978-3-030-260 61-3_37
20. Yadav, A., Singh, A., Dutta, M.K., Travieso, C.M. Machine learning-based classification of cardiac diseases from PCG recorded heart sounds. Neural Computing and Applications, 1-14 (2019). https://doi.org/10.1007/s00521-019-04547-5
21. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A. & San, TanR. A deep convolutional neural network model to classify heartbeats. Computers in biology and medicine 89, 389-396 (2017). https://doi.org/10.1016/j.compbiomed.2017.08.022
22. Yıldırım, Ö., Baloglu, U.B. & Acharya, U.R. A deep convolutional neural network model for automated identification of abnormal EEG signals. Neural Computing and Applications, 1-12 (2018). https://doi.org/10.1007/s00521-018-3889-z
23. Zeng, H., Edwards, M.D., Liu, G. & Gifford, D.K. Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics 32(12), 121-127 (2016). https://doi.org/10.1093/bioinformatics/btw255
24. Sharma, S., Sharma, S. & Athaiya, A. Activation functions in neural networks. International Journal of Engineering Applied Sciences and Technology 4(12), 310-316 (2017).
25. Dubey, A.K. & Jain, V. Comparative study of convolution neural network's relu and leaky-relu activation functions. In: Mishra S (ed) Applications of Computing, Automation and Wireless Systems in Electrical Engineering. Springer, Singapore, 873-880 (2019).
26. Wang, L., Yang, B., Chen, Y., Zhang, X. & Orchard, J. Improving Neural-Network Classifiers using Nearest Neighbor Partitioning. IEEE transactions on neural networks and learning systems 28(10), 2255-2267 (2016). 10.1109/TNNLS.2016.2580570
27. Krstinić, D., Braović, M., Šerić, L. & Božić-Štulić, D. Multi-label classifier performance evaluation with confusion matrix. Computer Science & Information Technology, 1-14 (2020). 10.5121/csit.2020.100801
28. Narváez, P., Gutierrez, S. & Percybrooks, W.S. Automatic Segmentation and Classification of Heart Sounds Using Modified Empirical Wavelet Transform and Power Features. Applied Sciences 10(14), 1-21 (2020). https://doi.org/10.3390/app10144791
29. Fayzrakhmanov, R. & Kulikov, A. Repp P. The Difference Between Precision-recall and ROC Curves for Evaluating the Performance of Credit Card Fraud Detection Models. Proceedings of International Conference on Applied Innovation in IT 6(1), 17-22 (2018).

30. Tiwari, S., Sapra, V. & Jain, A. Heartbeat sound classification using Mel-frequency cepstral coefficients and deep convolutional neural network. In: Koundai, D., Gupta, S., (ed) Advances in Computational Techniques for Biomedical Image Analysis. (Academic Press, 2020).
31. Janssens, A.C.J. & Martens, F.K. Reflection on modern methods: revisiting the area under the ROC curve. International journal of epidemiology 49(4), 1397-1403 (2020). https://doi.org/10.1093/ije/dyz274
32. Suboh, M.Z., Jaafar, R., Nayan, N.A. & Harun, N.H. Shannon Energy Application for Detection of ECG R-peak using Bandpass Filter and Stockwell Transform Methods. Advances in Electrical and Computer Engineering 20(3), 41-48 (2020). 10.4316/AECE.2020.03005
33. Zhu, H. & Dong, J. An R-peak detection method based on peaks of Shannon energy envelope. Biomed, Signal Process 8(5), 466–474 (2013). https://doi.org/10.1016/j.bspc.201 3.01.001
34. Lin, Y.J., Chuang, C.W., Yen, C.Y., Huang, S.H., Huang, P.W., Chen, J.Y. & Lee, S.Y. Artificial Intelligence of Things Wearable System for Cardiac Disease Detection. IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 67-70 (2019). 10.1109/AICAS.2019.8771630

Compliance with ethical standards

Conflict of interest There is no conflict of interest in this work.

Competing interests

The authors declare no competing interests.