Heart sounds classification using two-dimensional (2D) convolution neural network (CNN)

Andrew Yong Zheng Dao[1], Khoo Hee Kooi[2]

[1, 2]University of Wollongong Malaysia KDU Penang University College

Abstract

Heart sound signal classification using machine learning models is often an exploratory and iterative process as it might be a form of cardiac auscultation to listen to the heart sounds. In this paper, two-dimensional (2D) convolutional neural network (CNN) model is used to classify the four classes of electrocardiogram (ECG) signals, which are normal heart sound, murmur heart sound, extrasystole heart sound, and extra heart sound. A user interactive interface is developed to enable the users to diagnose the heart sound, using either audio file recording or AD8232 ECG sensor. Shannon energy is used for segmentation, two time domains, and five frequency domains are extracted as the numerical data from the segmented ECG signal for model training. The classification results of the 2D CNN is 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 predictive result of 2D CNN model.

Keywords Electrocardiogram · Convolutional Neural Network · Segmentation · Feature Extraction

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 high blood pressure, cholesterol levels, smoking, and adulthood stress are capable to trigger cardiovascular diseases (Kivimäki and Steptoe, 2017). According to World Health Organization (2020), 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 the European Union (Wilkins, *et al*., 2017). Indeed, the prevalence of cardiovascular disease in the Asia-Pacific regions was recorded at 21.1% in year 2011 and are in escalating trend (Mohammadnezhad, *et al*., 2016).

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 sounds. 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 (Marinho, *et al*., 2019).

Hear to understand the prospect of human-like sound can be performed by machine 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 (Li, *et al*., 2017). Sounds represent 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 a large amount of example which also known as training dataset (Zhou, 2018).

In the recent year, 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 (Nabih-Ali, *et al*., 2017; Yadav, *et al*., 2018).

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, and the hardware setup. 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 materials and the methods used for classifying electrocardiograms heart sounds are discussed. The pre-processing of the heart sound includes segmentation and feature extraction. Indeed, two-dimensional convolutional neural network is applied to classify four types of the heart sounds, which are normal heart sound, murmur heart sound, extrasystole heart sound, and extra heart sound.

2.1 Heart sound dataset description

The heart sound audio recording dataset is collected from PASCAL Classifying Hear Sounds Challenges, which has published by Bentley, *et al*. (2011). The dataset consists of dataset A and dataset B. Dataset A contains the audio data for four types of heart sounds, whereas dataset B contains three types of heart sounds. Dataset A consist of normal heart sound, murmur hear sounds, extra heart sounds, and artifact heart sounds, whereas dataset B contains normal heart sounds, murmur heart sound, and extrasystole heart sounds. 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*. (2016), hear sounds 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 sounds, the other heart sounds will be grouped and categorized into four files separately. Apart from this, there are three samples of the extra heart sound sample are collected from Advanced Physical Diagnosis Learning and Teaching website, which is published by University of Washington (Advance Physical Diagnosis Learning and Teaching at the Bedside, 2021).

2.2 Segmentation Process

In this paper, the segmentation process that proposed by Beyramienanlou and Lotfivand (2017) are implemented. The segmentation process 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 de-noising process using fast Fourier transform (FFT). FFT is used to visualize the frequency spectrum of the signal (Roy and Roy, 2017).

2.2.1 De-noising

The open source of Python library, Noisereduce is used to de-noise 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 smoothed 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 back the frequency to its original signal.

2.2.2 Normalization

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

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

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 (Beyramienanlou and Lotfivand, 2017). 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.

$$s[n] = -a^2[n]\log\left(a^2[n]\right) \tag{2}$$

where $a^2[n]$ 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 0.01 seconds intervals. The average Shannon energy is computed using the equation below.

$$E_s = -\frac{1}{N}\sum_{i=1}^{N} x_{norm}^2(i)\, log_{norm}^2(i) \tag{3}$$

where E_s represents the average Shannon energy of 0.01 seconds intervals, and N represents the number of the sample in the intervals (Chen and Zhang, 2020).

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 the mean and the standard deviation. The envelope decreases the signal base and placing the signal below the baseline. Equation (4) shown the calculation of the Shannon energy.

$$P(t) = \frac{E_s - \mu}{\sigma} \tag{4}$$

where μ is the mean for the random variable vector of E_s, σ represents the standard deviation of E_s, and $P(t)$ is the Shannon energy envelope which also known as normalized average Shannon energy.

2.2.6 Threshold value

A threshold value is the definition to determine 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.

$$\begin{aligned} threshold &= |k\mu(1-\sigma^2)|\ if\ \sigma < \mu, \\ threshold &= |k\sigma(1-\mu^2)|\ if\ \mu < \sigma \end{aligned} \tag{5}$$

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.

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 (Giannakopoulos, 2015). The definition of each of the sample is defined in Table 1.

Table 1. Definition for the feature (Giannakopoulos, 2015; Darji, 2017; Chen and Wu, 2019; Kostuchenko, *et al*., 2019; Yadav, *et al*., 2019).

Features name	Definition
Zero-crossing rate	The ratio of the sign change of the spectrum of an audio signal, that can be interpret 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 centre 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 transform of the signal.
Energy Entropy	The measurement of abrupt changes, the impurity of sub-frames' normalized energies.

Each feature in Table 1 is extracted as a single value, whereby the feature that extracted in the form of list is simplified into a statistical value. MFCCs and frequency features are represented by its mean and standard deviation value. The data is then saved in a comma-separated value (CSV) file. In the file, each row of data represents the features of an input audio file that contains the heart sound. Likewise, the *sound type* column represents the label of the heart sound in which 0, 1, 2, and 3 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 (Acharya, *et al.*, 2017). 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 activation function in CNN architecture (Yildirim, *et al*., 2018). Table 2 summarizes the architecture 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 (Zeng, *et al*., 2016). This is the down sampling 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 output into one-dimensional which are the vectors that can be processed by dense layer (Yildirim, *et al*., 2018). 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 (Sharma, *et al*., 2017). ReLU activation function is formulated as $f(x) = \max(0, x)$. The function graph of the ReLU activation function is illustrated in Figure 1.

Softmax activation function normally considers for the multi-class classification instead of the binary classes (Wang, *et al*., 2016; Dubey and Jain, 2019**).** 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 (Sharma, *et al*., 2017).

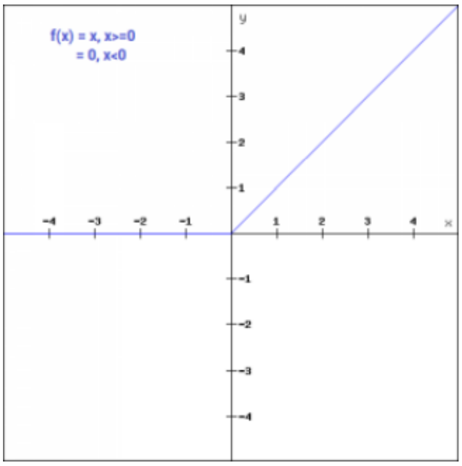


Figure 1. Graph for a ReLU activation function (Sharma, *et al*., 2017).

2.5 Electrocardiogram (ECG) sensor

AD8232 electrocardiograms (ECG) is selected as the sensor to receive the analogue signal of the heart sound. Indeed, Arduino UNO board is chosen as the micro-controller to interface with AD8232 ECG sensor. Likewise, the breadboard is used as a platform to connect the microcontroller and the sensor. Figure 2 shows the abstract view of the overall design of the hardware sensor and the system.

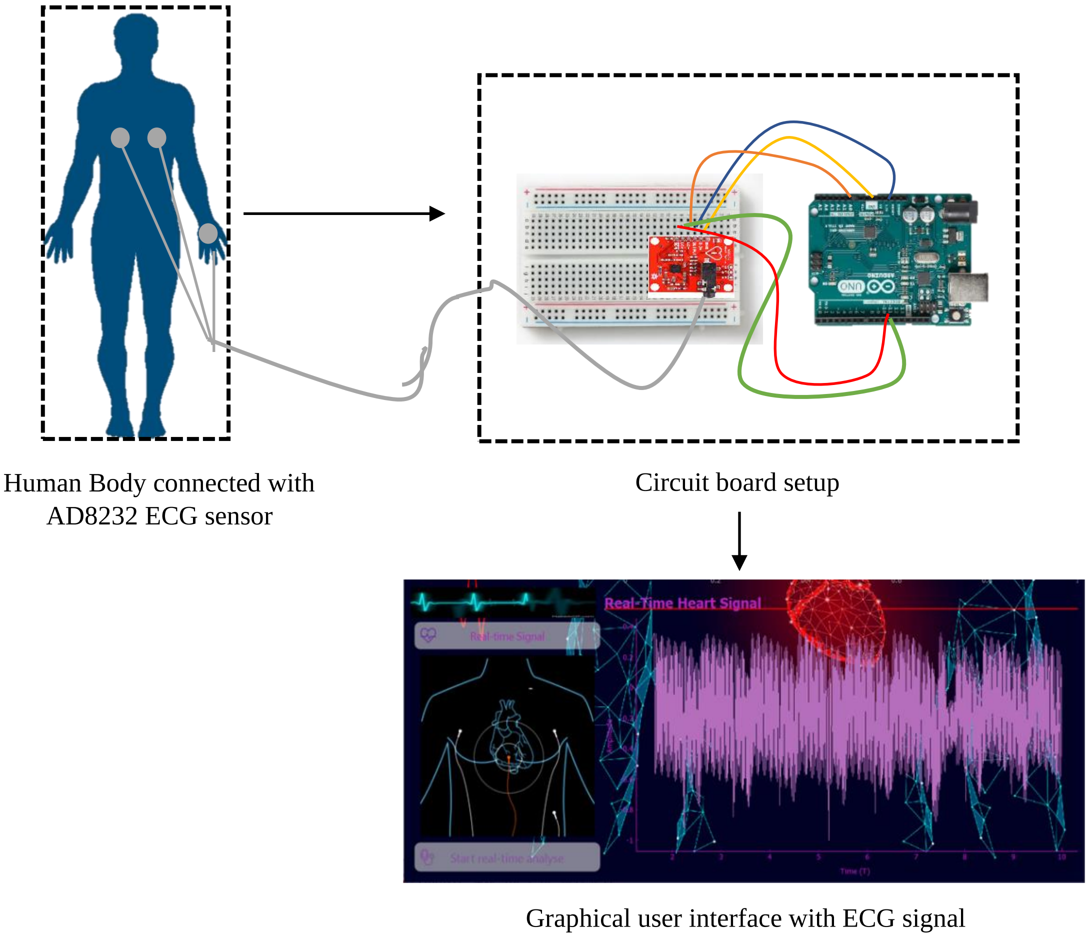


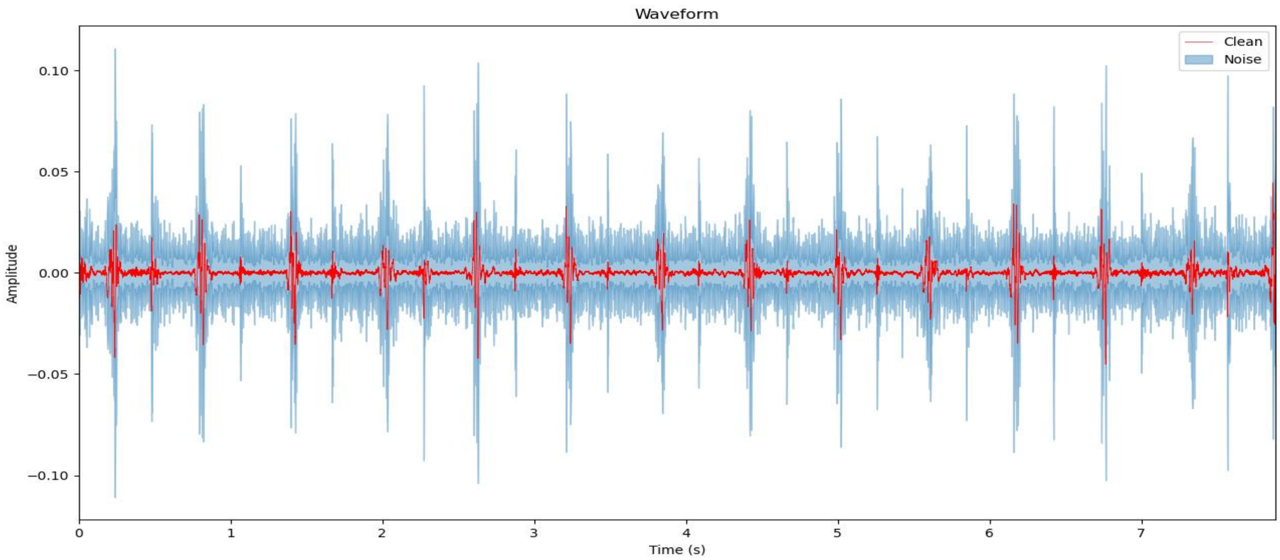
Figure 2. Abstract view of hardware setup.

3.0 Results

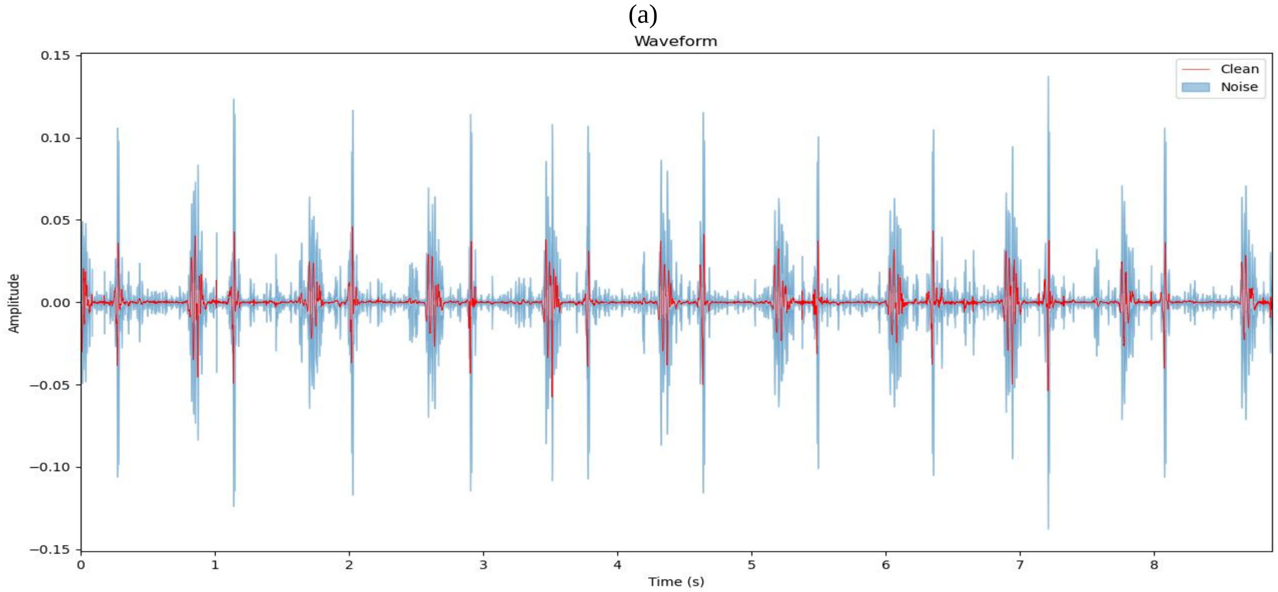
In this section, the results of the heart sound signal processing and the two-dimensional (2D) convolutional neural network (CNN) are shown. Also, the confusion matrix and the receiver operating characteristic (ROC) curve are computed and plotted to evaluate the overall classification performance of the proposed 2D CNN model, respectively.

3.1 Signal visualization

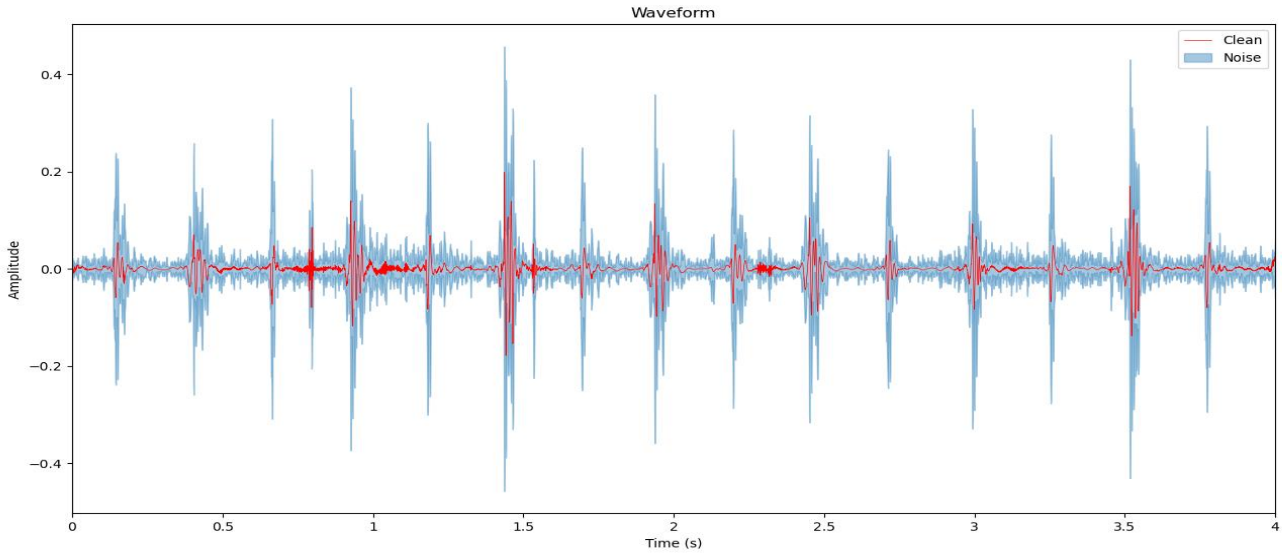
The clean signal and the background noise are separated before the application of the segmentation. The clean heart sound signals and the background noise is illustrated using the red and blue colours, respectively. The signal in red colour is passed for segmentation and feature extractions which illustrated in Figure 3. 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 Figure 4. Generally, the “lub” sound is known as $S1$, whereas “dub” sound is known as $S2$.



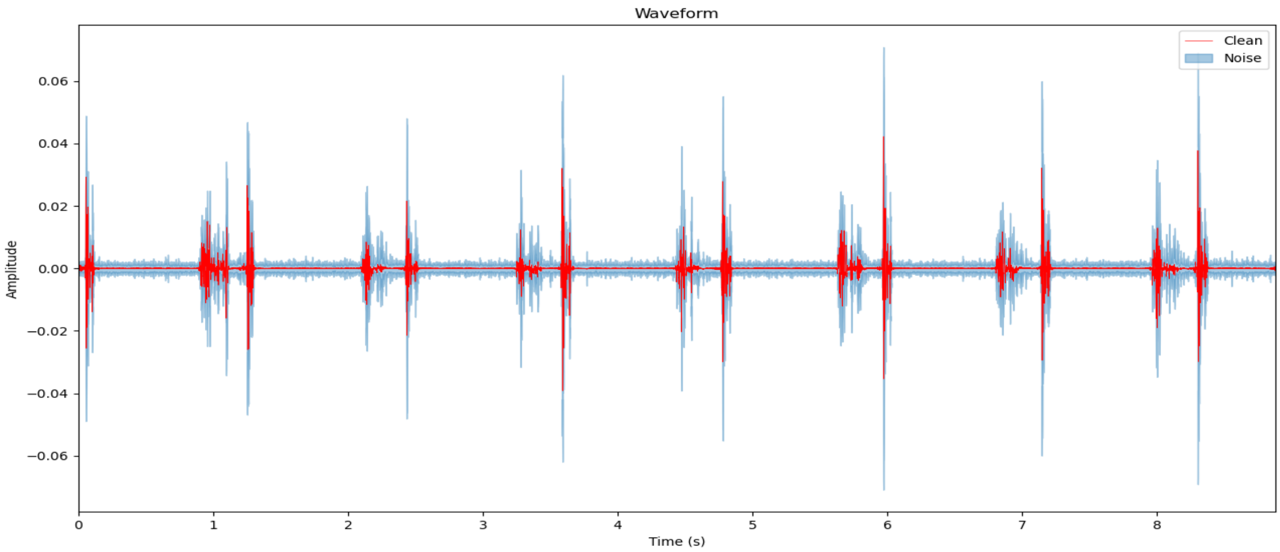
(a)



(b)

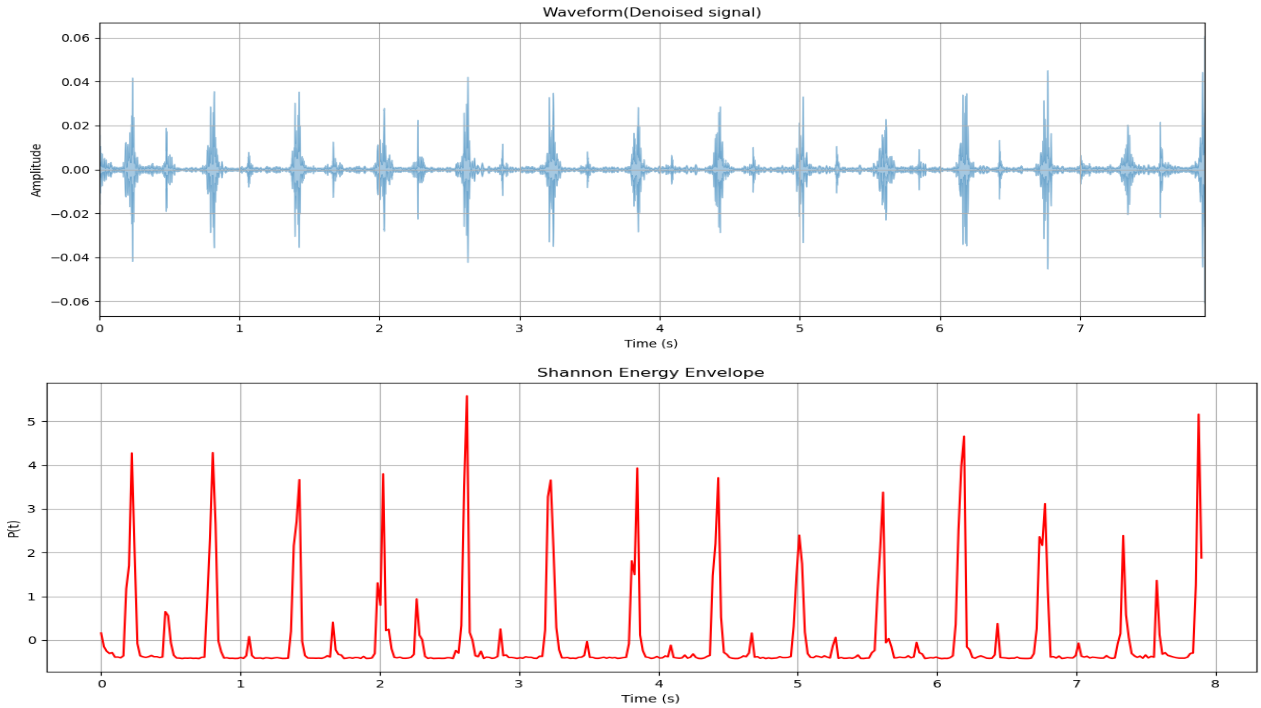


(c)

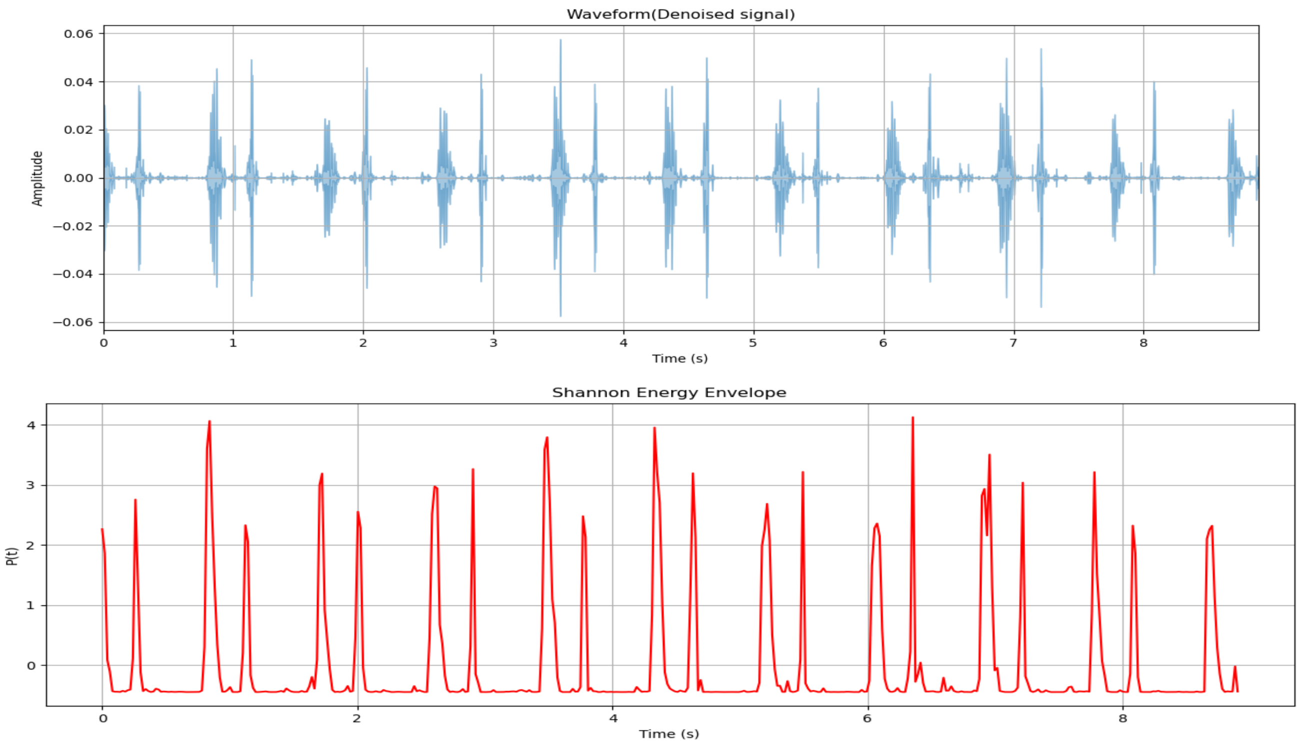


(d)

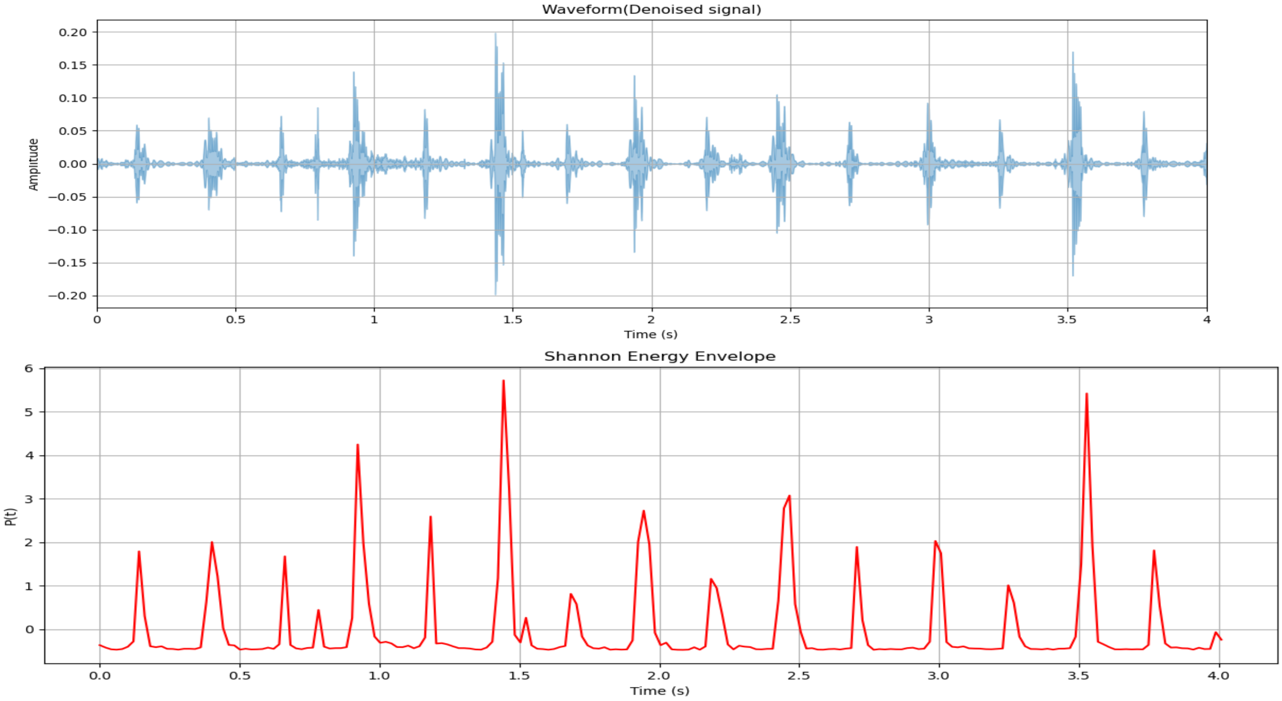
Figure 3. The visualization of the clean signals in time domain spectrum. (a) Normal heart sound. (b) Murmur heart sound. (c) Extrasystole heart sound. (d) Extra heart sound.


Waveform(Denoised signal)
Amplitude
Time (s)
Shannon Energy Envelope
P(t)
Time (s)

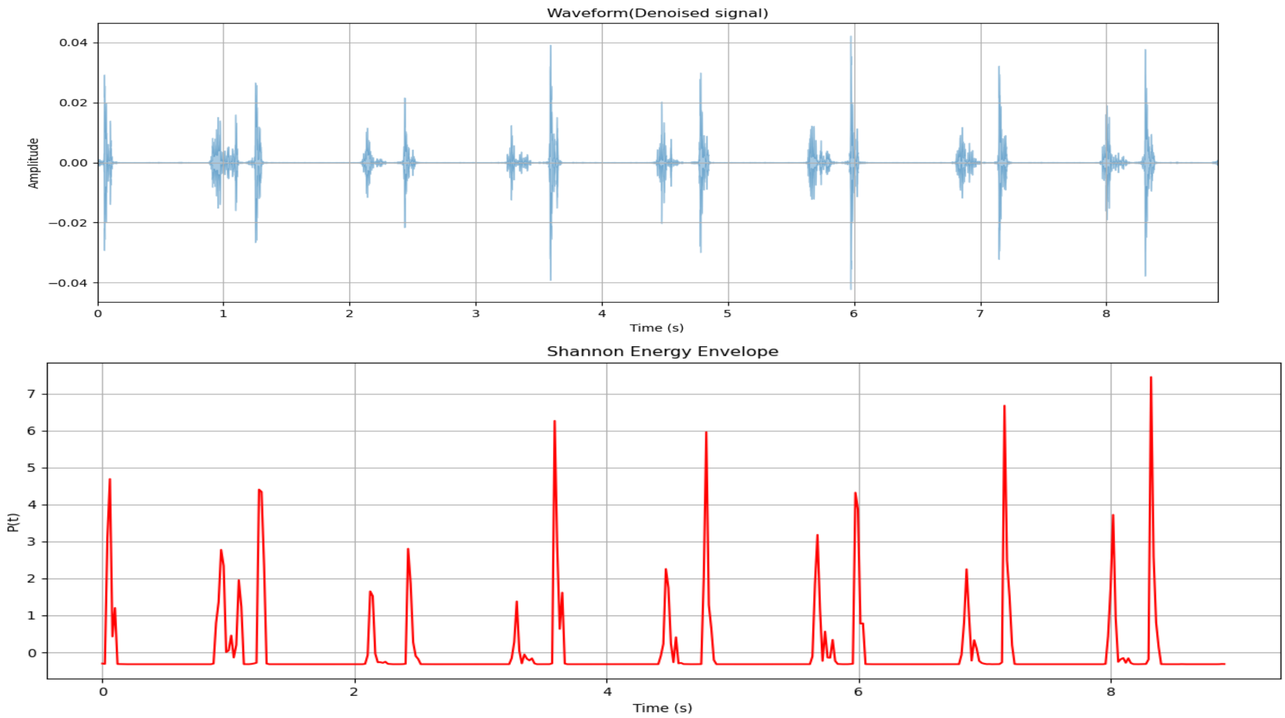
(a)


Waveform(Denoised signal)
Amplitude
Time (s)
Shannon Energy Envelope
P(t)
Time (s)

(b)



(c)



(d)

Figure 4. 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 epoch and batch size are adjusted accordingly. This experiment is to show the effect of the epoch and the batch size on the accuracy of the 2D CNN model in binary classification. 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 weight of the 2D CNN model is updated.

Throughout the first trial training of the model, the threshold value which derived while forming the Shannon energy envelope is included as one of the features to train the model in the second trial.

In the third 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. In addition, 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. The result of each trials of the 2D CNN training is shown in Table 3.

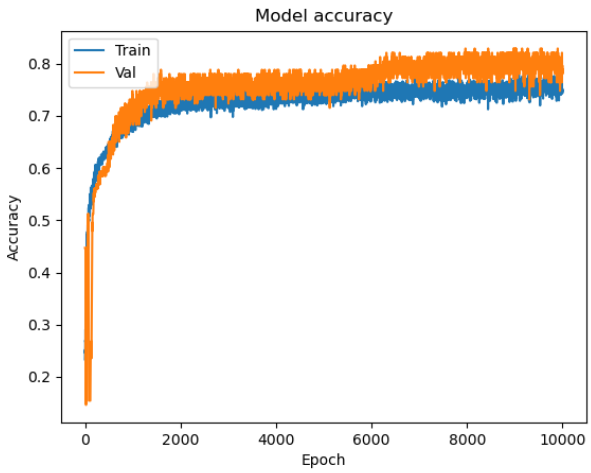
Table 3. The result of the training accuracy of each trials of the 2D CNN.

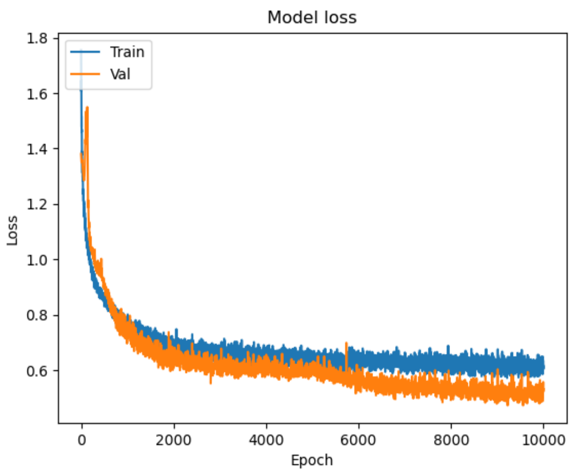
<table>
<tr><th colspan="5">First Trial</th></tr>
<tr><th>Heart Sound Types</th><th>Normal Heart Sound</th><th>Murmur Heart Sound</th><th>Extrasystole Heart sound</th><th>Extra heart sound</th></tr>
<tr><th>File samples</th><td>350</td><td>127</td><td>46</td><td>19</td></tr>
<tr><th>Epoch</th><th>Batch size</th><th colspan="2">Training Accuracy (%)</th><th>Loss</th></tr>
<tr><td>50</td><td>10</td><td colspan="2">64.3911</td><td>0.924833</td></tr>
<tr><td>100</td><td>10</td><td colspan="2">64.5756</td><td>0.938984</td></tr>
<tr><td>100</td><td>20</td><td colspan="2">64.5756</td><td>0.917870</td></tr>
<tr><td>100</td><td>50</td><td colspan="2">64.3911</td><td>0.920682</td></tr>
<tr><td>200</td><td>10</td><td colspan="2">64.2066</td><td>0.918694</td></tr>
<tr><td>1000</td><td>10</td><td colspan="2">65.1291</td><td>0.877867</td></tr>
<tr><td>1000</td><td>20</td><td colspan="2">66.6055</td><td>0.843457</td></tr>
<tr><td>10000</td><td>10</td><td colspan="2">64.5756</td><td>0.891557</td></tr>
<tr><th colspan="5">Second Trial</th></tr>
<tr><th>Heart Sound Types</th><th>Normal Heart Sound</th><th>Murmur Heart Sound</th><th>Extrasystole Heart sound</th><th>Extra heart sound</th></tr>
<tr><th>File samples</th><td>31</td><td>34</td><td>46</td><td>19</td></tr>
<tr><th>Epoch</th><th>Batch size</th><th colspan="2">Training Accuracy (%)</th><th>Loss</th></tr>
<tr><td>500</td><td>10</td><td colspan="2">52.3077</td><td>1.037885</td></tr>
<tr><td>3000</td><td>10</td><td colspan="2">47.6923</td><td>1.053638</td></tr>
<tr><td>10000</td><td>10</td><td colspan="2">35.3846</td><td>1.344266</td></tr>
<tr><th>Epoch</th><th>Batch size</th><th>Threshold value</th><th>Training Accuracy (%)</th><th>Loss</th></tr>
<tr><td>1000</td><td>10</td><td>Exclude</td><td>63.0769</td><td>0.856895</td></tr>
<tr><td>1000</td><td>10</td><td>Include</td><td>61.5385</td><td>0.799167</td></tr>
<tr><td>2000</td><td>10</td><td>Exclude</td><td>50.7692</td><td>1.044598</td></tr>
<tr><td>2000</td><td>10</td><td>Include</td><td>42.3077</td><td>1.237728</td></tr>
<tr><td>5000</td><td>10</td><td>Exclude</td><td>56.9231</td><td>0.999375</td></tr>
<tr><td>5000</td><td>10</td><td>Include</td><td>35.3846</td><td>1.343000</td></tr>
<tr><th colspan="5">Third Trial</th></tr>
<tr><th>Heart Sound Types</th><th>Normal Heart Sound</th><th>Murmur Heart Sound</th><th>Extrasystole Heart sound</th><th>Extra heart sound</th></tr>
</table>

File samples	46	46	46	22
Epoch	**Batch size**	**Threshold value**	**Training Accuracy (%)**	**Loss**
1500	10	Include	51.2500	1.044200
2000	10	Include	53.1250	1.04272
Fourth Trial				
Heart Sound Types	**Normal Heart Sound**	**Murmur Heart Sound**	**Extrasystole Heart sound**	**Extra heart sound**
File samples	350	127	598	286
Audio Augmentation				
Epoch	**Batch size**	**Threshold value**	**Training Accuracy (%)**	**Loss**
1000	100	Include	68.8006	0.712382
Data Normalization				
Epoch	**Batch size**	**Threshold value**	**Training Accuracy (%)**	**Loss**
4000	500	Include	71.6703	0.653764
4000	600	Include	74.6873	0.623295
(2 × 2) kernel size				
Epoch	**Batch size**	**Threshold**	**Training Accuracy (%)**	**Loss**
6000	600	Include	77.7370	0.572113
8000	600	Include	79.7208	0.508738
10000	600	Include	81.4842	0.455628

As shown in Table 3, the best result for the accuracy is 81.4842%. In this case, the number of epoch and batch size is 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 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 Figure 5.



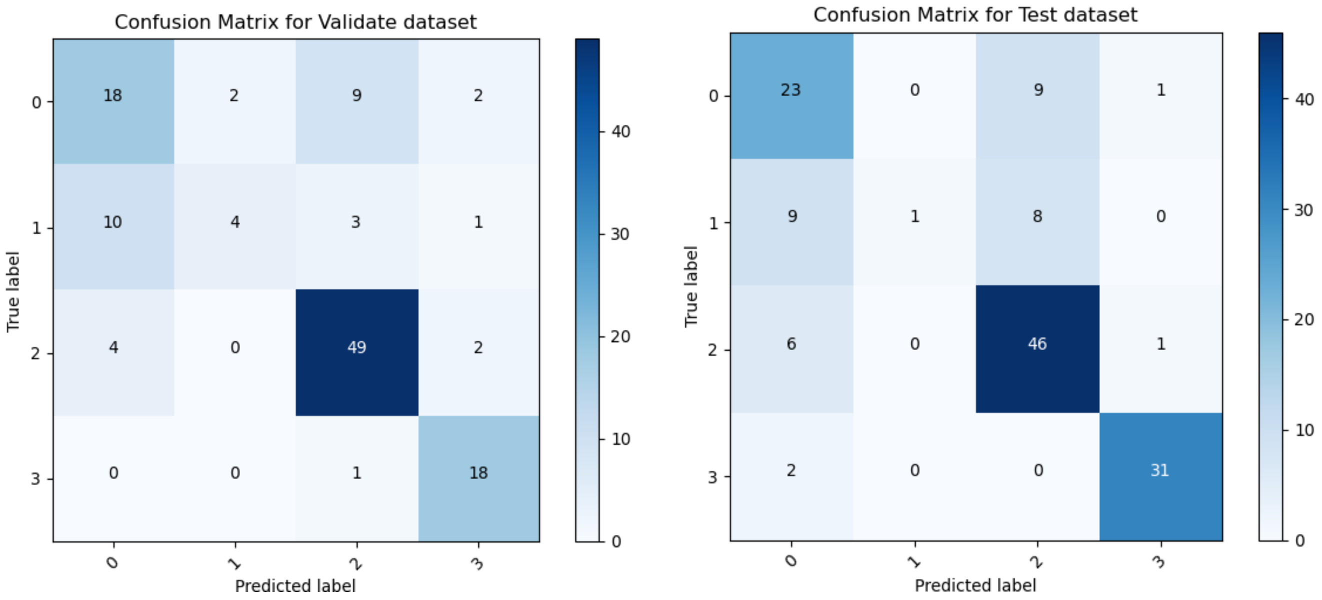


(a) **(b)**

Figure 5. 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 72.3577% and 73.7226%, respectively. While, the losses of the validate set and test set are 0.763331 and 0.758963, respectively. The confusion matrix in Figure 6 is used to further evaluate the performance of the model.



(a) (b)

Figure 6. 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 Narváez, *et al*. (2020) and Krstinić, *et al*. (2020), the formulae to calculate the accuracy and specificity, and sensitivity are shown in Equations (6) to (10) below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{6}$$

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

$$Specificity = \frac{TN}{FP + TN} \times 100 \tag{8}$$

$$Sensitivity = \frac{TP}{FN + TP} \times 100 \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{11}$$

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.

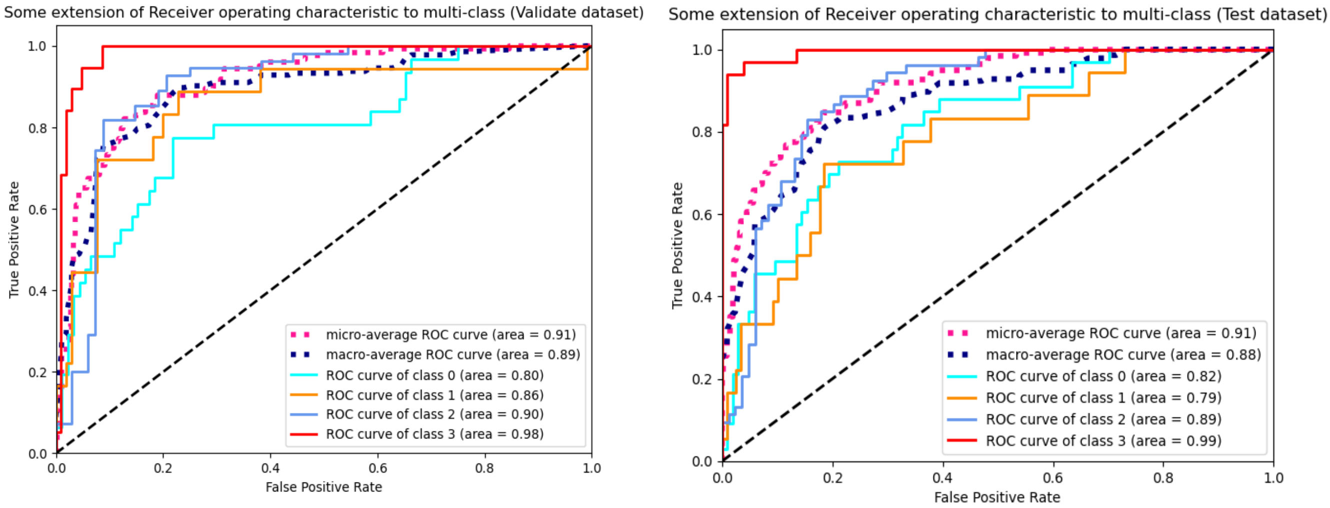
Classes	Accuracy (%)	Error rate (%)	Specificity (%)	Sensitivity (%)	Recall (%)	Precision (%)
Normal	78.049	21.951	84.783	58.065	58.065	56.250
Murmur	86.992	13.008	98.095	22.222	22.222	66.667
Extrasystole	84.553	15.447	80.882	89.091	89.091	79.032
Extra	95.123	4.878	95.192	94.737	94.737	78.261

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

Classes	Accuracy (%)	Error rate (%)	Specificity (%)	Sensitivity (%)	Recall (%)	Precision (%)
Normal	80.292	19.708	83.654	69.697	69.697	57.500
Murmur	87.591	12.409	100.000	5.556	5.556	100.000
Extrasystole	82.482	17.518	79.762	86.792	86.792	73.016
Extra	97.080	2.920	98.077	93.939	93.939	93.939

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 (FPR) and benefit (TPR) as the decision threshold varies in the proposed two-dimensional (2D) convolutional neural network (CNN) (Fayzrakhmanov, *et al*., 2018). The ROC curves for both validate dataset and test dataset is shown in Figure 7.



(a) **(b)**

Figure 7. ROC curve for the proposed 2D CNN model. (a) Validate dataset. (b) Test dataset.

In the ROC curve, the true positive rate (TPR) is plotted against the false positive rate (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, *et al*. (2018) 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 (Tiwari, *et al*., 2020). 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

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 (Norma)	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

(dotted line), and the higher the AUC value (Janssens and Martens, *et al*., 2020). The AUC value obtained from the ROC curve is tabulated in the Table 6.

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 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 sounds are further categorized into murmur heart sound, extrasystole heart sound, and extra heart sound.

The application of the 2D CNN model is inspired by Yıldırım, *et al.* (2018). In the experimental setup, Yıldırım, *et al*. 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.

In this study, the feature is extracted from the signal to reduce the training complexity and improve the classification accuracy of the model on the ECG signal. These features are widely used in the signal and audio classification as the features can represent the signal in differentiating the signal types.

4.0 Conclusion

In conclusion, the project has produced an application system that can classify the heart sound through the recorded heart sound audio file and AD8232 electrocardiogram (ECG) sensor. Indeed, the proposed two-dimensional (2D) convolutional neural network (CNN) model has achieved a high performance at the accuracy of 81.48% in classifying the ECG signal for four types of heart sound. Also, the Shannon energy envelope and the threshold value are adopted in segmentation process to detect the “lub” and the “dub” sound. Likewise, the experiment has proven that the threshold value which is derived from the mean and standard deviation of the average Shannon energy that included as one of the features values does aids in the classification of the heart sound. In addition, the hyper-parameters of the CNN and kernel size can be adjusted to achieve a higher classification accuracy.

5.0 Acknowledgement

The research work is mainly contributed by the first author's final year project (FYP) for the studies of Bachelor in Computer Science (Hons), UOW Malaysia KDU Penang University College and University of Lincoln dual-award degree program, supervised by the second author. Also, the authors would like to send gratitude to Dr. J. Joshua Thomas, the second marker, for reviewing and approving this project proposal at the initial stage.

References

Advance Physical Diagnosis Learning and Teaching at the Bedside (2021) Demonstrations: Heart Sounds & Murmurs. University of Washington. https://depts.washington.edu/physdx/heart/demo.html. Accessed 11th April 2021.

Bentley P, Nordehn G, Coimbra M, Mannor S, Getz R (2011) Classifying Heart Sounds Challenge. In: PASCAL. http://www.peterj bentley.com/heartchallenge/. Accessed 28 September 2020

Beyramienanlou H, Lotfivand N (2017) Shannon's energy-based algorithm in ECG signal processing. Computational and mathematical methods in medicine 2017:1-16. https://doi.org/10.1155/2017/8081361

Chakir F, Jilbab A, Nacir C, Hammouch A (2016) Phonocardiogram signals classification into normal heart sounds and heart murmur sounds. IEEE, 1-4. 10.1109/SITA.2016.7772311

Chen GF, Wu YD (2019) Audio Feature Analysis for Trombone. IEEE, 71-74. 10.1109/ICECE48499.2019.9058540

Chen P, Zhang Q (2020) 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. https://doi.org/10.1016/j.bspc.2019.101684

Darji, MC (2017) 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.

Dubey AK, Jain V (2019) 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, pp 873-880.

Fayzrakhmanov R, Kulikov A, Repp P (2018) 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. 10.13142/kt10006.13

Giannakopoulos T (2015) pyaudioanalysis: An open-source python library for audio signal analysis. PloS one 10(12):1-17. https://doi.org/10.1371/journal.pone.0144610

Janssens ACJ, Martens FK (2020) Reflection on modern methods: revisiting the area under the ROC curve. International journal of epidemiology 49(4):1397-1403. https://doi.org/10.1093/ije/dyz274

Kivimäki M, Steptoe A (2018) Effects of stress on the development and progression of cardiovascular disease. Nature Reviews Cardiology 15(4):1-15. https://doi.org/10.1038/nrcardio.2017.189

Kostuchenko E, Novokhrestova D, Pekarskikh S, Shelupanov A, Nemirovich-Danchenko M, Choynzonov E, Balatskaya L (2019) 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, pp 359-369.

Krstinić D, Braović M, Šerić L, Božić-Štulić D (2020) MULTI-LABEL CLASSIFIER PERFORMANCE EVALUATION WITH CONFUSION MATRIX. Computer Science & Information Technology 1-14.

Li J, Dai W, Metze F, Qu S, Das S. (2017) A comparison of deep learning methods for environmental sound detection. IEEE 126-130. 10.1109/ICASSP.2017.7952131

Marinho LB, de MM Nascimento N, Souza JWM, Gurgel MV, Rebouças Filho PP, de Albuquerque VHC (2019) A novel electrocardiogram feature extraction approach for cardiac arrhythmia classification. Future Generation Computer Systems 97:564-577. https://doi.org/10.1016/j.future.2019.03.025

Mohammadnezhad M, Mangum T, May W, Lucas JJ, Ailson S (2016) Common modifiable and non-modifiable risk factors of cardiovascular disease (CVD) among Pacific countries. World Journal of Cardiovascular Surgery 6(11):153-170. 10.4236/wjcs.2016.611022

Nabih-Ali M, El-Dahshan ESA, Yahia AS (2017) A review of intelligent systems for heart sound signal analysis. Journal of Medical Engineering & Technology 41(7):553-563. https://doi.org/10.1080/03091902.2017.1382584

Narváez P, Gutierrez S, Percybrooks WS (2020) Automatic Segmentation and Classification of Heart Sounds Using Modified Empirical Wavelet Transform and Power Features. Applied Sciences 10(14):1-21. https://doi.org/10.3390/app10144791

Tiwari S, Sapra V, Jain A (2020) 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, 7pp 115-131

Roy JK, Roy TS (2017) A Simple technique for heart sound detection and real time analysis. IEEE, 1-7. 10.1109/ICSensT.2017.8304502

Sharma S, Sharma S, Athaiya A (2017) Activation functions in neural networks. International Journal of Engineering Applied Sciences and Technology 4(12):310-316

Wang L, Yang B, Chen Y, Zhang X, Orchard J. (2016) Improving NeuralNetwork Classifiers using Nearest Neighbor Partitioning. IEEE transactions on neural networks and learning systems 28(10):2255-2267. 10.1109/TNNLS.2016.2580570

World Health Organization (2020) Cardiovascular Disease. WHO. https://www.who.int/health-topics/cardiovascular-diseases#. Accessed 20 September 2020

Wilkins E, Wilson L, Wickramasinghe K, Bhatnagar P, Leal J, LuengoFernandez R, Burns R, Rayner M, Townsend N (2017) European cardiovascular disease statistics 2017. Brussels, Belgium: European Heart Network.

Yadav A, Dutta MK, Travieso CM, Alonso JB (2018) Automatic Classification of Normal and Abnormal PCG Recording Heart Sound Recording Using Fourier Transform. IEEE. 10.1109/IWOBI.2018.8464131

Yadav A, Singh A, Dutta MK, Travieso CM (2019) Machine learning-based classification of cardiac diseases from PCG recorded heart sounds. Neural Computing and Applications 1-14. https://doi.org/10.1007/s00521-019-04547-5

Yıldırım Ö, Baloglu UB, Acharya UR (2018) A deep convolutional neural network model for automated identification of abnormal EEG signals. Neural Computing and Applications 1-12. https://doi.org/10.1007/s00521-018-3889-z

Zhou ZH (2018) A brief introduction to weakly supervised learning. National Science Review 5(1):44-53. https://doi.org/10.1093/nsr/nwx106

Zeng H, Edwards MD, Liu G, Gifford DK (2016) Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics 32(12):121-127. https://doi.org/10.1093/bioinformatics/btw255