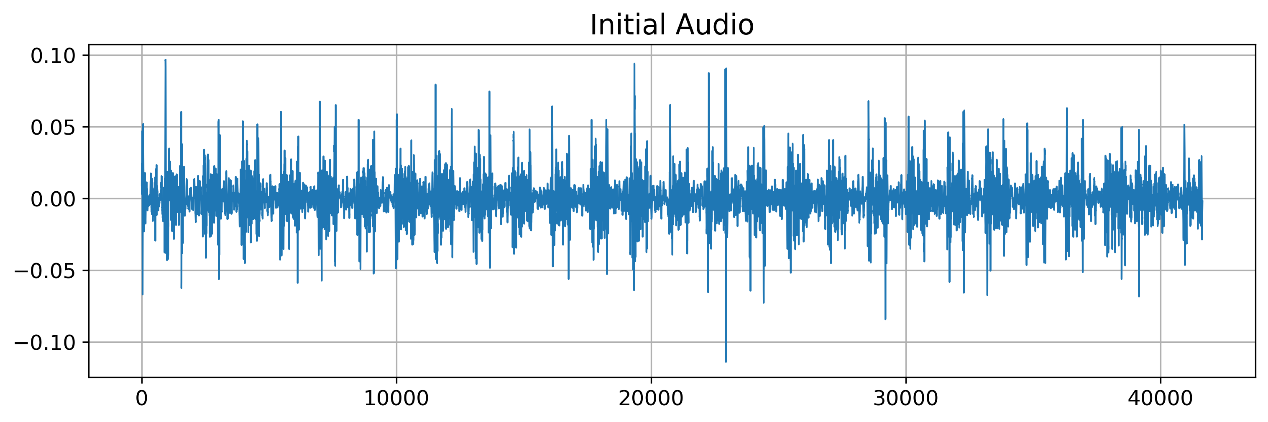
EN.520.680 Class project report

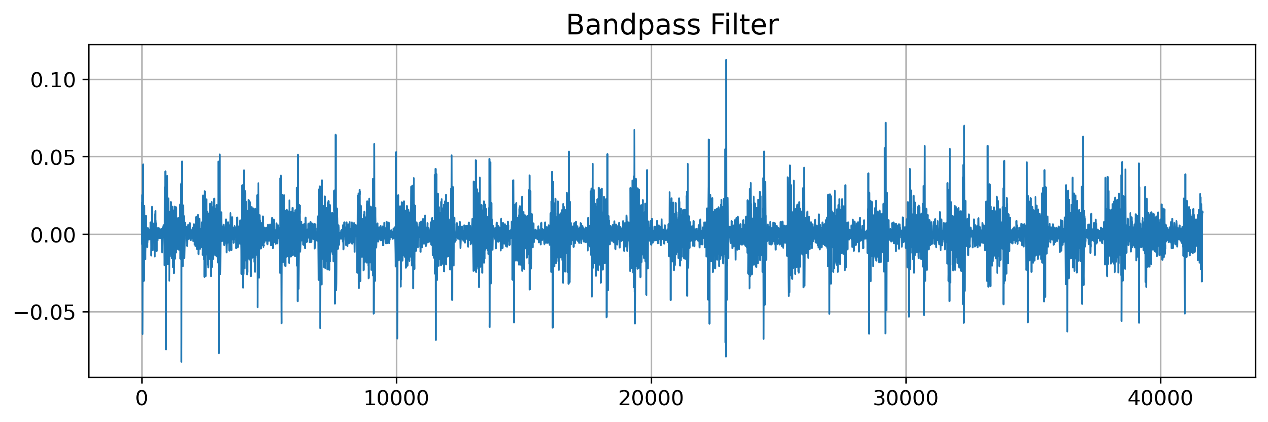
Qihua Gong

As written in the overview, my final project is an attempt to write a classifier for abnormal heart sounds based on the techniques I've learned in this class. In our last assignment, we tried single isolated word recognition under plp, dtw method. In this project, I also hope to expand to machine learning and neural networks on the basis of single recognition to achieve real audio signal recognition. In this report, I will not repeat the content written in abstract, mainly in the detailed process of how to implement the classifier.

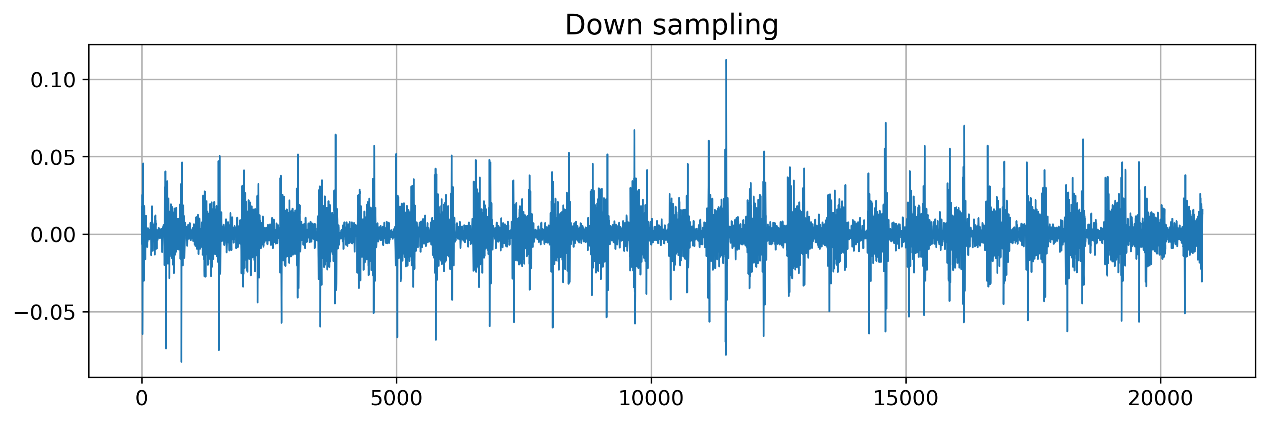
Regarding the use of the dataset, since this is just a project of a course, I don't have a lot of time to collect a large amount of data. Here, I will directly use a heart sound classification competition dataset. (https://github.com/yaseen21khan/Classification-of-Heart-Sound-Signal-Using-Multiple-Features-)The dataset contains five categories, namely one type of normal heart sounds and four types of abnormal heart sounds such as AS, MS, MR, and MVP. These four types of heart sounds are abnormal heart sounds in common heart diseases. There are about 200 raw audios in each class, I will take 50 samples in each class as testing and the rest as training data.

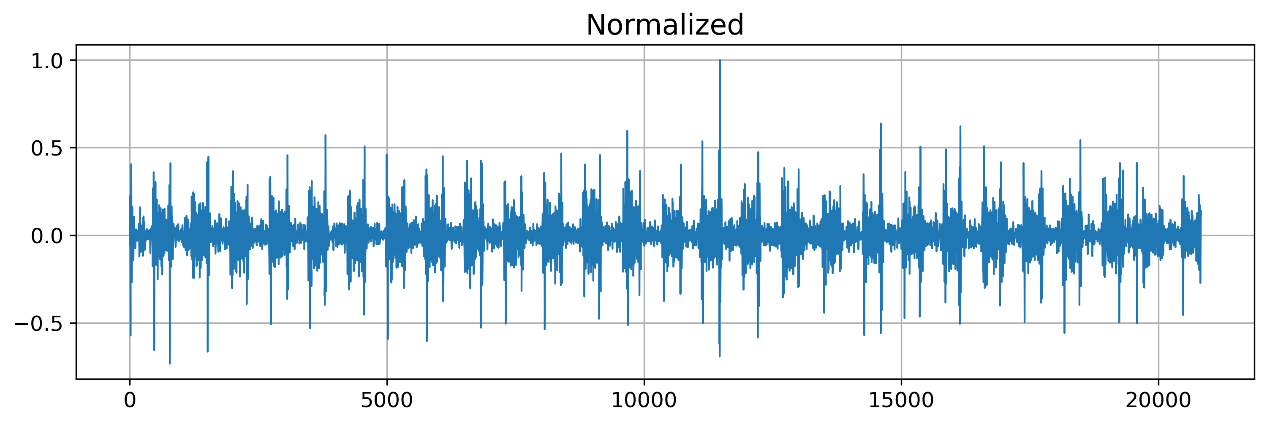
The first is audio preprocessing. In order to facilitate the computation of training and reduce training time, we need to preprocess the audio. Here, for the convenience of demonstration, I will process one audio signal and the preprocessed signal image will be printed. The first step is to read in the audio one by one, and then filter the audio. Since audio inevitably preserves some noise during production, we need to digitally filter the audio file to filter out high frequency noise as well as DC noise, while preserving the heart sound signal as much as possible. I am using the Butterworth bandpass filter here and exporting the image.



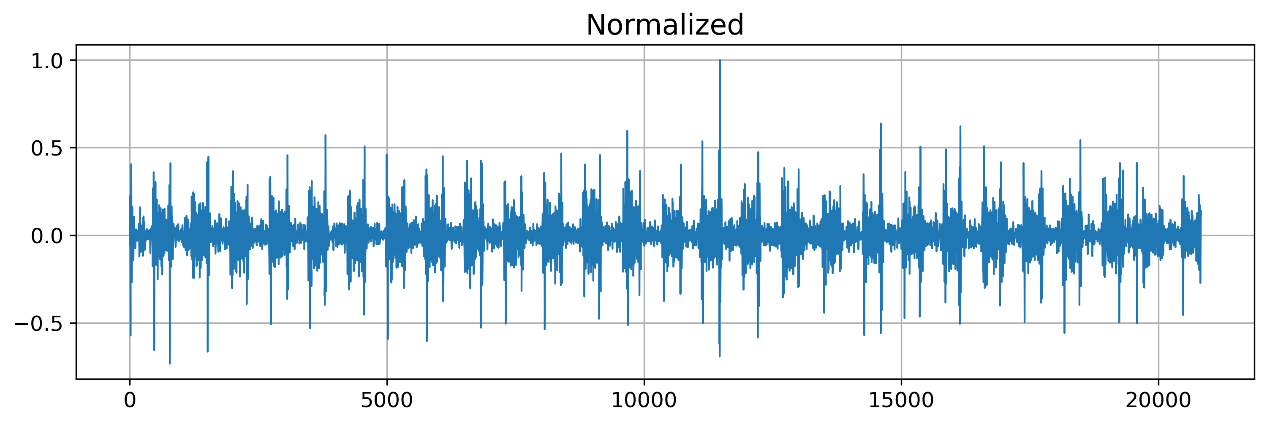


In the second step, in order to reduce the computational cost of the model, we should downsampling all the audio signals, considering that we have performed a median filter of 25-400hz on the audio, according to the Nyquist sampling law, we downsample the signal to 1000hz. Due to the large difference in scale of audio files in different datasets, we normalize all audio signals so that the range is in the interval [-1, 1].

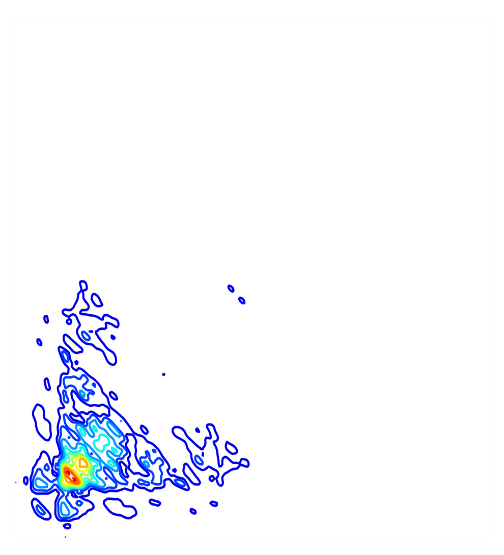
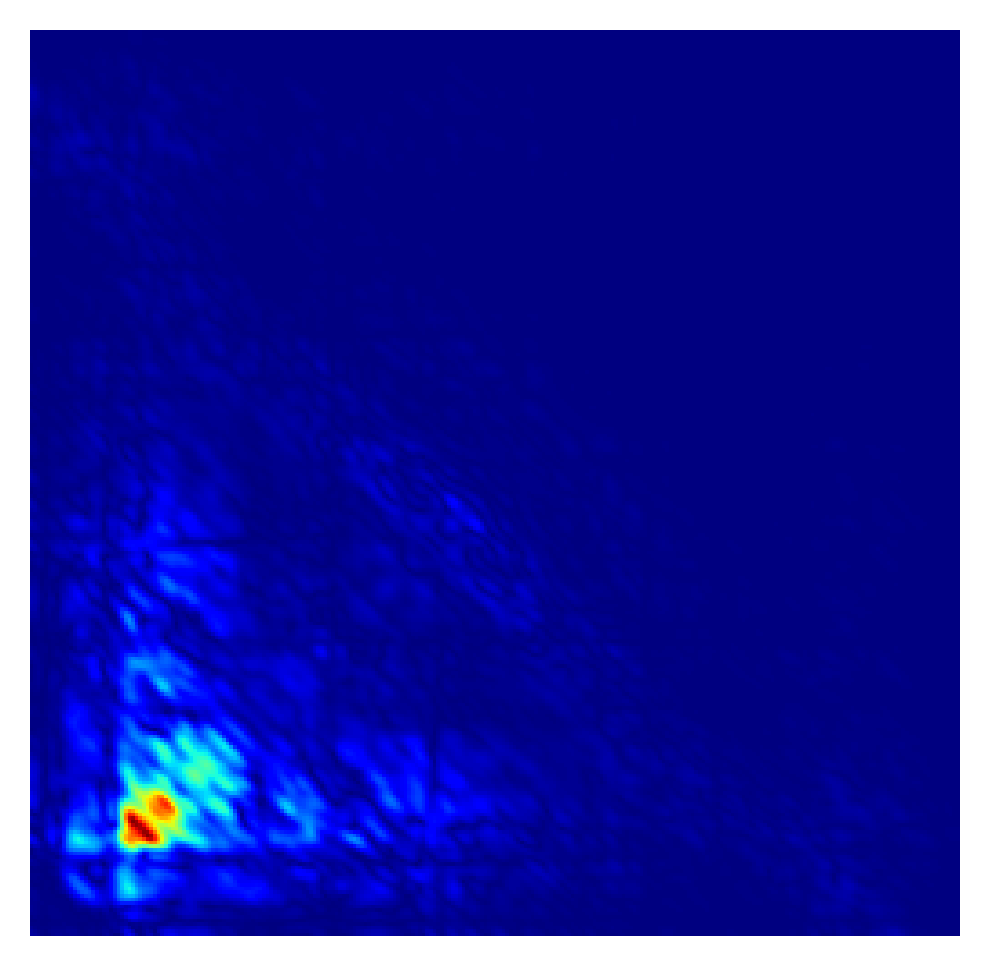




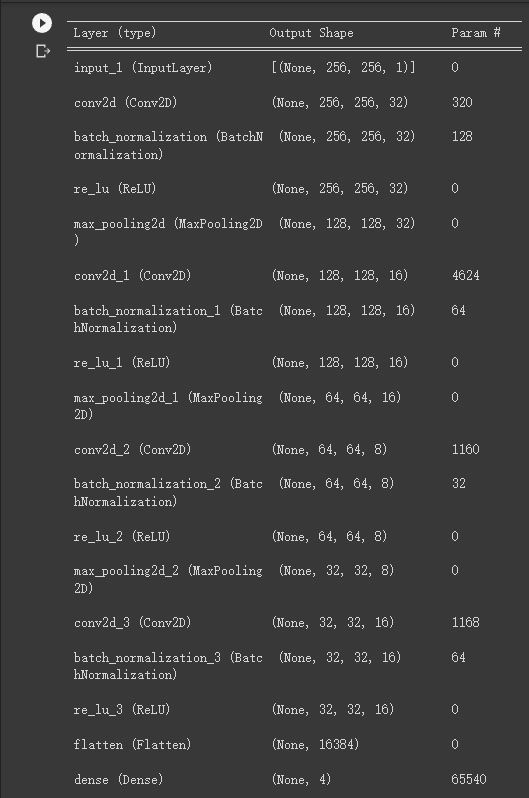
The third step is to cut the audio. In order to utilize the existing dataset as much as possible, we cut the longer audio. We cut the audio in units of 2.5s. At the same time, in order to get as much information as possible, we chose to cut with a 50% overlap. Finally, save the processed audio to four folders: AS, MS, MR, MVP.



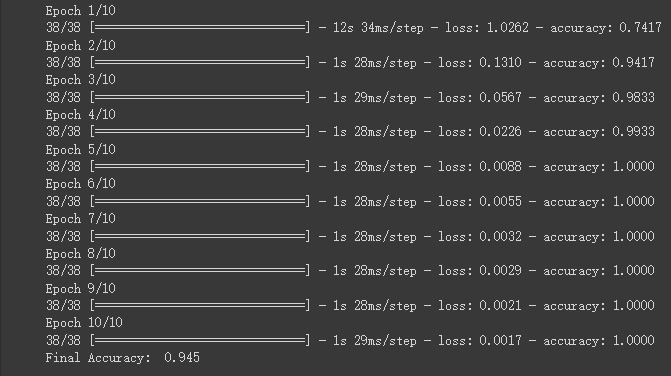
Next we need to perform feature extraction. I read some research papers on the Internet, saying that the features extracted by high-order spectral analysis methods in the field of modern digital signal processing are significantly better than the results of low-order feature extraction methods such as short-time Fourier transform and wavelet transform. In addition, the second-order spectral method among the higher-order spectral methods is the most widely used, it can well suppress the phase relationship in the signal, detect and quantify the phase coupling of non-Gaussian signals, and is often used for non-stationary medical signals, such as EEG , ECG, EMG. So here I will also use the second-order spectral analysis method for feature extraction. In other words, the analysis method is suitable for heart sound signals, and in the process of feature extraction, as many useful features as possible in the signal are retained to reduce noise. The process of feature extraction is mainly written from a python function polycoherence and polyspectrum. I still use the audio I just processed before to draw the polyspectrum feature map as a demonstration. In the actual training code, we can just apply the build in function.

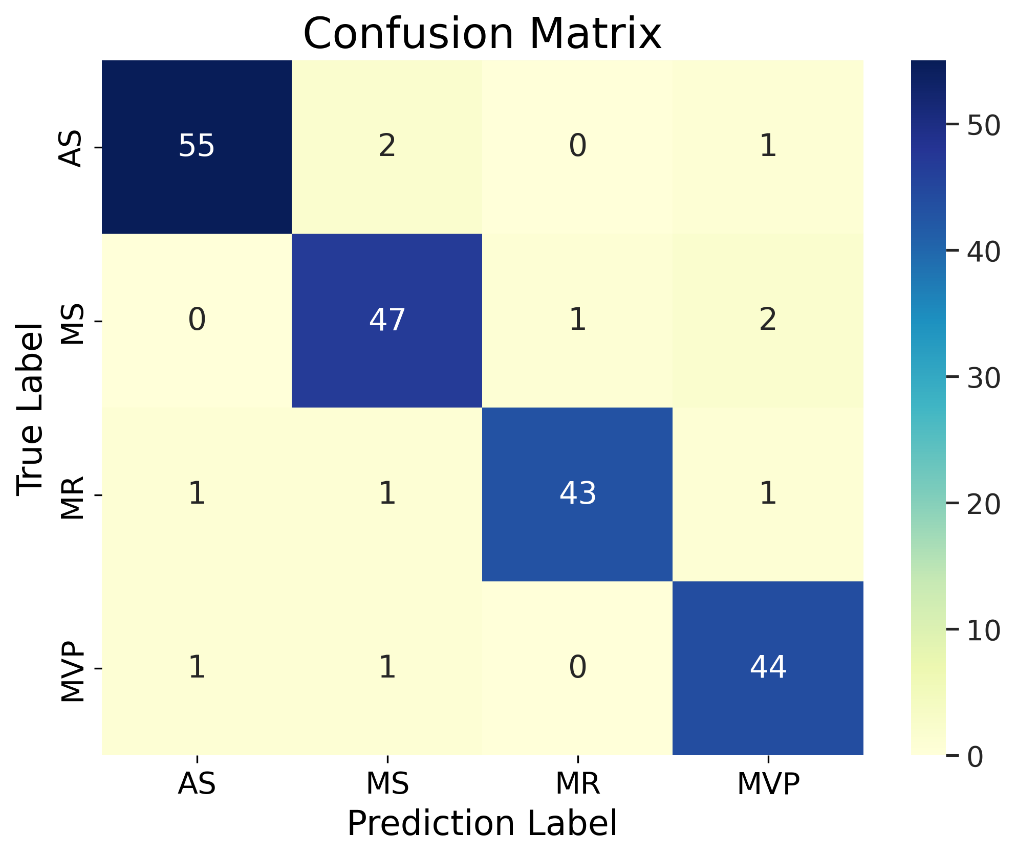


Then there is the construction of the convolutional neural network. Here I use a specific CNN network for processing audio signals. The convolutional neural network can extract the features in the second-order spectrum very well to complete the classification task, which is very suitable for our needs. The network uses a total of 4 convolutional blocks with convolutional layer, BN layer, and activation function, and finally the result is passed into the fully connected layer to obtain the final result. The specific structure is shown as follows:



Finally, we import the data into the network for training. Here we divide the 800 data into 600 training data and 200 testing data. The dataset is divided into four labels according to different classifications, and 10 epochs training is carried out. Finally, the test result accuracy rate reaches a satisfactory 92%. I draw the confusion matrix of the training results, as shown below:





In conclusion, as a project expansion after learning this course, I am very pleased to be able to achieve such a high accuracy rate at one time. Of course, there are still many areas for improvement in this project. For example, feature extraction can be more targeted without applying existing methods. The data and recognition types can also be expanded. The high recognition rate I get now may be largely due to the small number of recognition types required. Moreover, this project can also combine more hardware to achieve independent identification in the future. In short, after finishing this project, I still think it is of great significance. It opens my mind on the application of NLP technology. Thanks a lot for the guidance of the professor in this course.