Final Project Report

Team Member

1950071 Chenhang Ding

1952897 Qiyun Hu

1950061 Han Lu

Instructor

Ying Shen

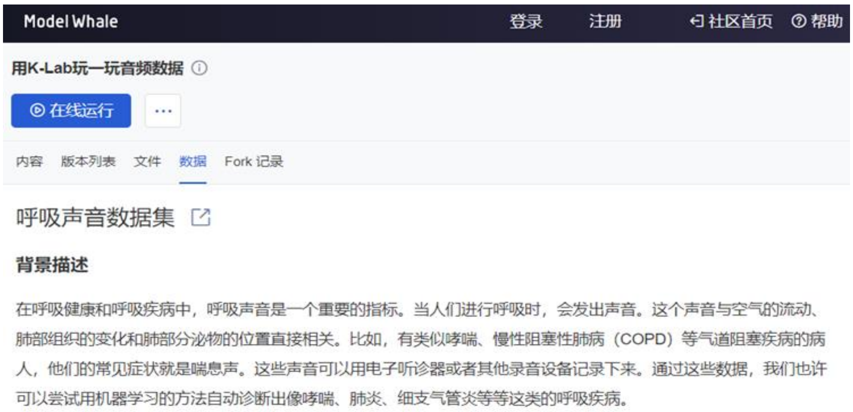
1. Project Content

In this section, we will briefly discuss the problem we want to solve, and the ideas about the resources we found that first came to our mind.

1.1 Paper Analyze

Nowadays, acoustic speech recognition has a wide range of applications, so we decide that our project should contain practical value.

Luckily, we've found a related problem background on K-Lab. The thing is that, respiratory sound can be a very useful indicator to point out the situation of a patient, so that with the help of our speech processing, related information can be extracted.



用K-Lab玩一玩音频数据

呼吸声音数据集

背景描述

在呼吸健康和呼吸疾病中，呼吸声音是一个重要的指标。当人们进行呼吸时，会发出声音。这个声音与空气的流动、肺部组织的变化和肺部分泌物的位置直接相关。比如，有类似哮喘、慢性阻塞性肺病（COPD）等气道阻塞疾病的病人，他们的常见症状就是喘息声。这些声音可以用电子听诊器或者其他录音设备记录下来。通过这些数据，我们也许可以尝试用机器学习的方法自动诊断出像哮喘、肺炎、细支气管炎等等这类的呼吸疾病。

1.2 Project Mission

After discussing about the resources we have, we set the direction for our project.

In short, we are going to solve a **multi-classification problem**. We divide all patients in the database into **four categories**: COPD, Healthy, Other and URTI. COPD and URTI are names of diseases. In fact, there are other diseases apart from COPD and URTI, but we set them together as the category of Other since the number of their samples are relatively small.

For the multi-classification problem, we will use the technique of **machine learning and deep learning**, combining what we have learned in the classroom with our practice.

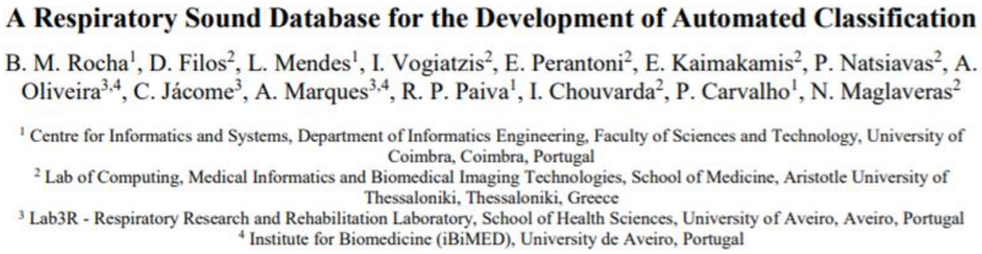
Besides, since the data includes noisy recordings, we want to solve such problem by **speech denoising**.

2. Project Survey

In this section, we'd like to show our survey on the selected topic and related method. After that, we will talk about our brainstorming of how to make full use of the dataset and our research target.

2.1 Paper Analyze Method

We consider the problem talked about above to be worth researching not only because of the valuable topic but also because of the well-prepared database and paper it brings. The paper which was published in 2018 is shown below:



A Respiratory Sound Database for the Development of Automated Classification

B. M. Rocha[1], D. Filos[2], L. Mendes[1], I. Vogiatzis[2], E. Perantoni[2], E. Kaimakamis[2], P. Natsiavas[2], A. Oliveira[3,4], C. Jácome[3], A. Marques[3,4], R. P. Paiva[1], I. Chouvarda[2], P. Carvalho[1], N. Maglaveras[2]

[1] Centre for Informatics and Systems, Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Coimbra, Portugal
[2] Lab of Computing, Medical Informatics and Biomedical Imaging Technologies, School of Medicine, Aristotle University of Thessaloniki, Thessaloniki, Greece
[3] Lab3R - Respiratory Research and Rehabilitation Laboratory, School of Health Sciences, University of Aveiro, Aveiro, Portugal
[4] Institute for Biomedicine (iBiMED), University de Aveiro, Portugal

We've also put this paper in our file. **All of our team members discuss and extract useful information about this paper:**

① A database was ultimately created. This paper hasn't put the thought of database into practice.

② The features, including crackles and wheezes, were given by experts. It means that acoustic speech features like MFCC hasn't been used, which is exactly what we are going to do.

③ Four events and six parts are included. It means that the dataset is comparatively complex, bringing difficulties to our data preprocessing.

④ Noisy sound alert. The paper straightly pointed out that data included not only clean respiratory sounds but also noisy recordings, providing authenticity to our challenge.

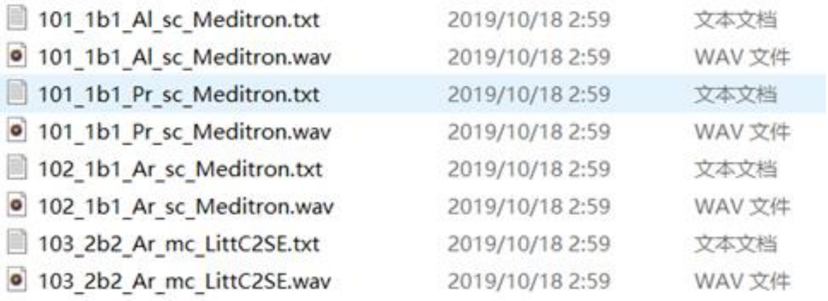
2.2 Dataset Analyze Method

To get effective information from our data set, we need to do some analysis on it so that we can choose some appropriate methods to start our project.

When we get the data set on the website, there are a lot of feature information including height, weight, gender, number of breaths, and number of pops that we need to extract. This will be in our subsequent experiments to reproduce the paper. Used, for this we extract the features and labels into a file to facilitate subsequent experiments.

An obvious problem is that each audio has different length and different breath-sound numbers, which means that it's impossible to simply use common ways and then get a unified form. We also take the loss of information during the treatment into consideration, so in addition to the usual method, we also do many different attempts.

In detail, in the step of generating MFCC, we tried to cut the dataset into audios with the same number of breath-sounds (called "trimed-audio" later), and into those with the same duration (called "same-audio" later). And to origin-audio, trimed-audio and same-audio, we calculate their MFCC, tiled MFCC, FilterBank and do some noise reduction on them. All of these processes will be mentioned later.

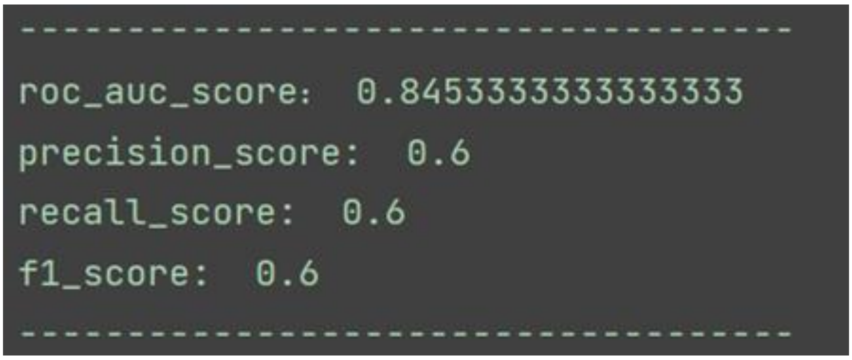


101_1b1_Al_sc_Meditron.txt	2019/10/18 2:59	文本文档
101_1b1_Al_sc_Meditron.wav	2019/10/18 2:59	WAV 文件
101_1b1_Pr_sc_Meditron.txt	2019/10/18 2:59	文本文档
101_1b1_Pr_sc_Meditron.wav	2019/10/18 2:59	WAV 文件
102_1b1_Ar_sc_Meditron.txt	2019/10/18 2:59	文本文档
102_1b1_Ar_sc_Meditron.wav	2019/10/18 2:59	WAV 文件
103_2b2_Ar_mc_LittC2SE.txt	2019/10/18 2:59	文本文档
103_2b2_Ar_mc_LittC2SE.wav	2019/10/18 2:59	WAV 文件

2.3 Research Target

We also surveyed and discussed about how to judge our work product. Moving back to the paper, we think that the best way to demonstrate our success is to get a better classification result than the method that the paper suggested.

Hence, we tried to use crackles and wheezes from the dataset as features to finish our feature engineering, and used SVM as the classifier to get the multi-classification result:



```
------------------------------------
roc_auc_score:  0.8453333333333333
precision_score:  0.6
recall_score:  0.6
f1_score:  0.6
------------------------------------
```

It is obvious that the precision score as 0.6 is not high enough, which means that after we use our own methods to complete the multi-classification and even speech denoising, the result should be higher than that.

3. Work Product

In this section, we will demonstrate the structure, the workflow and functionality of our program. We will demonstrate the process of classification and speech denoising in details with our own analysis, but our main analysis will be shown in the following sections.

3.1 Project Workflow

Our work can be divided into the following steps：

① Data preprocessing（Includes audio noise reduction）

② Feature engineering (including ordinary MFCC, compressed MFCC, FBank, GMM)

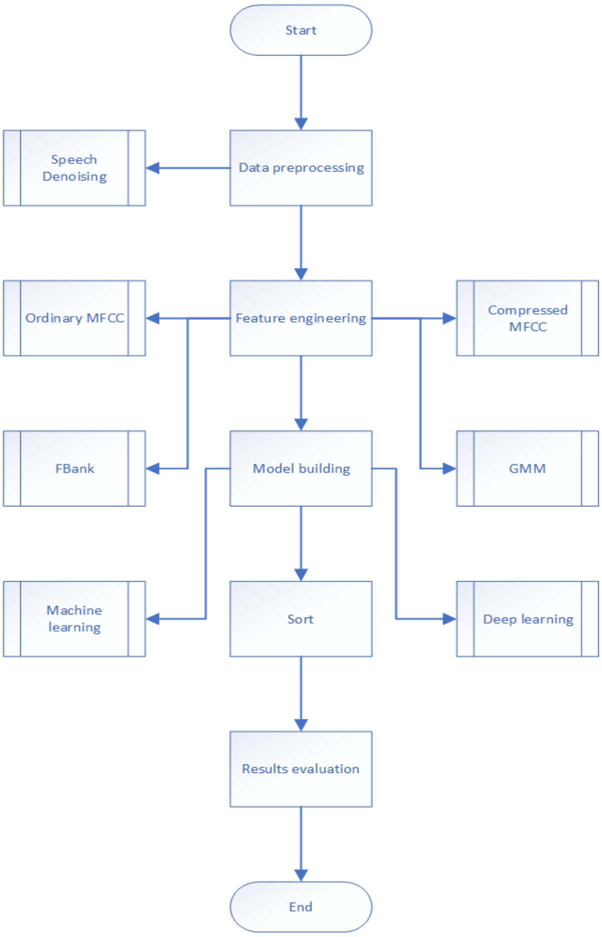
③ Model building

a. Machine learning, select classifiers (including SVM and LDA)

b. Deep learning, neural network construction

④ Classification

⑤ Results evaluation and collation



3.1.1 Data Preprocessing

There is a total of 920 audio data, but after our observations, we found that the duration of each audio is different, some are 20 seconds long, and some are 1 minute long. For extracting mfcc features, such unprocessed data is obviously not acceptable. After thinking and discussing with the teacher, we finally decided to select the smallest common breathing cycle for each audio, that is, 2 breathing cycles as our input and put it into the subsequent data feature processing.

Our training set and test set are divided randomly according to 4:1

3.1.2 Feature Engineering

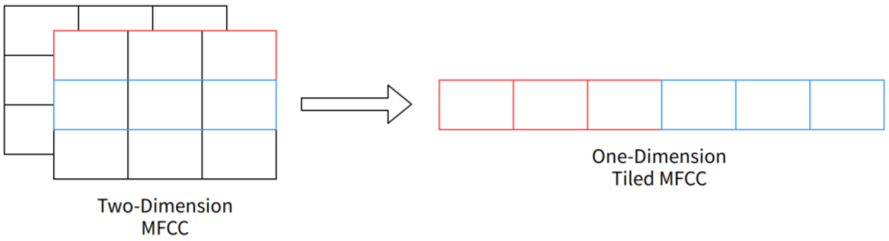
a) Common MFCC

As a very classic feature, and going to be part of the course this semester, MFCC is the first feature appearing in our mind. We calculate MFCC of each audio in the data set as the first of our project, which includes pre-emphasis, windowing, DFT, Mel-Filterbank, and so on. For we have learned it in our class, no more talk here about common MFCC.

But it's worth mentioning that to avoid the problem of too long data, we intercept first 2 breath sounds from each audio. Another advantage is that it meets the state requirement of HMM, which will be mentioned next.

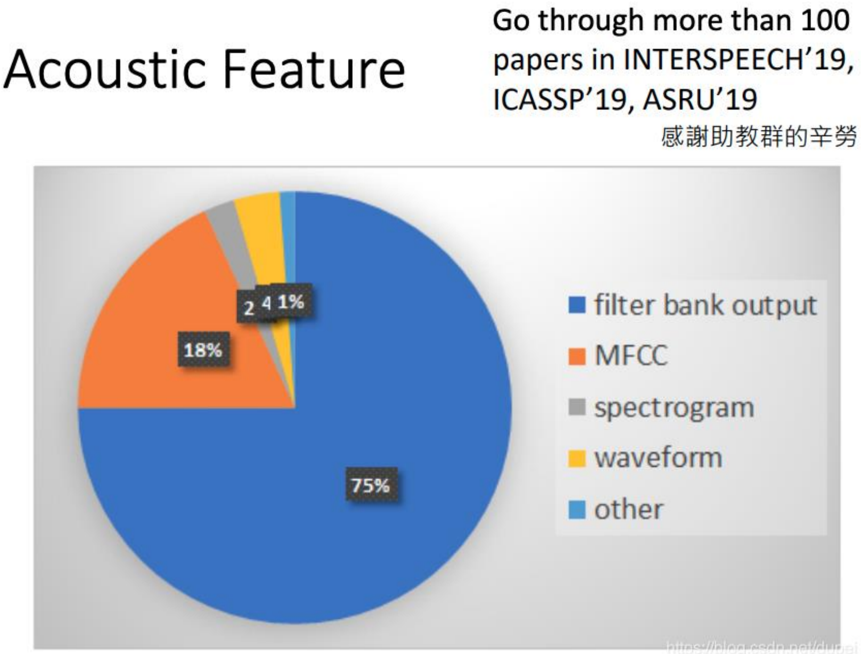
b) Tiled MFCC

Though we get common MFCC, it is hard to apply it to most classifiers. To use more methods for comparison, we need to convert it into a unified form. We tried a simplest method: reshape all the common MFCC in the same shape and then tiled them "in a line", which means that convert them into a one-dimensional vector. Such treatment can make MFCC able to be used in a lot of classifiers, but the question is that some information will be thrown away. This may lead to an inaccurate classification result.



c) Filterbank

In **Tiled MFCC**, we talked about information loss. In fact, even the common MFCC has the same problem, for it just reserves part of the whole information. After searching on the Internet for some self-learning, we found that, surprisingly, most people tend to choose FBank, also known as Filterbank, instead of MFCC as our feature.



Acoustic Feature

Go through more than 100 papers in INTERSPEECH’19, ICASSP’19, ASRU’19

感謝助教群的辛勞

We can analyze the difference between FBank and MFCC by paying attention to the step between them, which is DCT. DCT maps signals into a lower dimensional space. This step removes the correlation between signals. In other words, there could be correlation information in FBank, but we remove them to fit the need of conventional machine learning. For example, the hidden states in HMM are better to be independent without correlations. Besides, the calculation in GMM doesn't need such correlation.

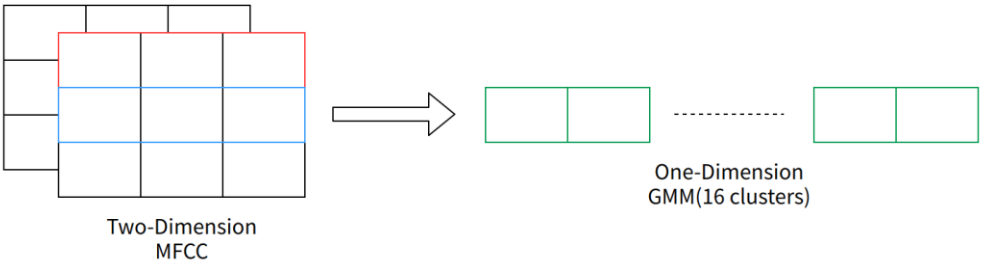
However, nowadays with the development of deep learning, whether to choose FBank and MFCC has become a topic worth discussing again. An increasing number of people are thinking that as correlation between signals is valuable in DNN and CNN, FBank is a better option.

We have been inspired by this relatively heated topic, so we decide to use FBank as our feature in deep learning.

d) GMM

Through GMM, we can easily reflect the two-dimension MFCC features into one-dimension features. Every element in the one-dimension feature vector represents the possibility that one sample is in the corresponding cluster. In this way, such a one-dimension feature vector can be used to construct a feature matrix.

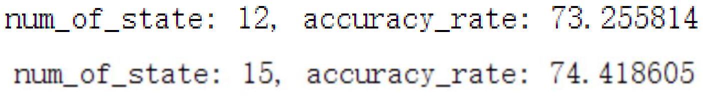
I have to mention that in our project, although there are only four patients 'situation categories, I don't consider it to be reasonable enough because it would make the classifier seemingly worthless. That's why I choose to set 16 clusters, as there are four respiratory events based on the four categories.



3.1.3 Conventional Machine Learning

a) HMM

This semester, we learned the application of HMM model in speech recognition, which makes analysis according to the hidden state of audio. So, we first used this model for classification in this project. We tried models with 12 and 15 states respectively, and the result is:



```
num_of_state: 12, accuracy_rate: 73.255814
num_of_state: 15, accuracy_rate: 74.418605
```

For a four-class problem, that's actually acceptable. But what we want to achieve is far beyond that, so we try many other methods.

b) SVM

SVM is a classic classifier for traditional machine learning. Using SVM, our aim is to find a hyperplane to partition the sample, whose principle is interval maximization. This is ultimately transformed into a convex quadratic programming problem.

Generally speaking, SVM is used to solve a binary classification problem, but we can make some slight changes to fit our need for multi-classification. We can use the thought of OneVsRestClassifier, which is also a package in sklearn. It gave every category a classifier, in this case SVM, for binary classification, and transform the binary classification result into multi-classification.

c) LDA

Apart from SVM, LDA is another common classifier to solve classification problems, and some people say that it is a simple version of SVM.

The principle of LDA is that labeled data is projected into a lower dimension and divided into different clusters in the meantime.

The other reason why I choose LDA as the classifier is that according to some research, although nowadays deep learning is very popular in voiceprint recognition, LDA or PLDA is still welcomed. It is because our features in voiceprint or related projects are relatively small within-class variance, large between-class distance.

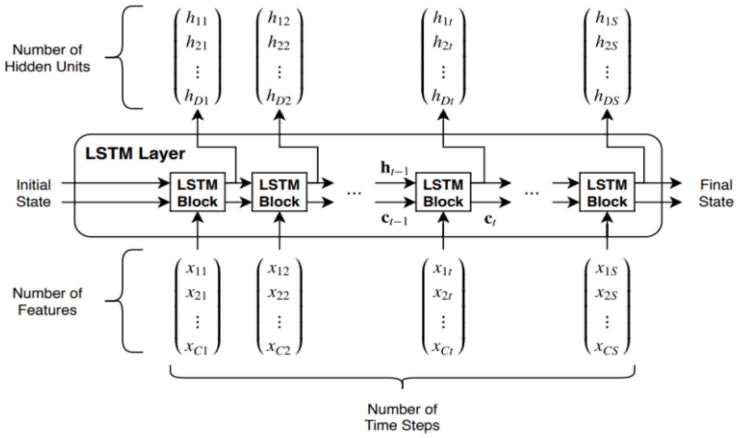
3.1.4 Deep Learning

Through our literature research, we found that compared with mfcc features, filterbank features can extract more features as the input of lstm, so we try to use the lstm model and its deformation in deep learning to process the input to solve the classification problem

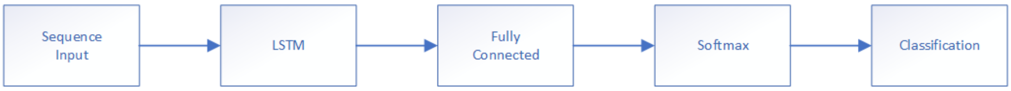
The LSTM network is a recurrent neural network (RNN) that can learn the long-term dependencies between the time steps of sequence data.

The core components of the LSTM network are the sequence input layer and the LSTM layer. The sequence input layer inputs sequence or time series data into the network. The LSTM layer learns the long-term correlation between the time steps of the sequence data.

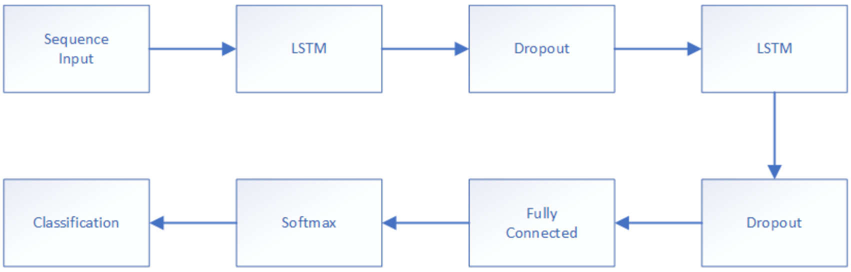
The following figure illustrates the flow of time sequence X with C features (channels) of length S through the LSTM layer.


Number of Hidden Units
h_{11} h_{21} ⋮ h_{D1}
h_{12} h_{22} ⋮ h_{D2}
h_{1t} h_{2t} ⋮ h_{Dt}
h_{1S} h_{2S} ⋮ h_{DS}
LSTM Layer
Initial State
LSTM Block
LSTM Block
...
$\mathbf{h}_{t-1}$
LSTM Block
...
LSTM Block
Final State
$\mathbf{c}_{t-1}$
$\mathbf{c}_t$
Number of Features
x_{11} x_{21} ⋮ x_{C1}
x_{12} x_{22} ⋮ x_{C2}
x_{1t} x_{2t} ⋮ x_{Ct}
x_{1S} x_{2S} ⋮ x_{CS}
Number of Time Steps

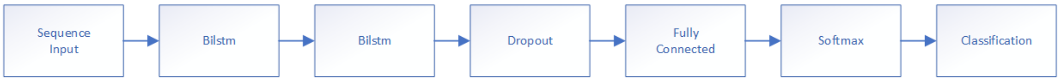
The following figure illustrates the architecture of a simple LSTM network used for classification. The network starts with a sequence input layer, followed by an LSTM layer. In order to predict class labels, the end of the network is a fully connected layer, a softmax layer and a classification output layer.



In this neural network, insert an additional LSTM layer with output mode'sequence' before the LSTM layer to increase the depth of the LSTM network. To prevent overfitting, a discard layer can be inserted after the LSTM layer.



Specify the input size as a sequence of size 26. Specify a hidden bidirectional LSTM layer with an output size of 125, and output the sequence. Then, specify a bidirectional LSTM layer with an output size of 100 and output the last element of the sequence. This command instructs the bidirectional LSTM layer to map its inputs to features and then prepares for output to the fully connected layer. Finally, 4 classes are specified by including a fully connected layer of size 4, followed by a softmax layer and a classification layer.



3.2 Speech Denoising

After listening to the audio of the data set, we found that most of the audios had some background noise, so we thought: in order to make the final classification more accurate, can we do noise reduction for these audios, and then reduce or even eliminate the impact of noise on the final result? Therefore, we have tried two ways of speech noise reduction: spectral subtraction and deep learning.

3.2.1 Spectral Subtraction

Spectral subtraction is a noise reduction method based on the audio signal itself.

Its basic principle is to estimate the spectrum of noise through part of the audio signal (usually the first few frames), and then subtract the noise from the original audio, so as to obtain the final noise reduction frequency. The main formula used in spectral subtraction is:

$$let\ D(w) = P_s(w) - \alpha P_n(w)$$

$$P_s'(w) = \begin{cases} D(w), if\ D(w) > \beta P_n(w) \\ \beta P_n(w), otherwise \end{cases}$$

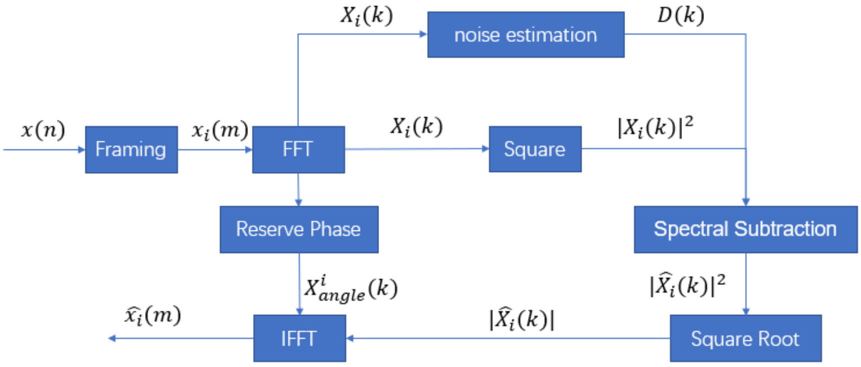
Where $P_s'(w)$ represents the output signal; α is called subtractive factor, which needs to be greater than 1 to ensure the denoising effect; β is used to calibrate the lower limit of the speech signal, which can make the music noise less obvious.

As for the specific value of the two, referring to some materials on the Internet, our α uses the following function:

$$\alpha = \begin{cases} 5(SNR < -5) \\ -0.15 * SNR + 4(-5 \leq SNR < 20) \\ 1(SNR \geq 20) \end{cases}$$

And we let β=0.02.

In detail, the program first uses FFT to convert the original audio signal to the frequency domain signal, then takes the first 5 frames as the noise reference, obtains the estimated frequency domain of the noise by means of taking the mean. And finally uses the above formula to calculate the audio signal after the noise reduction. The picture below shows the specific process:

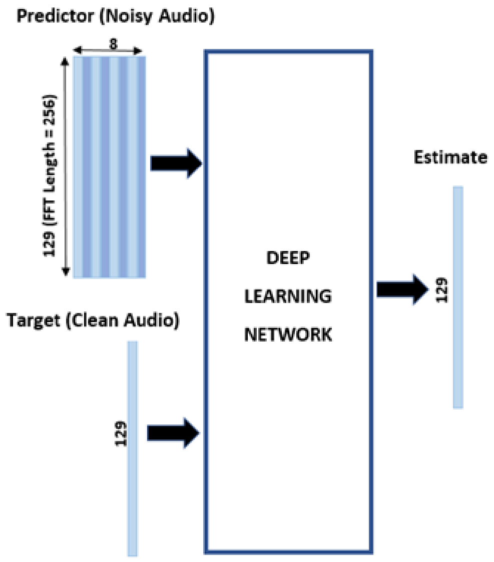


3.2.2 Neural Network

In **Spectral Subtraction**, it does little effect on most of the audios. We figured out that the thing they had in common was that they are started with a pure piece of noise and, obviously, it is unrealistic for us to capture the piece of noise for each audio, so we tried to use deep learning to denoise the sound.

The program used here refers to the official documentation of MATLAB, we changed some code so that we could use it to meet our requirements. In the program, we try to use two-layer fully connected neural network and convolutional neural network to reduce noise.

It is worth mentioning that the input of this network is 8*129 and the output is 1*129. 129 represents the spectral vector, which means that each output is actually computed by the current and the previous 7 noisy vectors.



(From the official documentation of MATLAB)

In order to make a comparison, a three-layer fully connected neural network is also used in the actual process, but there is little difference between the final results and the two-layer fully connected neural network results. Therefore, only the two-layer fully connected neural network results are shown below.

4. Result Evaluation

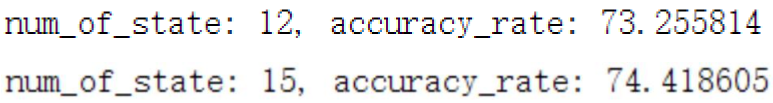
In this section, we will display the performance evaluation since we have already talked about all our working process. Of course, we will arrange our result in order, and in this section we will analyze at length about our result.

4.1 Classification Result

4.2.1 Conventional Machine Learning

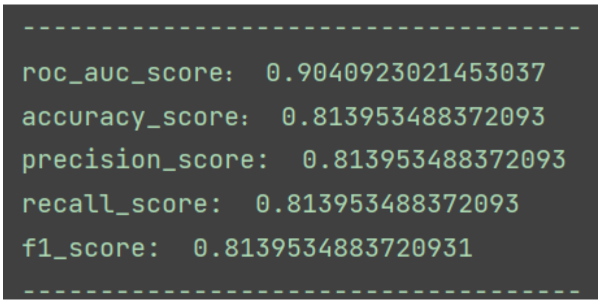
For this project, I did experiments for the SVM, LDA classifier and HMM model as mentioned above with different types of feature engineering, and the result is shown below:

HMM:



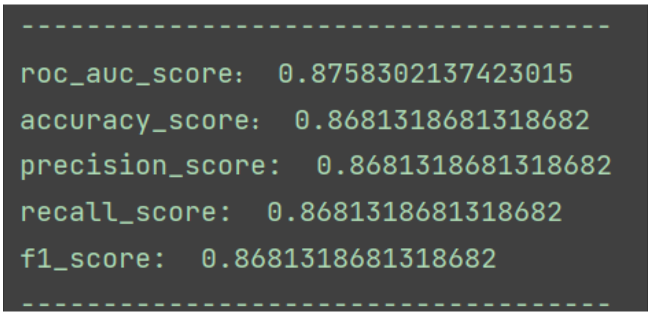
```
num_of_state: 12, accuracy_rate: 73.255814
num_of_state: 15, accuracy_rate: 74.418605
```

TiledMFCC+SVM:



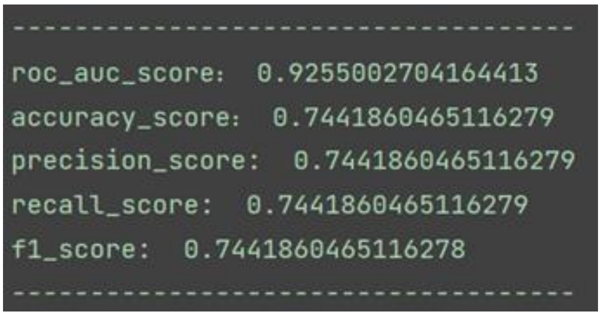
```
------------------------------------
roc_auc_score:  0.9040923021453037
accuracy_score:  0.813953488372093
precision_score:  0.813953488372093
recall_score:  0.813953488372093
f1_score:  0.8139534883720931
------------------------------------
```

GMM+SVM:



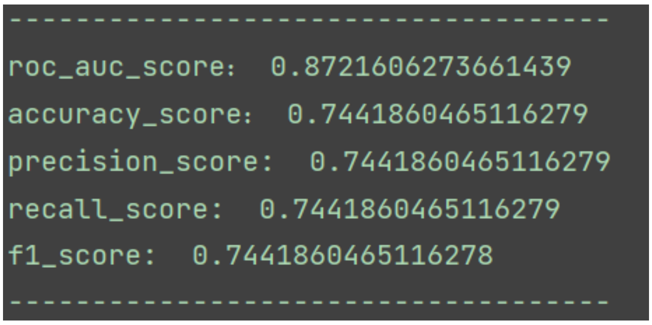
```
------------------------------------
roc_auc_score:  0.8758302137423015
accuracy_score:  0.8681318681318682
precision_score:  0.8681318681318682
recall_score:  0.8681318681318682
f1_score:  0.8681318681318682
------------------------------------
```

TiledMFCC+LDA:



```
------------------------------------
roc_auc_score:  0.9255002704164413
accuracy_score:  0.7441860465116279
precision_score:  0.7441860465116279
recall_score:  0.7441860465116279
f1_score:  0.7441860465116278
------------------------------------
```

GMM+LDA:

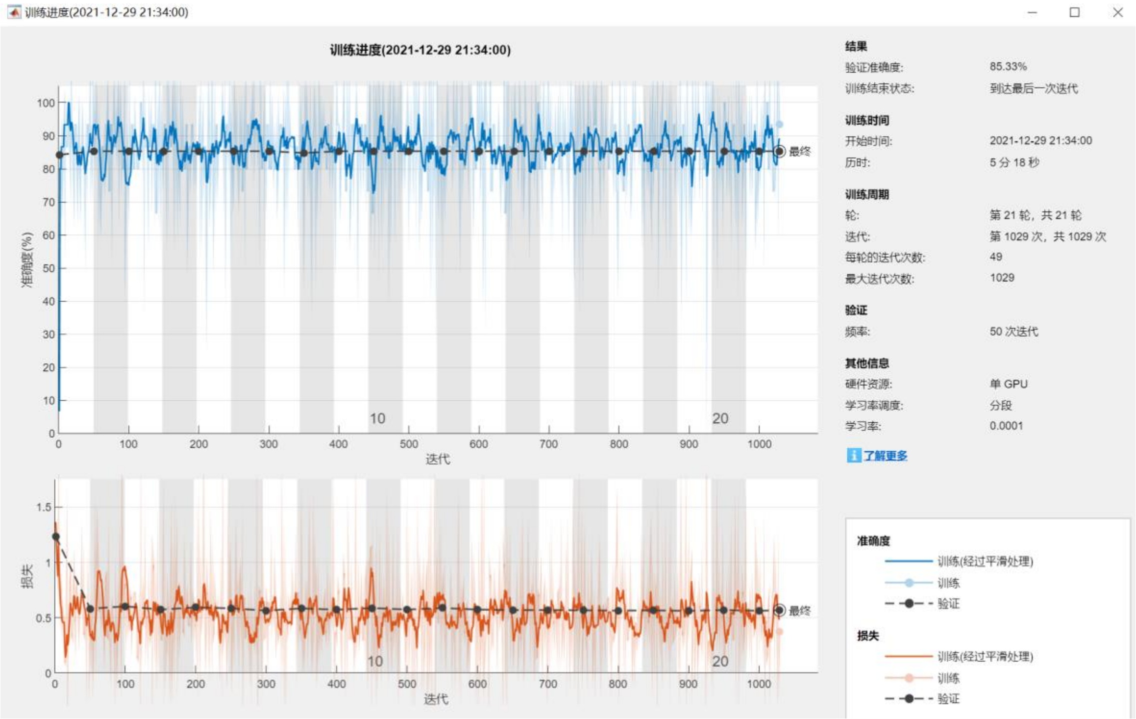


```
------------------------------------
roc_auc_score:  0.8721606273661439
accuracy_score:  0.7441860465116279
precision_score:  0.7441860465116279
recall_score:  0.7441860465116279
f1_score:  0.7441860465116278
------------------------------------
```

4.2.2 Deep Learning

We use three different deep learning nerves to classify the same mfcc and filterbank input. The classification results are shown in the figure below.

Lstm



结果

验证准确度: 85.33%

训练结束状态: 到达最后一次迭代

训练时间

开始时间: 2021-12-29 21:34:00

历时: 5 分 18 秒

训练周期

轮: 第 21 轮，共 21 轮

迭代: 第 1029 次，共 1029 次

每轮的迭代次数: 49

最大迭代次数: 1029

验证

频率: 50 次迭代

其他信息

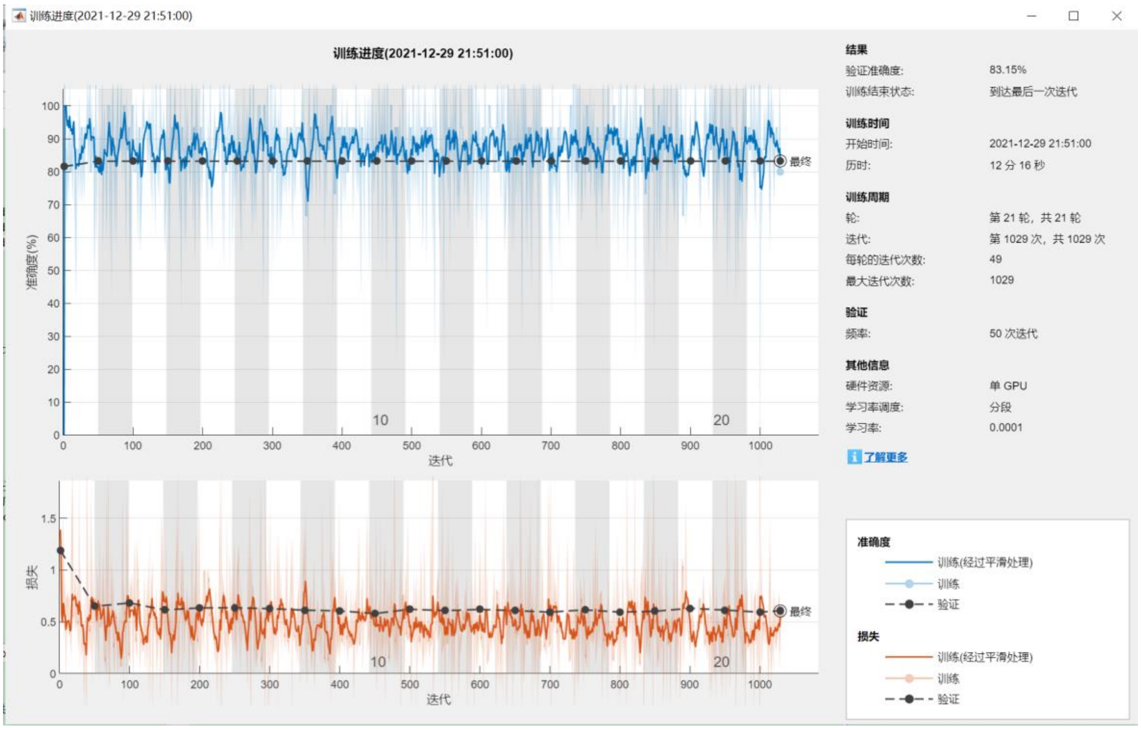
硬件资源: 单 GPU

学习率调度: 分段

学习率: 0.0001

了解更多

Deeper lstm



结果

验证准确度: 83.15%

训练结束状态: 到达最后一次迭代

训练时间

开始时间: 2021-12-29 21:51:00

历时: 12 分 16 秒

训练周期

轮: 第 21 轮，共 21 轮

迭代: 第 1029 次，共 1029 次

每轮的迭代次数: 49

最大迭代次数: 1029

验证

频率: 50 次迭代

其他信息

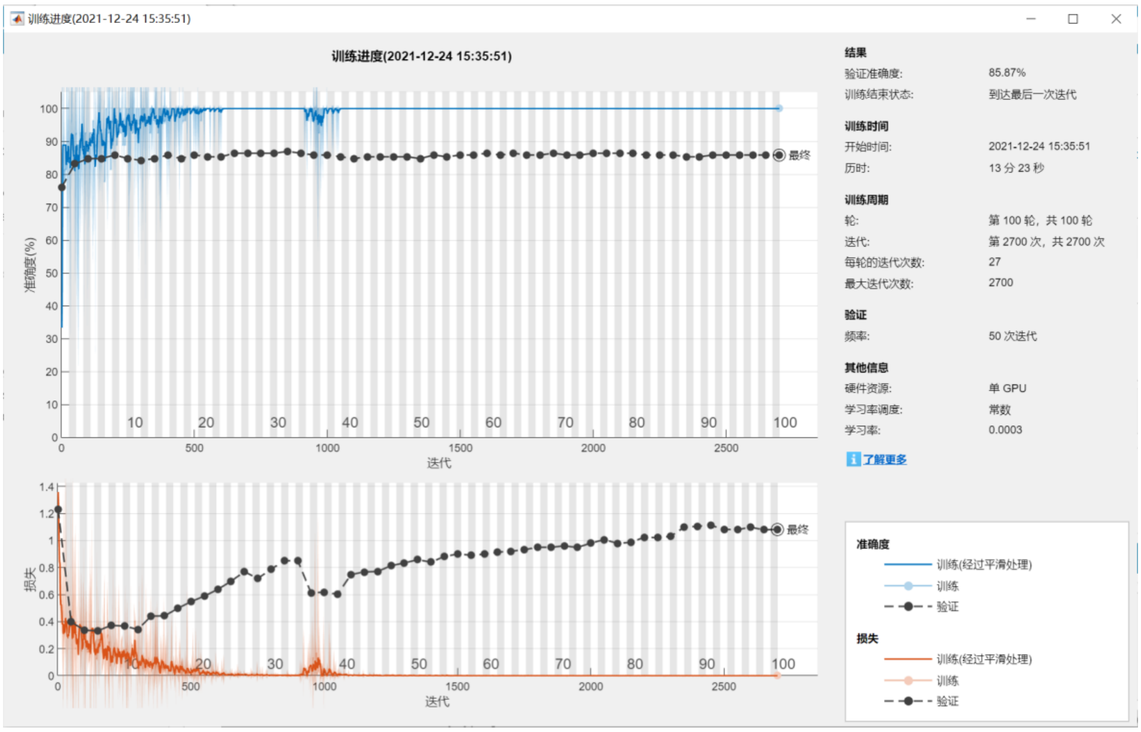
硬件资源: 单 GPU

学习率调度: 分段

学习率: 0.0001

了解更多

Bilstm



训练进度(2021-12-24 15:35:51)

结果

验证准确度: 85.87%

训练结束状态: 到达最后一次迭代

训练时间

开始时间: 2021-12-24 15:35:51

历时: 13 分 23 秒

训练周期

轮: 第 100 轮，共 100 轮

迭代: 第 2700 次，共 2700 次

每轮的迭代次数: 27

最大迭代次数: 2700

验证

频率: 50 次迭代

其他信息

硬件资源: 单 GPU

学习率调度: 常数

学习率: 0.0003

4.1.3 Classification Result Analysis

We have already displayed our result of conventional machine learning and deep learning, and now we are going to do some analysis.

To begin with, we'd like to summarize our result with the form of the table below:

Classification Method	Feature Engineering	Classifier	Network	AUC	Accuracy
Verbal (From the original paper)	Crackles and Wheezes By professionals	SVM	—	0.85	0.6
Conventional Machine Learning	Common MFCC	HMM	—	—	0.74
Conventional Machine Learning	Tiled MFCC	SVM	—	0.90	0.81
Conventional Machine Learning	Tiled MFCC	LDA	—	0.87	0.74
Conventional Machine Learning	GMM	SVM	—	0.88	0.87

Conventional Machine Learning	GMM	LDA	—	0.93	0.74
Deep Learning	—	—	LSTM	—	0.85
Deep Learning	—	—	Deeper LSTM	—	0.83
Deep Learning	—	—	BiLSTM	—	0.86

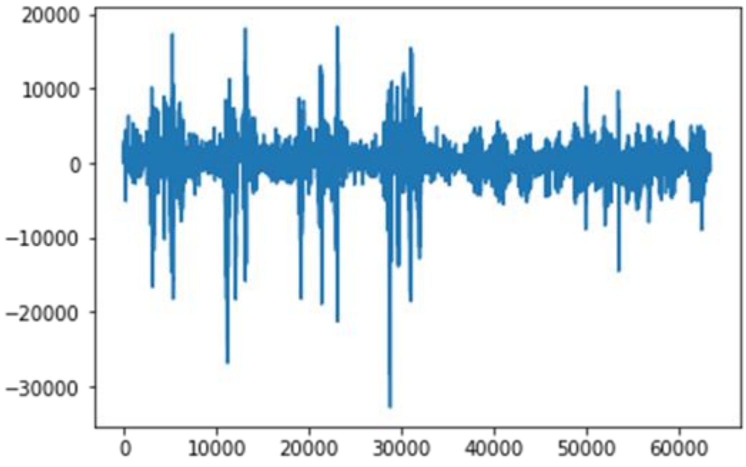
From the table we can know that, all of our experiments got a higher, and I think that it is fair to say a much higher accuracy score than the paper. Among them, GMM&SVM and BiLSTM stand out of the crowd.

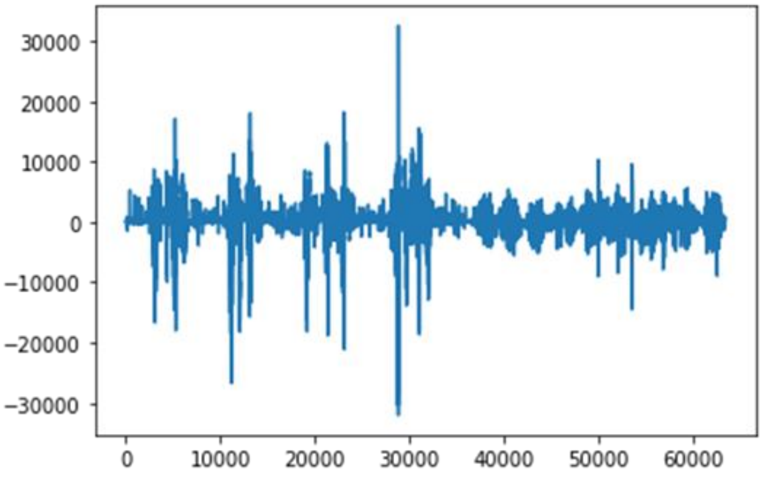
The former method indicates that GMM has a very good effect of clustering and extracting information from MFCC and the recording files in our project. The latter method indicates that the memory functionality of BiLSTM best fits our need.

4.2 Speech Denoising Result

4.2.1 Spectral Subtraction

The following two pictures show the comparison before and after noise reduction of one audio in our data set:



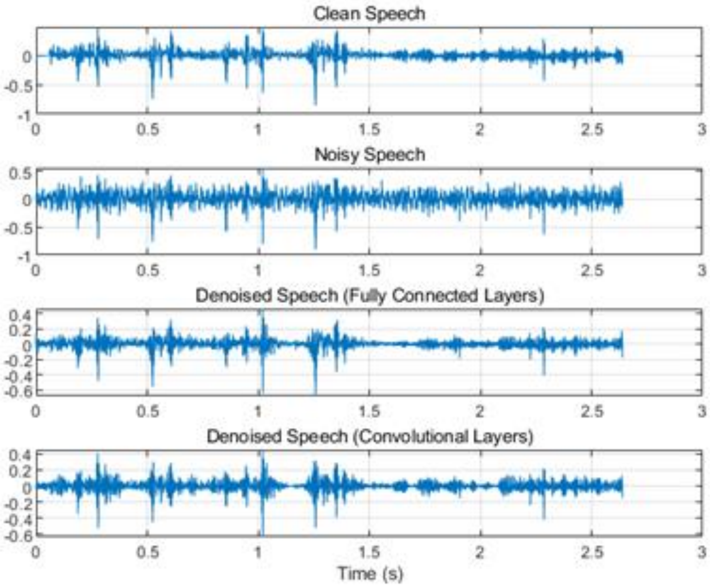


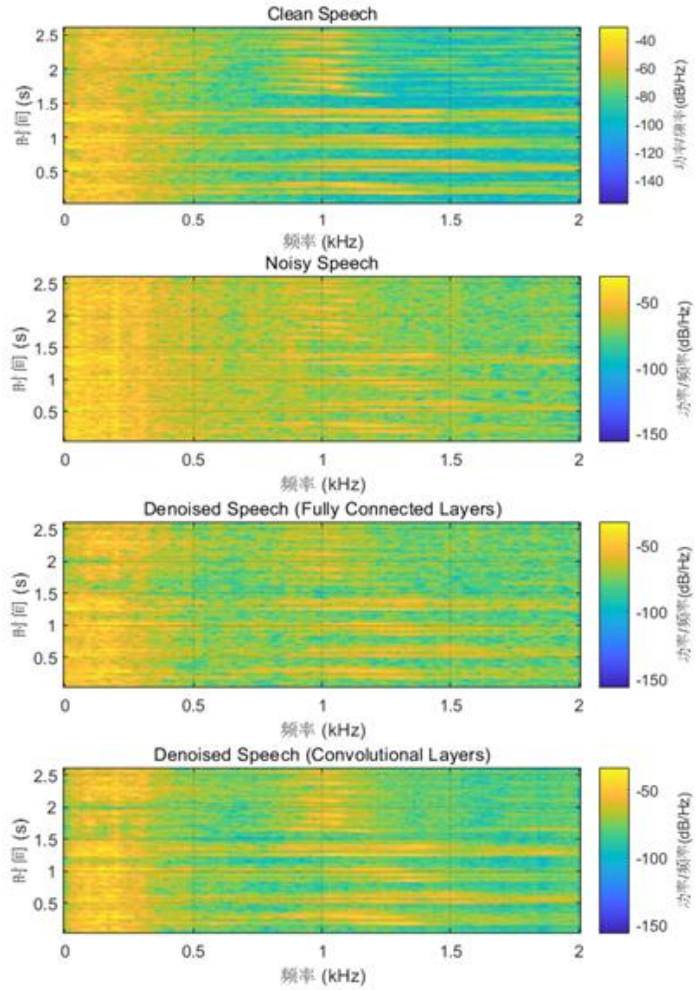
It can be seen that broadband noise is suppressed a lot, but the music noise still exists. When listening to the audio after noise reduction, the music noise can be heard, but it does remove a lot of background noise.

But unfortunately, more audios don't have such a good effect. Most of the denoised audios are converted into pure noise. It is because to most audios in our data set, their first 5 frames include some real voice, which is regarded as noise and removed in the process.

4.2.2 Neutral Network Denoising

We use the same audio as before, and plot the results:





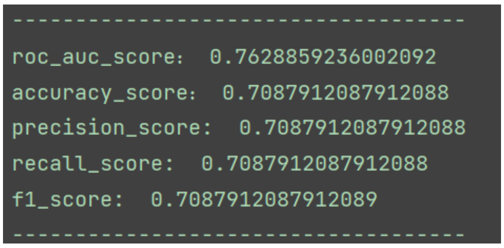
To be honest, the result given by fully connected layers are not very good, for even more noise appears in the denoised audio. In contrast, the result of convolutional neural networks is better. We can see that most useful information is reserved and some part of noise is definitely eliminated. But there is still the same problem: though it have a good effect on some audios, also a bad one on the most.

Both connected neural network and convolutional neural network cannot solve this problem, so we affirm it's because that: the situation in the data set is too complicated, and we cannot find a unified noise reduction method with our ability.

4.3 Final Result and Analysis

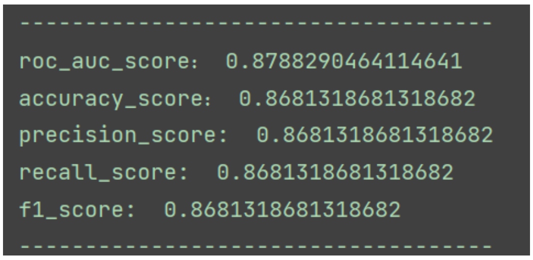
Now, we want to put denoising and FBank into our classification process to test its effect. We tried denoising with GMM feature engineering and SVM, LDA as classifier, and we tried to use FBank as our feature with BiLSTM. The result is shown below:

Denoise+GMM+LDA:



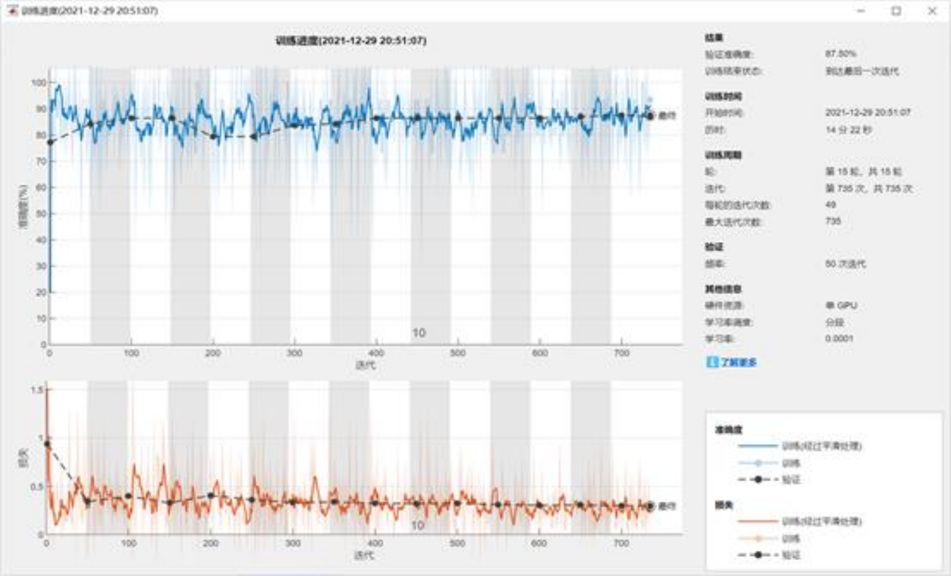
```
-----------------------------------
roc_auc_score:  0.7628859236002092
accuracy_score:  0.7087912087912088
precision_score:  0.7087912087912088
recall_score:  0.7087912087912088
f1_score:  0.7087912087912089
-----------------------------------
```

Denoise+GMM+SVM:



```
-----------------------------------
roc_auc_score:  0.8788290464114641
accuracy_score:  0.8681318681318682
precision_score:  0.8681318681318682
recall_score:  0.8681318681318682
f1_score:  0.8681318681318682
-----------------------------------
```

FBank+BiLSTM:



训练进度(2021-12-29 20:51:07)

结果
验证准确度: 87.50%
训练结束状态: 到达最后一次迭代

训练时间
开始时间: 2021-12-29 20:51:07
历时: 14 分 22 秒

训练周期
轮: 第 15 轮，共 15 轮
迭代: 第 735 次，共 735 次
每轮的迭代次数: 49
最大迭代次数: 735

验证
频率: 50 次迭代

其他信息
硬件资源: 单 GPU
学习率调度: 分段
学习率: 0.0001

The comparison can be clearly seen in the following table:

Method	Denoise	FBank	AUC	Accuracy
GMM+LDA	No	No	0.93	0.74
GMM+LDA	Yes	No	0.76	0.70
GMM+SVM	No	No	0.88	0.87
GMM+SVM	Yes	No	0.88	0.87
BiLSTM	No	No	——	0.86
BiLSTM	No	Yes	——	0.88

From this table we can know that, the result of denoising is not that good, but it is reasonable because of our analysis before. If we view LDA as the classifier, the accuracy is even lower. While the result of SVM remains the same, it actually took more time to finish calculating.

We can also validate the thought that DNN, such as BiLSTM can handle the correlation among FBank features since the accuracy score is 88%, which is 2% higher.

Ultimately, we can put our best results together so that we can check our project progress, displaying the comparison between the paper and our project.

Method	AUC	Accuracy
Crackels and Wheezes given by professionalists	0.85	0.6
Denoise+GMM+SVM	0.88	0.87
FBank+BiLSTM	——	0.88

From the final result, we can clearly see that our methods lead to a much better AUC and Accuracy Score than that from the method given by the paper, indicating staged success of our work.

5. Self-Assessment

In this section, we will display the performance evaluation since we have already talked about all our working process. Of course, we will arrange our result in order, and in this section we will analyze at length about our result.

Advantage:

① Our topic selection is relatively new. Many audio categories focus on the sounds of animals and musical instruments. We use the sound of a person’s breathing to distinguish whether a person has a certain respiratory disease. The topic itself has strong practical significance and value.

② For the case of unbalanced data, we also operate through different operations, so that the characteristics of cases with fewer data samples can also be well learned, making the results of the overall model more reliable

③ We conducted comparative tests on different feature selection, classifier selection, and neural network machine learning method selection, and selected the best results. At the same time, in the process of comparison, we also found that different characteristics are suitable for different classification methods, which also confirms the conclusions of our previous investigations.

④ Due to the particularity of our data set, the breathing sound and sound noise are likely to affect each other, so we have added a noise reduction step. We use basic spectral subtraction and neural network noise reduction, and have achieved good results, making our subsequent model training more reliable.

⑤ Our model training has achieved very good results. Compared with the 60% accuracy rate reproduced in the paper, our two final models have achieved an improvement of more than 25%. You must know that in medical diagnosis, any point of accuracy Upgrading can benefit tens of thousands of patients, so our model has achieved very good results.

Disadvantage：

① Of course, our model can be improved. First of all, we have not done enough in the use of the data set. As mentioned earlier, we only selected 2 breathing cycles for data alignment. In fact, we did not get much information about the remaining data. Good use, which also led to the final accuracy of our model not exceeding 90%. Later, we can study how to add more breathing data to our data set

② In addition, although we use two different methods to reduce noise, but through the results, we can find that the effect of noise reduction is not very significant. We think there are two reasons. One is that the breath sound audio data is too similar to the environmental noise and it is difficult to distinguish it. The other reason is that the noise reduction method we choose may not be suitable for the data set, and it can be used later. Try other different noise reduction methods

6. Summary

In general, our project to classify respiratory diseases based on breathing sounds has successfully completed the predetermined goal. Compared with the accuracy of the previous paper, we have improved by more than 25% on this basis.

This semester's speech recognition course has been done with the perfect summary. We not only use the mfcc and Gmm knowledge learned in the classroom to extract features, but also learn many machine learning and deep learning models for speech recognition outside of class, and introduce methods in the field of noise reduction into our model.

Through this project, our team has learned a lot. Thank you for this course and the teacher, that opened the door to speech time for us. We will apply the knowledge learned in the classroom to other learning research.