

REALIZATION AND APPLICATION OF VOICEPRINT RECOGNITION TECHNOLOGY

1、THE PROBLEM WE WANT TO SOLVE

We hope to realize a speaker identification technology based on voice print recognition, which mainly realizes two voicing recognition processes, one is to identify the speaker's identity, that is, to identify who is speaking among multiple speakers, and the other is speaker verification, that is, to judge whether a voice is said by a certain person. The main features we implement include: determining which voice is spoken by a registered person (or the voice is not registered), recognizing who is speaking in real time, and adding a voice print lock to the file management system so that the file owner can open the file.

There are differences in frequency spectrum, prosody and language features, so vowels are unique. So we can choose voice as an identity marker.

2、THE RELATED SURVERY

Voiceprint recognition needs three main steps: preprocessing, feature extraction, matching and recognition.

2.1 Pre-processing and Feature Extraction

The main purpose of pre-processing is to improve the quality of the voiceprint signal, reduce noise and interference, so that the subsequent processing steps are easier to capture the effective information of the voiceprint. Then, the effective information that can describe the characteristics of the speaker is extracted from the sound signal, and the feature vector is formed for the subsequent modeling. Common pretreatment and feature extraction methods include MelSpectrogram, Spectrogram, MFCC, Fbank, etc.

2.1.1 Spectrogram

A spectrogram is an image that visualizes an audio signal in time and frequency, where the horizontal axis represents time, the vertical axis represents frequency, and the color represents the strength of the signal. It is obtained by performing a Short-Time Fourier Transform (STFT) on the audio signal.

The spectrogram captures the energy distribution of the audio signal at different frequencies, but does not take into account the difference in the perception of

frequencies by the human ear.

2.1.2 Melspectrogram

On the basis of the spectrum diagram, Meir filter banks are introduced, which simulate the perception of different frequencies by the human ear. The Mayer spectrogram is more in line with the characteristics of the human auditory system and generally performs better in speech processing tasks.

The vertical axis of the Mayer spectrogram is the Mayer scale, not the direct frequency scale.

2.1.3 MFCC

MFCC is a feature extracted from the Mayer spectrum graph, which is obtained by taking logarithm of the Mayer spectrum graph and performing discrete cosine transform (DCT). MFCC captures the short-time dynamic characteristics of audio signals and is often used in speech recognition and sound print recognition.

MFCC can reduce the dimension of features and retain important information in the voice signal.

2.1.4 Fbank

Fbank is a method of filtering an audio signal through a set of filters and calculating the energy, similar to a Meir filter bank. Fbank extracts the energy of an audio signal across a range of frequency bands.

Fbank is commonly used for speech processing tasks and is a computationally efficient method for spectrum feature extraction.

To sum up:

We mainly compare Fbank and MFCC methods.

What Fbank features hope to achieve is to conform to the essence of the sound signal and to fit the reception characteristics of the human ear.

The detailed comparison are as follows:

- Computation: MFCC is carried out on the basis of FBank, so the computation of MFCC is larger

- Feature differentiation: FBank features have higher correlation (adjacent filter banks overlap), and MFCC has better discrimination, which is the reason why MFCC is used in most speech recognition papers instead of FBank

- Information content: The extraction of FBank features is mainly aimed at conforming to the essence of sound signal and fitting the characteristics of human ear reception. The MFCC does DCT de-correlation, so the Filter Banks contain more information than the MFCC

- Using the Gaussian Mixture Model (GMM) of diagonal covariance matrix

Because the correlation of different feature dimensions is ignored, MFCC is more suitable for the feature.

- DNN/CNN can make better use of the correlation of Filter Banks features to reduce losses.

From the current trend, because of the gradual development of neural networks, FBank features are becoming more and more popular.

2.2 Training Model

After the voiceprint features are extracted, the model is trained or compared to identify the speaker or verify the speaker's identity. A variety of deep learning models can be used, including EcapaTdn, TDNN, Res2Net, ResNetSE, ERes2Net, CAM++ and other deep learning models commonly used in speech processing. Use a model for training and testing. In the training phase, the model is trained using a labeled voiceprint dataset. In the validation or testing phase, a model is used to identify or validate new sound samples.

2.2.1 TDNN

TDNN stands for Time-Delay Neural Network.

TDNN is a kind of neural network structure based on delay layer. Its main feature is to introduce a time delay layer in the network to capture the time information in the input sequence. TDNN typically consists of multiple time delay layers, each of which processes a different time step of the input. This structure allows the network to efficiently learn long-term dependencies in time series data.

TDNN is widely used in speech recognition, speech processing, natural language processing and other fields. In speech intelligibility models, TDNN may be used to learn the timing features of speech signals to improve the model's adaptability to different speech situations.

2.2.2 ECAPA-TDNN

ECAPA-TDNN stands for Emphasized Channel Attention, Propagation, and Aggregation in TDNN.

ECAPA-TDNN is a voice print recognition model based on TDNN structure, which introduces channel attention mechanism, information propagation and aggregation to improve the performance of the model. The initial frame layer of the method can be reconstructed into one-dimensional Res2Net modules with influential jump connections, similar to SE-ResNet, into which Squeeze-and-Excitation blocks are introduced to explicitly model the inter-channel dependencies. Second, neural networks are known to learn layered features, with each layer operating at different levels of complexity. To take advantage of this complementary information, features at different levels can be aggregated and propagated. Finally, the statistical aggregation module is improved by introducing channel-related frame attention. This allows the network to focus on a different subset of frames during the statistical

estimation of each channel.

2.2.3 Res2Net

Res2Net is a neural network structure based on residual connections. By constructing hierarchical residual connections within a residual block, multi-scale features are represented at granularity level. In voicing recognition, Res2Net may be used to improve the modeling ability of different scale speech features.

2.2.4 ResNetSE

ResNetSE stands for Residual Networks for Speaker Embeddings.

The ResNetSE approach focuses on channel relationships and proposes a new architectural unit, which we call the "squeeze and excitation" (SE) block, that adaptively recalibrates channel feature responses by explicitly modeling the interdependencies between channels. The research shows that these blocks can be stacked on top of each other to form a SENet architecture that generalizes very efficiently over different data sets. It was further demonstrated that SE blocks bring significant performance improvements to existing state-of-the-art CNNs at a slight increase in computational cost.

This method is to improve the representational ability of CNNs. The core building block of convolutional neural networks (CNN) is the convolutional operator, which enables the network to construct information features by fusing spatial and channel information in the local receptive domain of each layer. Extensive research has investigated the spatial components of this relationship in an attempt to enhance the representation of CNN by improving the spatial coding quality of the entire feature hierarchy.

2.2.5 ERes2Net

ERes2Net is a new architecture for enhanced Res2Net. It combines local and global feature fusion technologies to improve performance. Local feature fusion (LFF) fuses features into a single residual block to extract local signals. Global feature fusion (GFF) uses acoustic features of different scales as inputs to aggregate signals globally.

2.2.6 CAM++

CAM++ is an extended version of the channel attention module that enhances the model's focus on a particular channel. The channel attention mechanism helps the model make better use of important features in the input data.

CAM++ is an efficient network based on context-aware masking. The network uses a dense-connected time delay neural network (D-TDNN) as its backbone and a novel multi-granularity pool to capture text information at different levels. A large number of experiments conducted on two common benchmarks, VoxCeleb and CN-Celeb, show that This architecture outperforms other mainstream speaker verification systems with lower computational cost and faster inference speed.

3、THE STRUCTURE OF PROGRAM

3.1 DataSet

Training data set: The audio files are encoded as single channels and sampled with 16kHz and 16 bit accuracy.

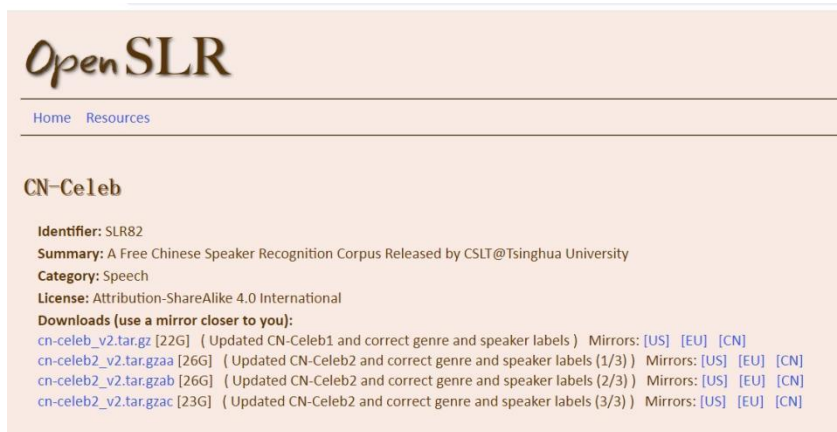
Data set 1:

Contains more than 130,000 quotes from 1,000 Chinese celebrities.

Dataset 2:

More than 520,000 quotes from 2,000 Chinese celebrities.

There are more than 630,000 discourse data from 11 different life scenarios, including songs, interviews, movies, and entertainment programs.



The screenshot shows the OpenSLR website interface. At the top, there's a navigation bar with 'Home' and 'Resources'. Below it, the 'CN-Celeb' dataset is highlighted. The page provides the following information:

- Identifier:** SLR82
- Summary:** A Free Chinese Speaker Recognition Corpus Released by CSLT@Tsinghua University
- Category:** Speech
- License:** Attribution-ShareAlike 4.0 International
- Downloads (use a mirror closer to you):**
 - [cn-celeb_v2.tar.gz](#) [22G] (Updated CN-Celeb1 and correct genre and speaker labels) Mirrors: [\[US\]](#) [\[EU\]](#) [\[CN\]](#)
 - [cn-celeb2_v2.tar.gz](#) [26G] (Updated CN-Celeb2 and correct genre and speaker labels (1/3)) Mirrors: [\[US\]](#) [\[EU\]](#) [\[CN\]](#)
 - [cn-celeb2_v2.tar.gz](#) [26G] (Updated CN-Celeb2 and correct genre and speaker labels (2/3)) Mirrors: [\[US\]](#) [\[EU\]](#) [\[CN\]](#)
 - [cn-celeb2_v2.tar.gz](#) [23G] (Updated CN-Celeb2 and correct genre and speaker labels (3/3)) Mirrors: [\[US\]](#) [\[EU\]](#) [\[CN\]](#)

The test data set is approximately 17,000 pieces.

cn-celeb-test.zip (2918.60M) [下载](#)

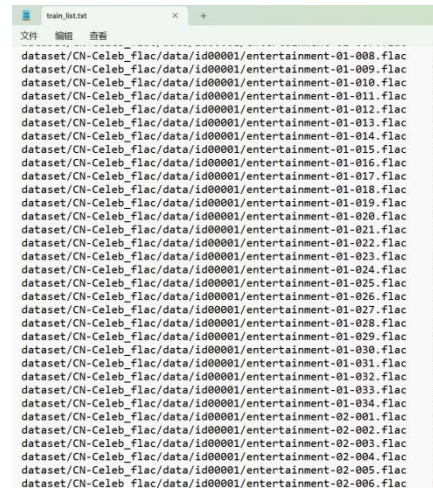
File Name	Size	Update Time
cn-celeb-test/enroll_list.txt	10278	2023-08-04 17:10:43
cn-celeb-test/enroll/id00900-enroll.flac	486591	2023-08-04 17:10:01
cn-celeb-test/enroll/id00897-enroll.flac	629380	2023-08-04 17:10:00
cn-celeb-test/enroll/id00924-enroll.flac	432864	2023-08-04 17:10:01
cn-celeb-test/enroll/id00993-enroll.flac	726359	2023-08-04 17:10:01
cn-celeb-test/enroll/id00898-enroll.flac	808637	2023-08-04 17:10:00
cn-celeb-test/enroll/id00963-enroll.flac	1373795	2023-08-04 17:10:01

3.2 Make a File List of the Dataset

Run create_data.py to get the list of files for the dataset.

Get train_list.txt.

Used to load and organize data when training and evaluating deep learning models.



The screenshot shows a file explorer window titled 'train_dataset'. It displays a list of files in a directory structure. The files are organized by dataset, speaker, and audio format. The list includes files from 'dataset/CN-Celeb_flac/data/id00001/entertainment-01-008.flac' to 'dataset/CN-Celeb_flac/data/id00001/entertainment-02-006.flac'. Each file is listed with its full path and a corresponding number (likely a count or index) in the rightmost column.

File Path	Count
dataset/CN-Celeb_flac/data/id00001/entertainment-01-008.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-009.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-010.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-011.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-012.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-013.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-014.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-015.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-016.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-017.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-018.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-019.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-020.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-021.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-022.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-023.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-024.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-025.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-026.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-027.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-028.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-029.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-030.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-031.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-032.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-033.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-01-034.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-02-001.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-02-002.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-02-003.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-02-004.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-02-005.flac	1
dataset/CN-Celeb_flac/data/id00001/entertainment-02-006.flac	1

3.3 Train

(1) FILTERING OF DATA SETS

Only audio with a length of 0.5-3s is retained.

The audio volume is normalized to maintain a consistent sound volume.

(2) DATA LIST

Training data data.

Evaluation of registered data (used to register known speaker voice print information in the system for subsequent speaker verification or identification tasks).

Evaluate test data.

(3) DATA ENHANCEMENT

Using speed perturbation enhancement, the data is divided into three different categories, each corresponding to a different speed.

(4) USING NOISE ENHANCEMENT

Ambient noise is added to audio to enhance data. The probability is 0.2.

(5) DATA ENHANCEMENT AT THE SPECAUGMENTATION SPECTRUM LEVEL

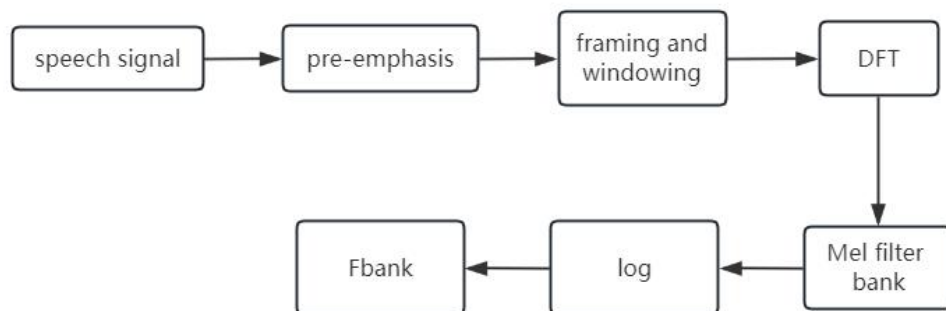
Random spectrum mask.

Random time mask.

Data enhancement can improve the robustness of the model and help to make

the model more adaptable to different sound changes

(6) USING FBANK



(7) OPTIMIZER

The optimizer is Adam. Adam is a common optimization algorithm for gradient descent. The First Moment Estimation (i.e. the mean value of the gradient) and the Second Moment Estimation (i.e. the uncentralized variance of the gradient) of the gradient are comprehensively considered, and the update step is calculated.

(8) LEARNED

WarmupCosineSchedulerLR was selected as the learning rate scheduling function. It will increase the learning rate at the beginning of the training, and then gradually decrease the learning rate.

The argument is min_lr: float 1e-5.

max_lr: 0.001.

warmup_epoch: 5, the learning rate gradually increases to 0.001 for the first five rounds, and gradually decreases after that.

The model has a higher learning rate at the early stage of training, which helps to converge faster, and achieves a more stable optimization process by cosine annealing mechanism at the later stage of training. Improve the generalization ability and training stability of the model.

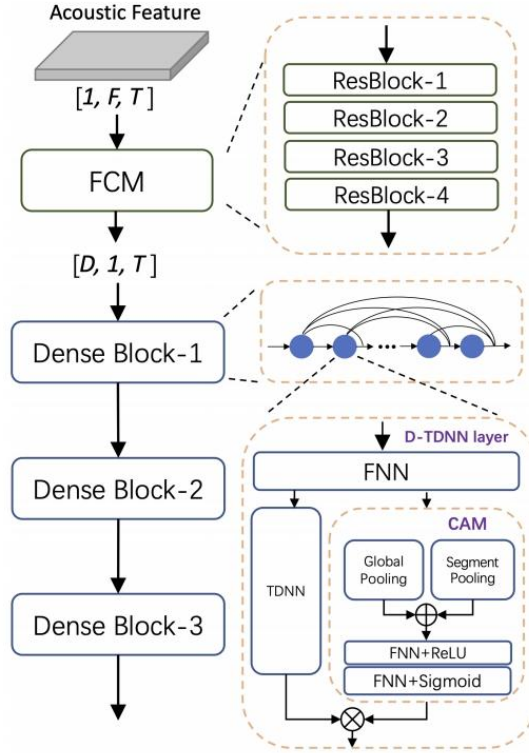
(9) LOSS FUNCTION

ArcFace Loss: Additive Angular Margin Loss (additive angular margin loss), adding angular interval m to θ , introduces angular cosine interval, that is, the Angle of the corresponding feature point of the sample in the feature space. By adjusting the interval, the similarity between samples of the same speaker can be enhanced, and the angle between samples of different speakers can be ensured to be large enough, so that the model can learn distinguishable voiceprint features more easily.

3.4 CAM++ Model

The network is based on a densely connected time delay neural network (D-TDNN), which uses a novel multi-granularity pool to capture context information at different levels. This architecture outperforms other mainstream speaker verification systems with lower computational cost and faster inference speed.

Details of the model:



FCM: Front-end convolutional module. Consisting of multiple two-dimensional convolution blocks with residual connections, the acoustic features are encoded in the time-frequency domain to obtain high-resolution time-frequency details, and the resulting feature map is then flattened along the channel and frequency dimensions to be used as input to D-TDNN.

D-TDNN is the backbone, containing 3 blocks, each containing a series of D-TDNN layers, for better speaker verification, the depth of the D-TDNN network has been increased, the original D-TDNN had only two blocks, each with 6 and 12 D-TDNN layers, now there are 3 blocks, the number of layers is 12, 24 and 16.

D-TDNN consists of a feedforward neural network FNN and TDNN layers. Each D-TDNN layer contains an improved CAM module that assigns different attention weights to the output features of the internal TDNN layer. Multi-granularity pooling combined with global average pooling combined with segmented average pooling effectively aggregates different levels of context information.

CAM: context-aware masking. This module focuses on the speaker of interest, blurring irrelevant noise.

The output feature of FNN is represented as X , and X is input to the TDNN layer to extract the local time feature F .

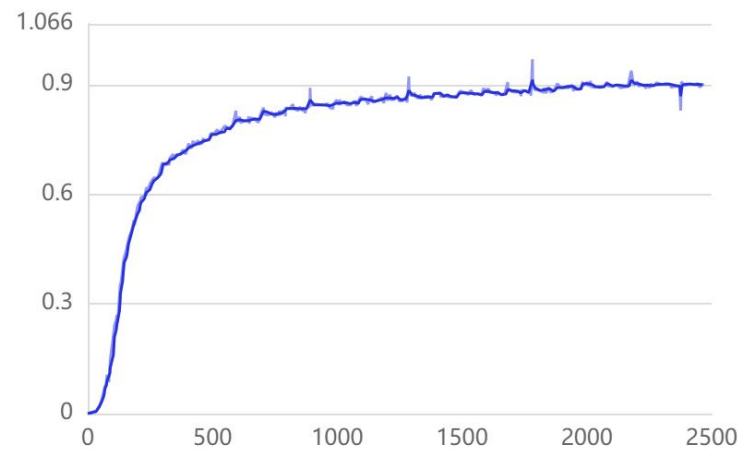
Use global average pooling to obtain context information at the global level, and use Segment Pooling to obtain context information at the segment level. By aggregating different levels of context information, the predictive context aware mask M is obtained.

M and local time feature F are multiplied element-by-element, resulting in efficient context-aware masking in each D-TDNN layer.

4、TRAINING RESULT

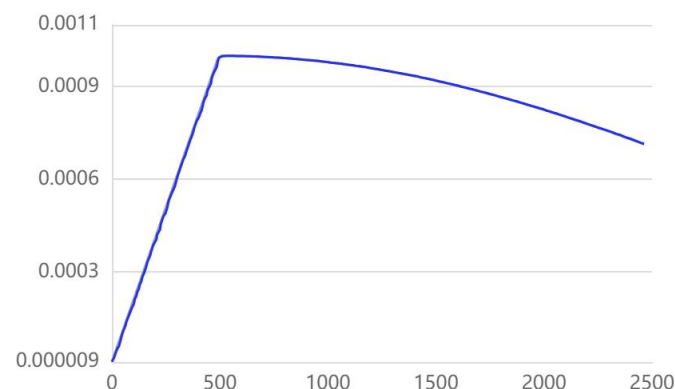
The horizontal coordinate is the number of steps, and a training round is about 100 steps.

Train/Accuracy

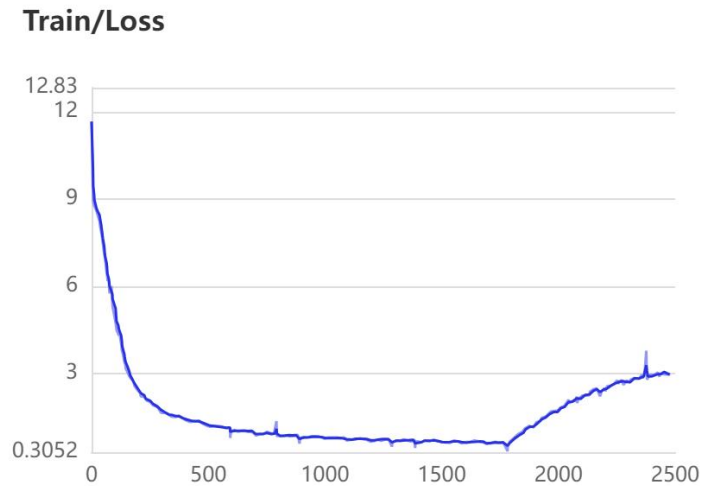


The recognition accuracy rate gradually increases and converges to about 0.9.

Train/lr



The learning rate was WarmupCosineSchedulerLR method, which steadily increased to 0.001 in the first 5 rounds and then slowly decreased.



The LOSS gradually decreases to a relatively small value, and then the phenomenon of rebound appears. The speculated reasons are as follows:

Overfitting: The model may overfit on the training data and degrade on the validation or test set. This causes accuracy to stagnate and LOSS to increase because the model is over-reliant on noise or specific patterns in the training data.

Learning rate scheduling function: When the learning rate decreases, the step size of the model in the learning direction decreases, which may cause the model to oscillate in a certain region of the parameter space and cannot continue learning, which may lead to stagnation of accuracy and increase of LOSS.

5、 PERFORMANCE EVALUATION

Round 25 results:



MinDCF is an indicator to evaluate system performance, which takes into account false alarm rate and false alarm rate, and can be calculated by adjusting the decision threshold. The lower the MinDCF, the better the system performance.

Round	Optimal threshold	EER	MinDCF
9	0.28	0.13167	0.66159
18	0.25	0.12088	0.60756
24	0.24	0.11140	0.57194
25	0.24	0.11197	0.57535

6、FUNCTIONALITY OF PROGRAM

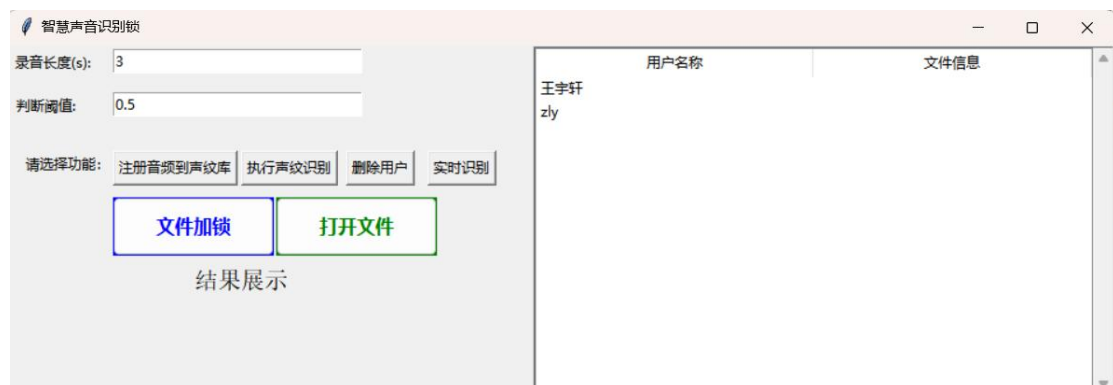
6.1 User Manual

Our main functions include registering audio to the voiceprint library, performing voiceprint recognition, deleting users, real-time voiceprint recognition, as well as file locking and file encryption for opening files.

The user can select the recording length, that is, the audio duration of a recording. The user can also select the recognition threshold, that is, the degree of similarity of a voice in the recorded sound and sound print library. The larger the threshold, the more similar the two people are required to be identified as the same person, and the smaller the threshold, the opposite.

6.1.1 Basic Functions of Voiceprint Recognition:

(1) Register audio to voiceprint library: Click the button, the user records his voice, the prompt box will be displayed after the arrival time, enter the recorded user name, click "OK" to complete the registration.

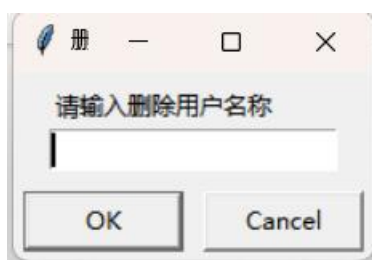


(The database now has the voice data of two people named 王宇轩 and zly. We can see the information in the list on the right)

(2) Voice print recognition: The user can click the button to start recording. The user records a voice within a specified time, and the system performs voice print recognition after the specified time.



(3) Delete user: Click the button and enter the user name you want to delete in the prompt box. Click "OK" and delete successfully.

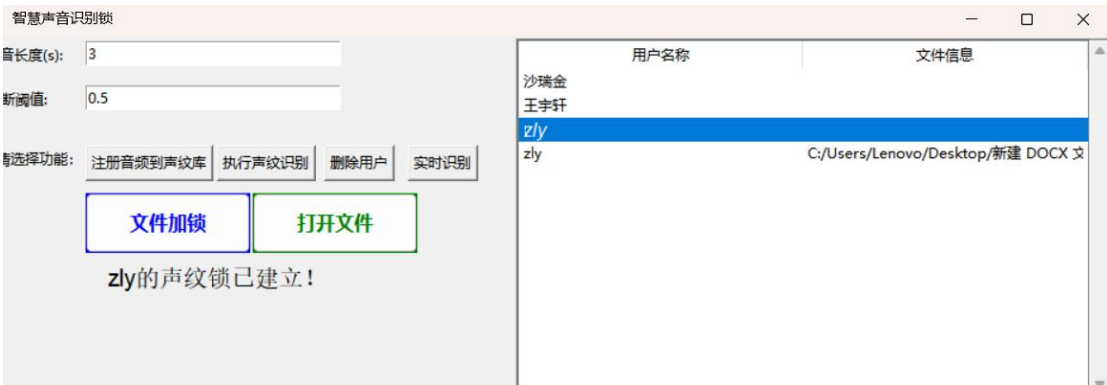
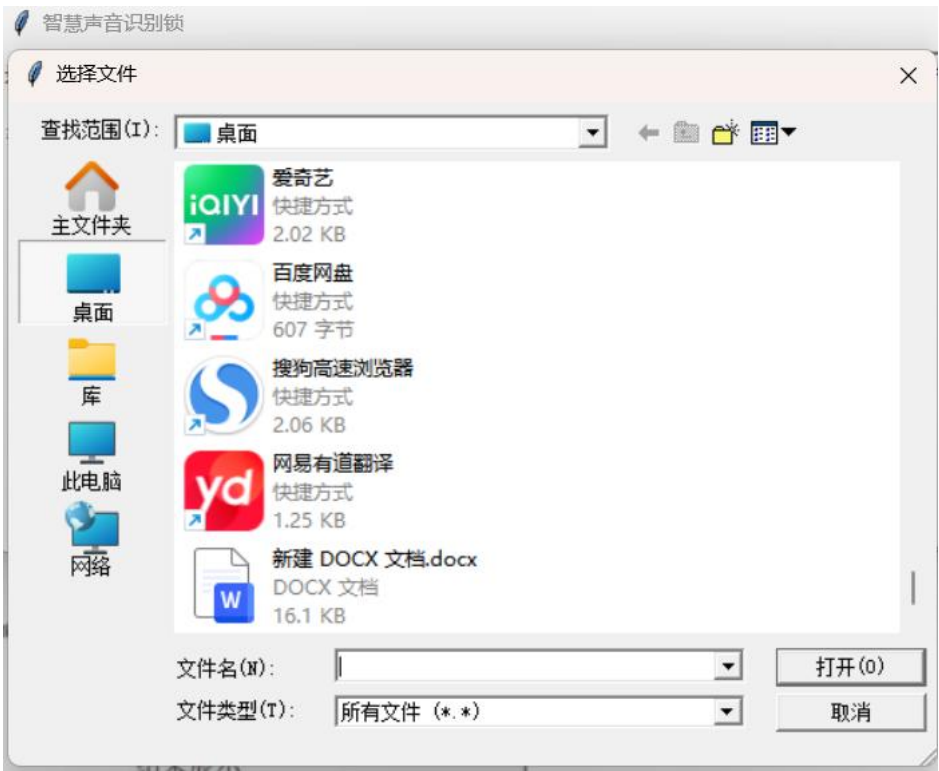


(4) Real-time recognition: Click the button to start real-time recognition. After the "Please speak" prompt appears, the user starts to speak. Users matching the current voice print library will prompt "A user is speaking".



6.1.2 File lock function

(1) File lock: Click the "file lock" button, select the file you want to lock, and select a registered user in the pop-up prompt box as the voice print lock of the current file. After the successful addition, the added file lock will be displayed in the user file information list on the right.



(2) Open the file: click the "Open file" button, select the file you want to open, and start recording after the selection is completed. If the file matches, the file can be successfully opened. If the file does not match, an error message is displayed.



6.2 Function implementation introduction

(1) The process for "Performing voiceprint recognition" :

After clicking the "Perform voiceprint recognition" button, the self.recognize method is called to implement voiceprint recognition. The specific code is as follows:

```
# 识别
1个用法
def recognize(self):
    threshold = float(self.threshold.get())
    record_seconds = int(self.record_seconds.get())
    # 开始录音
    self.result_label.config(text="正在录音...")
    audio_data = self.record_audio.record(record_seconds=record_seconds)
    self.result_label.config(text="录音结束")
    name, score = self.predictor.recognition(audio_data, threshold, sample_rate=self.record_audio.sample_rate)
    if name:
        self.result_label.config(text=f"说话人为: {name}, 得分: {score}")
    else:
        self.result_label.config(text="没有识别到说话人, 可能是没注册。")
```

In the recognize method in the VoiceRecognitionGUI class, a voiceprint recognition operation is performed. Here's the code for the recognize method:

(2) The process of "real-time voiceprint recognition" :

Triggered by a click on the recognize_real_button, the button's text tag will switch to either "End voiceprint recognition" or "Live voiceprint recognition" when clicked. When you click, the recognize_thread method is called. The code for the recognize_thread method is as follows:

1个用法

```
def recognize_thread(self):
    if not self.recognizing:
        self.recognizing = True
        self.recognize_real_button.config(text="结束声纹识别")
        threading.Thread(target=self.recognize_real).start()
        threading.Thread(target=self.record_real).start()
    else:
        self.recognizing = False
        self.recognize_real_button.config(text="实时声纹识别")
```

In the `recognize_thread` method, recognizing the state of a recognizing variable is used to determine whether to start or end real-time voicing recognition. If enabled, set `self.recognizing` to `True` and change the text of the button to "End voiceprint recognition". Two threads are then started, one for real-time voicing recognition (`recognize_real` method) and the other for real-time recording (`record_real` method).

Here's the code for the `recognize_real` method, which `recognize_real` does voiceprint recognition in real time in a loop, updating the text labels on the interface based on the recognition results:

```
# 识别
1个用法
def recognize_real(self):
    threshold = float(self.threshold.get())
    while self.recognizing:
        if len(self.record_data) < self.infer_len: continue
        # 截取最新的音频数据
        seg_data = self.record_data[-self.infer_len:]
        audio_data = np.concatenate(seg_data)
        # 删除旧的音频数据
        del self.record_data[:len(self.record_data) - self.infer_len]
        name, score = self.predictor.recognition(audio_data, threshold, sample_rate=self.record_audio.sample_rate)
        if name:
            self.result_label.config(text=f"【{name}】正在说话")
        else:
            self.result_label.config(text="请说话")
```

In this code, by checking the state of `self.recognizing`, when it is in the real-time voiceprint recognition state, it continuously obtains the latest audio data, carries out voiceprint recognition, and then updates the text label on the interface according to the recognition result.

(3) The process of "open file" :

The file opening function is triggered by clicking the "Open File" button, whose click event invokes the `unlock_file` method. Here is the code for the `unlock_file` method:

In `unlock_file` method, first by `filedialog askopenfilename` for user to select the file path. Then, the `record_audio` object is called to make the recording. Next, you perform voiceprint recognition using the predictor object's recognition method to get the speaker's name and score. If the speaker is recognized, the speaker's voice lock information is used to determine whether the file can be opened. If the sound

lock matches, try to open the file using subprocess.run, and if the sound lock does not match, an error message is displayed. Finally, according to the recognition result, the text label on the interface is updated and the related prompt box is displayed.

```
def unlock_file(self):
    file_path = filedialog.askopenfilename(title="选择文件", filetypes=[("所有文件", "*.*)"])

    if file_path:
        # Do something with the selected file path, e.g., display it or pass it to your functions
        print("Unlock file:", file_path)
        threshold = float(self.threshold.get())
        record_seconds = int(self.record_seconds.get())
        # 开始录音
        self.result_label.config(text="正在录音...")
        audio_data = self.record_audio.record(record_seconds=record_seconds)
        self.result_label.config(text="录音结束")
        name, score = self.predictor.recognition(audio_data, threshold, sample_rate=self.record_audio.sample_rate)
        if name:
            self.result_label.config(text=f"说话人为: {name}, 得分: {score}")
            if file_path in self.get_files_by_voice(name):
                try:
                    result = subprocess.run(['start', '', file_path], shell=True, check=True)
                except subprocess.CalledProcessError as e:
                    messagebox.showinfo("提示", f"无法打开文件: {str(e)}, 该文件正在被占用")
            else:
                messagebox.showerror("错误", "声纹不匹配, 文件打开失败!")
        else:
            self.result_label.config(text="没有识别到说话人, 可能是没注册。")
            messagebox.showerror("错误", "声纹不匹配, 文件打开失败!")
```

7、ADVANTAGES AND DISADVANTAGES

Advantages:

The robustness of the model was improved by using speed disturbance enhancement, noise enhancement and data enhancement at SpecAugmentation spectrum level, and the model had better adaptability to different sound changes.

Using Additive Angular Margin Loss makes it easier for the model to learn distinguishable voiceprint features.

Using the CAM++ model, this architecture outperforms other mainstream speaker verification systems with lower computational cost and faster inference speed.

Disadvantages:

The maximum accuracy can only reach about 90%, then convergence begins, and then the LOSS of the model gradually increases.

Locking files is only implemented inside the program, while the program is running, files can still be opened from other paths without the need for voice unlock.