ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

Speaker recognition system based on MFCC feature extraction CNN architecture

Zhiyi Ji[1,a,#], Guanghao Cheng[2,b,#], Tianyu Lu[3,c,#], Zhiqi Shao[4,d,#]

[1]*Wuxi Taihu University, Wuxi, China*
[2]*Central South University, Changsha, China*
[3]*Tianjin University of Technology, Tianjin, China*
[4]*Shandong Institute of Petroleum and Chemical Technology, Dongying, China*
[a]*15006261789@126.com,* [b]*cguanghao150@gmail.com,* [c]*lty00218@outlook.com,*
[d]*1270769812@qq.com*
[#]*Co-first author*

Abstract: *This project adopts a self-designed neural network architecture to develop a concise and efficient speaker identification system. The main structure of the system consists of two major components: First, the MFCC (Mel-Frequency Cepstral Coefficients) feature extraction, which captures the unique voice characteristics of the speaker through meticulous audio signal processing; Second, the convolutional neural network (CNN), composed of multiple convolutional layers, pooling layers, and a fully connected layer, is primarily used for in-depth analysis and learning of the extracted features, thereby achieving high-precision speaker identification. Through the MFCC feature extraction and CNN processing, the system was trained and tested on a self-built data set, achieving an accuracy of 89%, realizing high-precision identification. The system is characterized by its simplicity and efficiency, making it suitable for deployment on edge devices without relying on powerful central servers, enabling quick response.*

Keywords: *MFCC feature extraction; Convolutional Neural Network (CNN); Speaker recognition; Identity recognition; Audio processing*

1. Introduction

With the rapid advancement of information technology, speech technology has emerged as a focal point of research in the scientific and technological domain. Speaker identification, an integral component of speech technology, has demonstrated significant potential for application across various domains including security authentication, smart home systems, and intelligent assistants in recent years. Nevertheless, traditional speaker identification methods are constrained by limitations in processing efficiency, accuracy, and system simplicity that hinder their ability to meet the escalating practical demands. Henceforth, it is imperative to develop an efficient, precise, and streamlined speaker identification system.

1.1. Context and Importance

In today's information society, identity authentication plays a crucial role in ensuring information security. Speaker identification technology, as a biometric-based method of authentication, offers the advantages of non-contact and resistance to forgery, and is increasingly garnering widespread attention. This technology automatically identifies and verifies the speaker's identity by analyzing their unique characteristics within the voice signal. Not only does this enhance the security of information systems, but it also delivers more convenient and personalized service experiences for users. Consequently, research and application of speaker identification technology hold significant practical importance and offer extensive market potential.

1.2. Limitations of current treatment approaches.

However, the current speaker identification systems available in the market have encountered certain issues. Traditional methods exhibit inefficiency when handling large volumes of audio data and fail to meet real-time requirements. Moreover, accuracy of identification often significantly decreases in

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

complex environments, such as those with noisy interference or speech variation. Additionally, while some systems demonstrate acceptable recognition performance, their high complexity poses challenges for deployment in resource-constrained edge devices. Consequently, there is an urgent demand for a speaker identification method that can effectively balance efficiency, accuracy, and simplicity.

1.3. What approaches can be employed to address this issue?

To address the limitations of current methodologies, this study introduces a speaker identification system based on MFCC feature extraction and convolutional neural network processing. Initially, we employed the MFCC feature extraction method to effectively capture the spectral characteristics of speech signals, providing robust support for subsequent identification. Subsequently, we utilized a meticulously designed convolutional neural network architecture to process the extracted features, ensuring high identification performance while significantly reducing computational resource consumption. Additionally, data augmentation techniques and regularization methods were incorporated to enhance the model's generalization capability and mitigate overfitting tendencies. Through these strategies' implementation, an efficient, precise, and streamlined speaker identification system was successfully developed.

1.4. The Structural Organization of the Research Paper

This paper first introduces the background and significance of the project, as well as the problems with existing methods. Then it introduces related technologies, followed by an explanation of our solution, including the MFCC feature extraction method and the design of the convolutional neural network architecture. Then it introduces the data sources and richness used in the experiment. Next, it describes the design and implementation process of the system, including the construction of the neural network, the training strategy, and optimization methods. Finally, it analyzes and discusses the experimental results, summarizes the contributions and shortcomings of the project, and looks forward to the future research direction and application prospects.

Through this paper, we hope to provide new ideas and methods for the development of speaker identity recognition technology and serve as a reference and inspiration for researchers in related fields.

2. Related Technologies

This chapter will provide explanations of the relevant technologies.

2.1. Speaker Recognition Technology

Speaker recognition, as a significant branch of speech recognition, aims to confirm a speaker's identity through in-depth analysis and feature extraction of the speech signal.The former identifies the target speaker from a group of candidates, while the latter verifies whether the claimed identity is true[1]. Our system aims to provide a simple and efficient solution to promote the deployment of intelligent voice systems at the edge side.

2.2. Key Technologies

Our system design is mainly based on two core technologies:

2.2.1. MFCC Feature Extraction

MFCC (Mel-Frequency Cepstral Coefficients) has been successfully proven to use for audio classification[2]. Combined with the powerful feature learning and classification capabilities of convolutional neural networks (CNN), significant achievements have been made in image and speech recognition fields. By integrating these two technologies, we aim to build a speaker recognition system that is both efficient and accurate.

2.2.2. Convolutional Neural Networks

They are a repreConvolutional Neural Networks (CNN) are a type of deep feedforward neural network characterized by local connections and shared weights[3]. Sentative algorithm in deep learning, doing well in handling image-related machine learning problems such as image classification, object detection, and image segmentation, significantly improving performance in various visual tasks.

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

CNNs have the capability of representation learning, enabling shift-invariant classification of input information according to their hierarchical structure. They can perform both supervised and unsupervised learning. The parameter sharing of convolutional kernels in hidden layers and the sparsity of interlayer connections allow CNNs to learn grid-like topology features, such as pixels and audio, with relatively small computational overhead, achieving stable results without additional feature engineering. CNNs are extensively used in computer vision and natural language processing[4].

The basic structure of a CNN generally includes: convolutional layers, pooling layers, activation functions, fully connected layers, and output layers.

Convolutional Layer: Convolution matrix, or convolution kernel.

Pooling Layer: Also known as the subsampling layer, is a deep learning structure that reduces data dimensionality, decreases parameters and computation, and mitigates overfitting. It is a crucial layer for controlling overfitting in convolutional neural networks.

Max Pooling is a common technique used for pooling operations in deep learning. It divides image pixels into different subunits and selects the pixel with the maximum value in each subunit as the output pixel (Figure 1). The advantages of Max Pooling include effectively reducing the number of parameters, computational load, overfitting risk, and retaining the most important features of the original image[5].

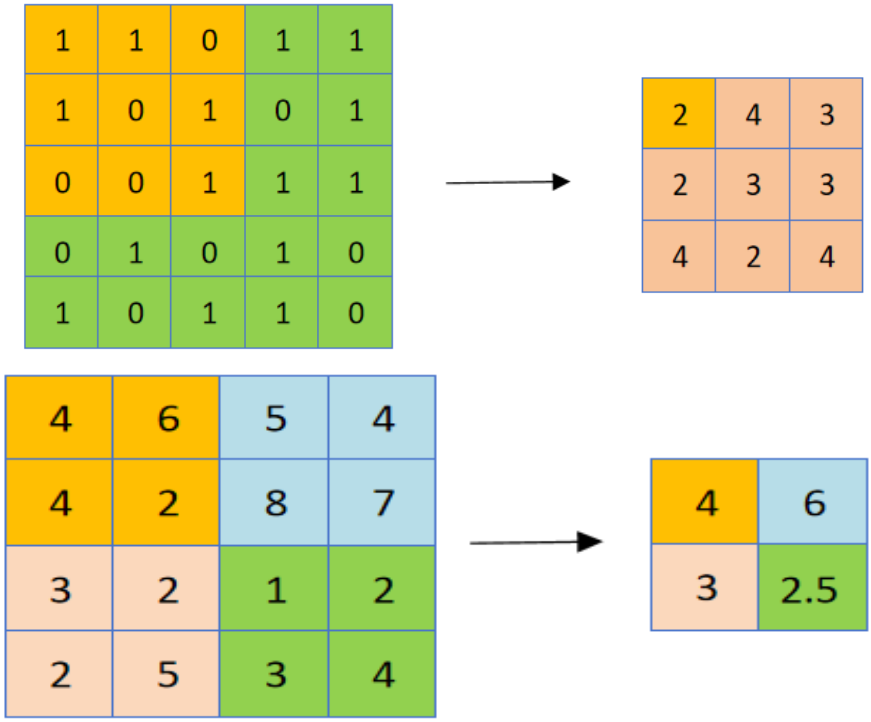

Figure 1: Max Pooling Diagram

Activation Functions:

Common activation functions include the Sigmoid function, Tanh function, ReLU function, and Softplus function.

Fully Connected Layer:

Often used as the final layer in classification problems, its primary function is to fully connect the data matrix and then output data according to the number of classifications.

3. System Design and Implementation

In this chapter, we will explain the system architecture and introduce some optimization techniques.

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

3.1. Overall System Architecture

This system adopts a deep learning architecture based on convolutional neural networks for speaker identification. The overall architecture consists of several key components, including CNN layers, pooling layers, and fully connected neural network (FCN) layers, as shown in Figure 2.

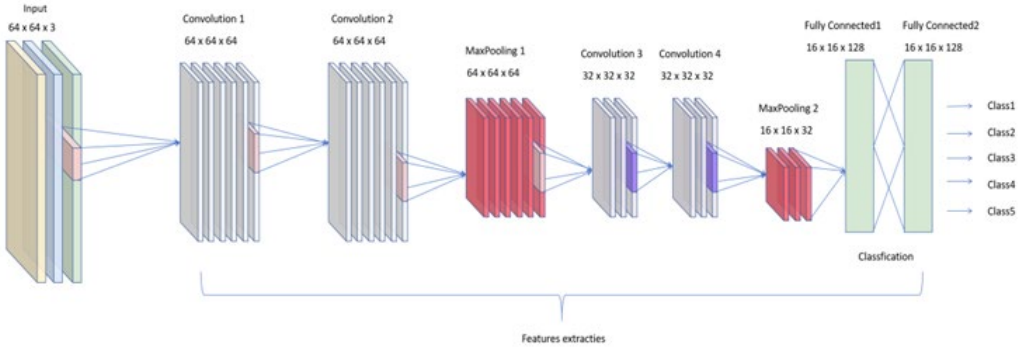

Figure 2: System Overview Diagram

The following are introductions to these components:

3.1.1. Convolutional Neural Network (CNN) Layers

The convolutional layer is the core part of the neural network, responsible for extracting useful features from the input data. Compared to the currently widely used deep neural networks (DNNs), CNNs can significantly reduce model size while maintaining performance[6]. These convolutional kernels learn and optimize during training to extract the most useful features for speaker recognition. The convolution layer architecture in this project is "Conv3-32x4", which means the convolution kernel size is 3x3, and the number of output channels is 32. The subsequent four layers will use the output of this layer as their input source, performing convolution operations on the input data and outputting 32 feature maps.

A convolutional layer contains several different convolvers, observing various local features of the speech. The pooling layer reduces the number of input nodes for the next layer by aggregating the output nodes of the convolutional layer with a fixed window length, controlling the model's complexity. Generally, the pooling layer uses the Max Pooling algorithm, selecting the maximum value within the fixed window length as the output[6]. Finally, the fully connected layer aggregates the pooling layer's output values to obtain the final classification decision. This structure has achieved superior performance in image processing[7].

3.1.2. Pooling Layer

The pooling layer follows the convolutional layer, primarily performing downsampling to reduce data dimensionality while retaining important features. This helps reduce computation, improve model generalization, and prevent overfitting. We adopted the Max Pooling method, which selects the maximum value within the pooling window as the output, effectively extracting the most prominent features. The default pooling window size in this project is 2x2.

3.1.3. Fully Connected Neural Network (FCN) Layer

After processing through multiple convolutional and pooling layers, the data is flattened and fed into the fully connected neural network layer. This layer integrates the features extracted earlier and learns to map these features to specific speaker identities through training. The neurons in the fully connected layer connect to all neurons in the previous layer, capturing complex relationships between different features. This project includes 512 neurons in the fully connected layer (Figure 3).

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

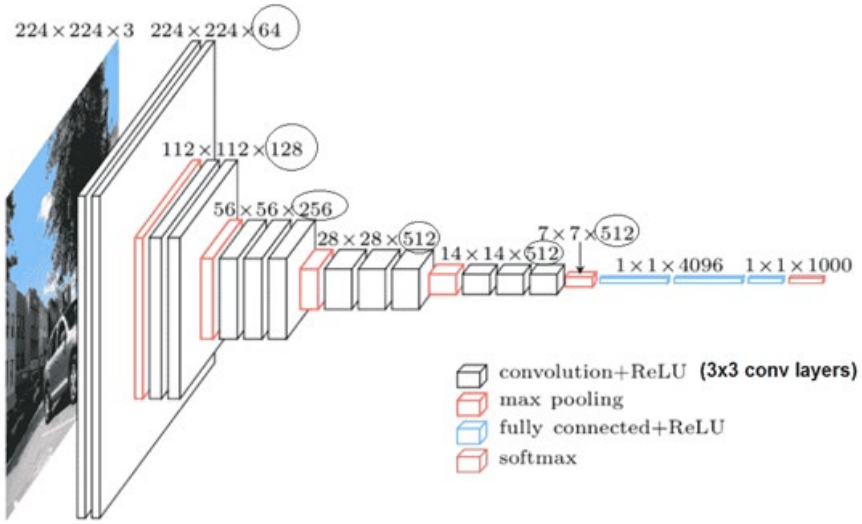

Figure 3: FCN Diagram

3.2. System Optimization

To improve model performance and generalization capability, we adopted several optimization techniques in the system design, including regularization and Dropout.

3.2.1. Regularization

Regularization is a technique to prevent model overfitting by adding a penalty term to the loss function. It achieves enhanced model generalization by sparsifying network parameters or augmenting auxiliary data, thereby preventing overfitting[7]. We used the L2 regularization method, encouraging the model's weight parameters to remain small. By adjusting the value of the regularization coefficient, we can control the penalty for model complexity, finding a balance point where the model can adequately fit the training data while maintaining good generalization capability.

3.2.2. Dropout

Dropout is an effective technique to prevent overfitting, randomly setting the output of some neurons to zero during training. Dropout processing in convolutional neural networks is a method to effectively avoid excessive number of parameters in the network[8]. This prevents the model from over-relying on certain neurons or features during training, thereby improving the model's robustness. We applied the Dropout technique after the fully connected layer, adjusting the Dropout rate to control the proportion of neurons set to zero. During the testing phase, all neurons are retained, but their outputs are multiplied by a factor related to the Dropout rate to ensure the output expectation is consistent with the training phase.

3.2.3. FBank

Filter Bank (FBank) is a feature extraction method used in speech signal processing. It primarily converts time-domain speech signals into frequency-domain representations, effectively transforming and representing the frequency information of speech signals. An increase in the filter bank leads to an increase in the number of training samples, which is beneficial to the CNN training[9] and representing the frequency information of speech signals. Through convolution and pooling operations, CNNs can extract useful patterns and features from FBank features, improving overall recognition performance.

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

4. Experimental Tests

4.1. Preparation of experimental data

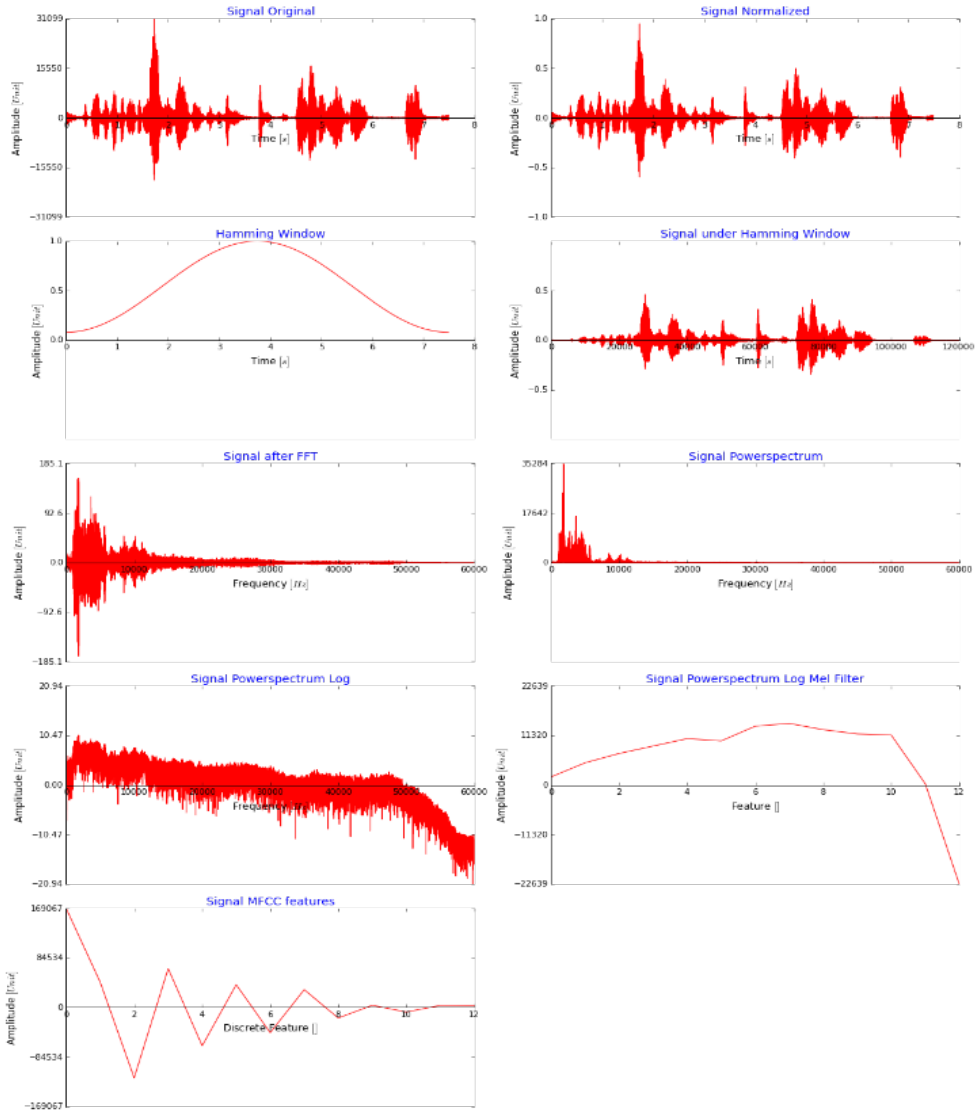

Figure 4: An example spectrum chart of the feature extraction process.

For system training and validation purposes, an array of diverse audio datasets were utilized encompassing member recordings as well as audial resources sourced from online platforms. This comprehensive dataset augmentation facilitated enhanced adaptability of our system across varied real-world scenarios. During our project's nascent phase, preliminary acoustic information essential for modeling was gathered via individual member contributions alongside web-sourced sound bites with prompt collaboration among team members ensuring swift acquisition of requisite initial acoustic inputs crucial for modeling endeavors. Subsequent efforts focused on extracting Linear Predictive Coding (LPC) features; however suboptimal results prompted exploration of superior feature extraction methodologies online culminating in adoption of Mel-Frequency Cepstral Coefficients (MFCC) processing[10]. A randomized seed was employed to shuffle both feature sets along with their respective labels ensuring uniformity thereby enhancing model robustness against sequence-based biases during training while bolstering its generalization capabilities. Following MFCC coefficient acquisition, partitioning into 4:1 ratioed train-test sets ensued culminating in generation of a mat file paving way for integration with Convolutional Neural Networks facilitating subsequent modeling exercises (Figure 4).

4.2. Introduction to the Experimental Procedure

The experimental process is as follows: As shown in the Figure 5, the main process of this experiment

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

is as follows:

Collecting speech audio: The audio is collected by recording the members and collecting it online. Eight different speakers' audio is obtained, with an average duration of 55 minutes per audio. The audio is preprocessed beforehand to eliminate noise and retain human voice (using the processing method of "VR Architecture" in the software "ultimate vocal remover 5", and the VR model "4_HP-Vocal-UVR"). A specific function is used to read each audio file and cut it into multiple segments. Each segment is exported as a new .wav file based on a specified time interval and step size and stored in a different folder according to different labels. All extracted MFCC features and labels are randomly scrambled and then divided into a training set and a test set in a certain ratio (usually 4:1).

Please note that the above information is subject to change without notice. Training and testing samples:

The training samples are divided into three steps:

MFCC feature extraction loads audio files (usually stored in .wav or .mp3 format), extracts MFCC features from the original audio signal, reads the sampling rate and signal of the audio file, performs preprocessing steps on the audio data, such as noise reduction and normalization, to ensure the clarity and consistency of the signal. Applies a window function (such as Hamming window) to each frame of the signal to reduce the influence of boundary effects on the spectral analysis[11]. Performs a fast Fourier transform (FFT) on the windowed signal to convert the time domain signal to the frequency domain. Converts the spectrum to the Mel frequency domain using a set of Mel filters, applies a discrete cosine transform (DCT) to the log Mel spectrum to obtain a set of Mel frequency cepstral coefficients (MFCC), used to represent the audio features.

The CNN is trained to process the audio file, extracts MFCC features, and converts them to image format, then randomly shuffles the data to prepare for training. Divides the extracted MFCC features into training set and testing set. Each sample is usually accompanied by a label indicating which category the sample belongs to. Selects an optimizer (such as Adam), a loss function (such as cross-entropy), and an evaluation metric (such as accuracy). This step compiles the model's computational graph for efficient training. Uses the training set to train the model. In each training iteration, the model adjusts its weights to minimize the loss function.

Build a CNN model library According to different task requirements, create multiple CNN model structures. Save these model structures as files or serialized formats (such as JSON) for easy management. Load predefined model structures from files when needed for initialization, training, or inference - convenient for reuse and improving work efficiency.

Test samples are divided into two steps:

MFCC feature extraction After the training is over, use the test set to evaluate the model to obtain the final performance indicators (such as accuracy, precision, recall, etc.).

Build a CNN model Use the CNN model trained in the training stage to predict the test samples and evaluate the accuracy of the model.

Matching degree calculation Matching degree calculation refers to comparing the similarity between the test samples and the training samples (or categories). It mainly includes the following steps:

Extract feature vectors Use the trained CNN model to predict each test sample and obtain a probability distribution, representing the probability of each class for the sample.

Extract feature vectors Extract the output of a certain layer in the CNN model as the feature vector. Usually, the output before the last fully connected layer is chosen as the feature vector.

Calculating the cosine similarity is one of the most commonly used methods, where the similarity score is used to determine the degree of matching between the test sample and the training sample or class. If the similarity score is high, it is assumed that the test sample is highly matched with the given class. The formula for calculating the cosine similarity is as follows:

$$\text{Cosine Similarity} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|} \quad (1)$$

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

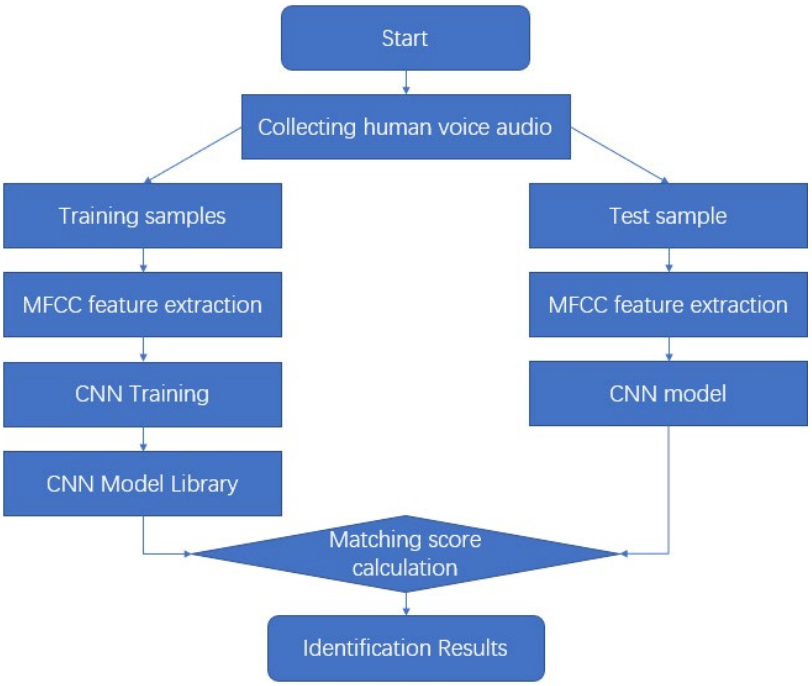

Figure 5: A diagram of the experimental procedure

4.3. Results and Analysis of Experiments

During the initial stages of system design, our convolutional neural network (CNN) model was relatively rudimentary, consisting of just one convolutional layer followed by a fully connected layer. This initial design led to suboptimal training performance. As a result, we sought to enhance our CNN model to achieve improved results. Ultimately, we developed a CNN model comprising three convolutional layers, two pooling layers, and one fully connected layer. We also implemented additional enhancements such as regularization and dropout techniques to effectively address overfitting issues and bolster the model's generalization capabilities while improving its stability[12] [13]. The resulting training outcomes were notably more favorable. The following data graph illustrates the output obtained from sampling every 200 instances out of a total of 101,200 during training:

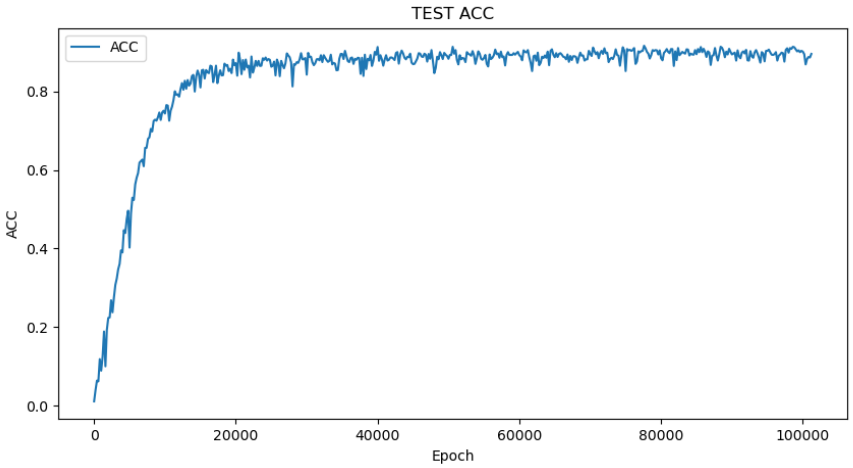

Figure 6: Test set accuracy rate

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

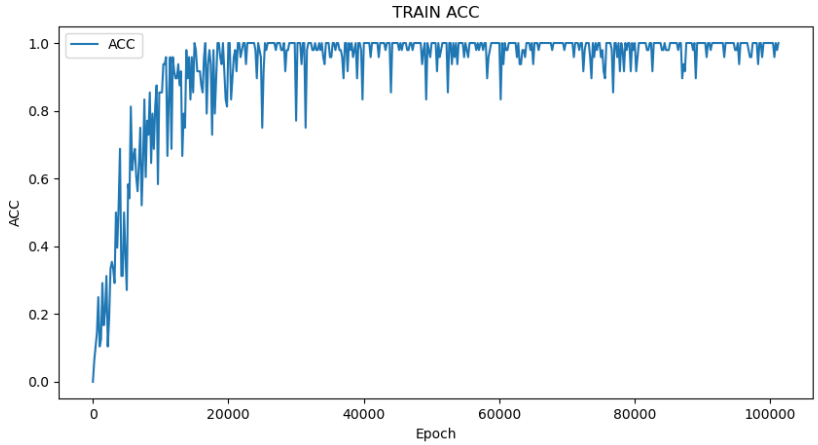

Figure 7: Training set accuracy rate

The figure above shows the accuracy of the test set (Figure 6) and the training set (Figure 7). It can be observed from both figures that the accuracy of the training set increases with the number of training iterations, reaching its maximum value after approximately 20,000 iterations. After multiple rounds of training, the model demonstrates good performance on both the training and test sets.

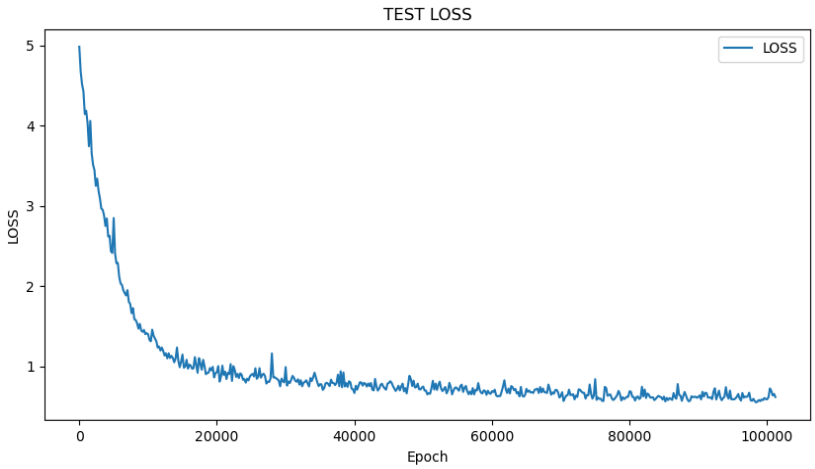

Figure 8: Test set loss rate

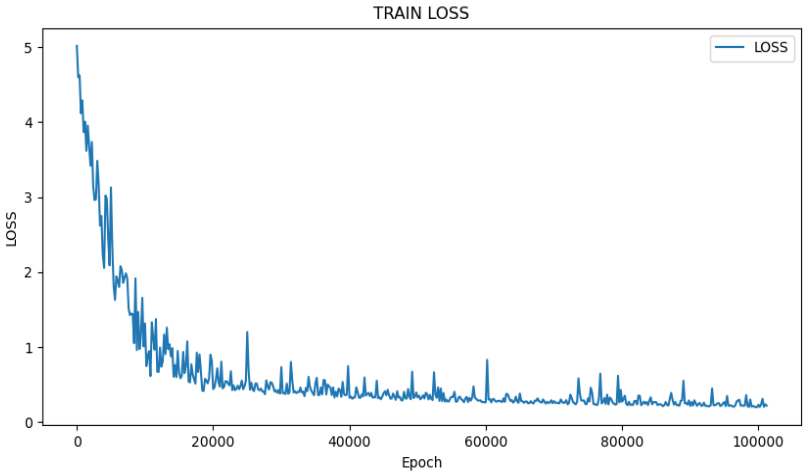

Figure 9: Training set loss rate

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

The graph presents the loss rate for both the test set (Figure 8) and the training set (Figure 9). It is apparent from both graphs that with an increase in the number of training iterations, the loss rate of the training set diminishes, reaching its minimum value after around 20,000 iterations. After undergoing multiple rounds of training, the model demonstrates robust performance on both sets. Our continuous refinement and design efforts have led to the development of a proficient and accurate speaker identification system capable of extracting vital signal features from a speaker's voice and performing speaker identification using a CNN-based deep learning model. In future studies, we will persist in refining and enhancing this system to further bolster its adaptability and resilience.

Table 1: Results of multiple experiments

Result	TRAIN		TEST	
	Loss	Acc	Loss	Acc
1-CNN, 1-FCN	0.35	0.93	0.43	0.65
1-CNN, 1-FCN (Dropout)	0.37	0.98	0.44	0.68
1-CNN, 1-FCN (Optimize)	0.31	0.96	0.40	0.72
Result	TRAIN		TEST	
	Loss	Acc	Loss	Acc
2-CNN, 1-Pooling, 1-FCN	0.28	0.98	0.38	0.71
2-CNN, 1-Pooling, 1-FCN (Dropout)	0.28	0.96	0.36	0.77
2-CNN, 1-Pooling, 1-FCN (Optimize)	0.27	1.00	0.35	0.81
Result	TRAIN		TEST	
	Loss	Acc	Loss	Acc
3-CNN, 2-Pooling, 1-FCN	0.24	0.97	0.32	0.78
3-CNN, 2-Pooling, 1-FCN (Dropout)	0.24	1.00	0.28	0.85
3-CNN, 2-Pooling, 1-FCN (Optimize)	0.20	0.98	0.31	0.89

The Table 1 presents the three stages of our CNN model design: an initial stage featuring a single CNN layer, a middle stage comprising one FCN layer and two CNN layers, as well as a pooling layer, and a final stage consisting of three CNN layers, two pooling layers, and one FCN layer. Additionally, Dropout was implemented in each stage to optimize the final outcome.

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

Table 2: Results of Initial stage experimental

Result	TRAIN		TEST	
	Loss	Acc	Loss	Acc
1-CNN, 1-FCN	0.35	0.93	0.43	0.65
1-CNN, 1-FCN (Dropout)	0.37	0.98	0.44	0.68
1-CNN, 1-FCN (Optimize)	0.31	0.96	0.40	0.72

Following the final optimization, we observed a significant increase in the accuracy of the training set during the initial stage; however, the performance of the test set was suboptimal. Consequently, after optimizing and enhancing the CNN model, we progressed to the middle stage[14] (Table 2).

Table 3: Results of Middle stage experimental

Result	TRAIN		TEST	
	Loss	Acc	Loss	Acc
2-CNN, 1-Pooling, 1-FCN	0.28	0.98	0.38	0.71
2-CNN, 1-Pooling, 1-FCN (Dropout)	0.28	0.96	0.36	0.77
2-CNN, 1-Pooling, 1-FCN (Optimize)	0.27	1.00	0.35	0.81

The training set continues to exhibit a high accuracy rate, with a decrease in the loss rate compared to the initial stage. Additionally, there has been an increase in the accuracy rate and a decrease in the loss rate of the test set. However, despite these improvements, the performance is still suboptimal. Consequently, we persist in optimizing and enhancing the CNN model as we progress towards the final stage[15] (Table 3).

From the table data, we can see that the loss rate of the training set has a significant reduction compared to the initial stage, while the accuracy remains high. The loss rate of the test set has also improved compared to the initial stage, and the accuracy of the test set is the final precision. It can be seen that after we modify and optimize the CNN structure, the final accuracy of our deep learning model has been significantly improved (Table 4).

Table 4: Results of Final stage experimental

Result	TRAIN		TEST	
	Loss	Acc	Loss	Acc
3-CNN, 2-Pooling, 1-FCN	0.24	0.97	0.32	0.78
3-CNN, 2-Pooling, 1-FCN (Dropout)	0.24	1.00	0.28	0.85
3-CNN, 2-Pooling, 1-FCN (Optimize)	0.20	0.98	0.31	0.89

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

5. Conclusion

Through the design and development of a speaker recognition system based on MFCC feature extraction and CNN architecture, we have achieved the following gains:

1) Principles of MFCC feature extraction: In the audio feature extraction stage, we improved the feature extraction method and learned the advantages and principles of MFCC feature extraction, making our system's experimental results more ideal.

2) Composition and function of CNN architecture: In the system design stage, we understood the composition and functions of the CNN architecture, and we optimized the system by adding convolutional layers and pooling layers. We also adopted various optimization techniques, including regularization and dropout, to prevent overfitting of the model.

3) Design thinking: Through initial design and continuous optimization and improvement, we developed our design thinking in the system design.

Although we have achieved certain results in our project research, there are still some unresolved issues and future research opportunities, including:

1) Future application scenarios: Our design direction for this project is a lightweight speech recognition system, aimed at serving speech recognition scenarios in the future and applied to scenarios with low power requirements and low consumption.

2) Optimizing System Structure: In the future, we will further optimize the system structure, improve the system's shortcomings, enhance its stability and versatility, and make our system lighter, reducing the system's consumption and power consumption during operation.

Put in a nutshell, our future research projects will focus on solving the above problems in order to achieve a more comprehensive and in-depth understanding of MFCC feature recognition combined with CNN.

References

[1] Yu, M., Yuan, Y., Dong, H., & Wang, Z. (2006) Text-dependent speaker recognition method using MFCC and LPCC features, Journal of Computer Applications, 26.04: 883-885.
[2] Tiwari, V. (2010). MFCC and its applications in speaker recognition. International journal on emerging technologies, 1(1), 19-22.
[3] Juanhong, L., Yu, H., & Heyu, H. (2020) End-To-End Speech Recognition Based On Deep Convolution Neural Network, Computer Applications and Software, 37.4: 192-196.
[4] Feng, C., & Cheng, W. (2023) Speech Recognition Algorithm Based on Residual Convolutional Neural Network, Computer and Digital Engineering, 51.2
[5] Huy, P., Fernando, A., Navin, C., Y Oliver, C., & Maarten De, V. (2018) DNN Filter Bank Improves 1-Max Pooling CNN for Single-Channel EEG Automatic Sleep Stage Classification., The IEEE Engineering in Medicine and Biology Society, 453-456.
[6] Zhang, Q., Liu, Y., Pan, J., & Yan, Y. (2015) Continuous speech recognition by convolutional neural networks, Chinese Journal of Engineering: 1212-1217.
[7] LeCun, Y., Huang, F. J., & Bottou, L. (2004, June). Learning methods for generic object recognition with invariance to pose and lighting. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. (Vol. 2, pp. II-104). IEEE.
[8] Chen, K., & Wang, A. (2024). Survey on regularization methods for convolutional neural network. Computer Applications Research, 04, 961-969.
[9] O-Yeon, K., Min-Ho, L., Cuntai, G., & Seong-Whan, L. (2020) Subject-Independent Brain–Computer Interfaces Based on Deep Convolutional Neural Networks, IEEE Transactions on Neural Networks and Learning Systems, 31.10: 3839-3852.
[10] Mirco, R., & Yoshua, B. (2018) Speaker recognition from raw waveform with sincnet, 2018 IEEE Workshop on Spoken Language Technology (SLT 2018), abs/1808.00158: 1021-1028.
[11] Can, C., & Yingcai, Y. (2014) Application of Window Function in Signal Processing, Journal of Beijing Institute of Graphic Communication: 71-74, 77. doi:10.3969/j.issn.1004-8626.2014.04.029.
[12] Wang, Y. (2021). Research of Speech Recognition Model based on Convolutional Neural Network And its Training Optimization. Chongqing University of Posts and Telecommunications. doi:10.27675/d.cnki.gcydx.2021.000406
[13] Zhao, X., & Zhang, K. (2022) Speech recognition based on three-layer structure optimized

ISSN 2616-5775 Vol. 7, Issue 7: 47-59, DOI: 10.25236/AJCIS.2024.070707

convolutional neural network, Journal of Shihezi University: Natural Science Edition, 40.1: 127-132.
[14] Chang-zheng, L., & Lei, Z. (2016) Research on Optimization Algorithm of Convolution Neural Network in Speech Recognition, Journal of Harbin University of Science and Technology, 21.3: 34-38.
[15] Zhichao, W., Ji, X., Pengyuan, Z., & Yonghong, Y. (2018) Structure optimization and computing acceleration for convolutional neural network acoustic models, Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 30.3: 416-422.