

Deep Hashing Method for Content-based Speech Retrieval Using Time Sequence Speech Visualization Feature

Ying-Jie Hu, Qiu-Yu Zhang*, and Qi-Wen Zhang

School of Computer and Communication, Lanzhou University of Technology,
Gansu, China

modaoshiyan@163.com

Accepted 17 December 2024; Revised 25 March 2025; Accepted 8 May 2025

Abstract. A deep hashing method using a time sequence speech visualization feature was proposed to improve speech retrieval accuracy, the efficiency of deep learning, and the noise robustness of speech deep hashing. The speech data is transformed into spectrogram time sequences. Two deep learning models, the 3D CNN (Convolutional Neural Networks)-BiLSTM (Bidirectional Long Short-Term Memory) deep model, and the bidirectional ConvLSTM deep model, are constructed to learn from the time sequence speech visualization feature to generate deep hashing for speech full-text retrieval. Experimental results demonstrate that the proposed models only require fewer iterations to achieve satisfied training accuracy. Moreover, the time sequence speech visualization feature can effectively represent speech content to achieve high retrieval accuracy and recall. Additionally, the proposed deep hashing exhibits high robustness to noise. The proposed method has higher accuracy and is robust to speaker identity compared to existing methods.

Keywords: speech retrieval, deep hashing, time sequence speech visualization feature, 3D CNN-BiLSTM, bidirectional ConvLSTM

1 Introduction

Content-based speech retrieval is a Query-By-Example (QBE) technique that has become more important for explosive multimedia information and increasing demand for intelligent applications. It can search for associated files by a spoken query from a massive volume of voice messages, voice records from conferences, spoken commands, and other audio data. Some methods convert speech into text using automatic speech recognition (ASR) to search from spoken audio data [1], but its accuracy is greatly reduced by word error rate (WER) [2]. Many researchers have introduced hash technology [3] even deep hashing [4-5] to learn from speech features to improve accuracy and complexity. However, its results are greatly affected by the speaker's biometric characteristics, tempo, speech length, etc., often only achieving accurate retrieval of the same speech, making it difficult to achieve keyword retrieval, fuzzy retrieval, and full-speech retrieval. To achieve QBE based on speech hashing, we propose a new time sequence speech visualization feature combined with a Chinese pronunciation model to generate hashes. This approach mitigates the adverse effects of biometric characteristics and other factors. It not only enables the retrieval of the same speech from the same speaker but also allows for the retrieval of speech from different speakers based on content. Furthermore, the generated hashes can grow with the length of the actual speech content, eliminating the limitation of speech length. This enables keyword retrieval and full-speech retrieval.

In both content-based speech retrieval schemes, constructing a low-complexity, high-performance deep learning model is the core to improve retrieval accuracy and noise robustness. The most widely used LSTM model features a memory gate that allows for learning relatively long-term relationships between time series frames through selective memory. It is particularly suitable for speech, video, and so on [6]. The ordinary LSTM can only predict the next frame by learning the previous and the current. However, the pronunciation, the word, and the phrase often have a deep relationship with the context of before and after in natural language. Therefore, the BiLSTM model, which has two sets of LSTM units with a forward propagation algorithm and a backward propagation algorithm, is introduced to learn from the input sequence to obtain the context feature [7]. In addition, because speech signals can be converted into a waveform or log-mel spectrograms, many researchers have intro-

* Corresponding Author

duced the CNN model, which is widely used in image processing, to learn the visual features of speech [8]. They apply 1D CNN to obtain the deep features of speech from raw signals, the Mel Frequency Cepstral Coefficient (MFCC), and so on [9]. Its convolution operation can ensure the translation invariance of images, significantly reduce the influence of audio volume and speaker identity in speech visualization images, and improve accuracy by extracting more representative semantic features [10]. Learning efficiency has also improved because of the lower amount of data in the operation.

CNN can learn spatial features and has fewer parameters than LSTM, but its ability to learn temporal features is weaker. LSTM can learn from temporal changes, but its complexity is high. Many studies have combined CNN and LSTM to improve learning from local features and contextual relationships [11]. Especially in the 2D CNN-LSTM model, convolutional operations combined with LSTM for sequential data can further improve accuracy [12]. However, the input data of the 2D CNN model requires converting the entire speech into a single image, and the image size must be consistent to ensure accuracy. However, speech length varies with different tempos and content, which weakens the ability to learn from temporal characteristics and limits the practicality of content-based speech retrieval. Subsequently, video and other time series data research have applied the 3D CNN model to learn image sequences, which can better extract the temporal features between images combined with LSTM.

Although the CNN-LSTM model can combine the advantages of two deep models, the input of the traditional LSTM unit can only be a one-dimensional sequence, making it more suitable for the original sampled speech data in fields such as speech recognition. The output of the CNN layer must be flattened before being sent to the LSTM, making it difficult to learn spatial and temporal features. In addition, the LSTM units are fully connected, which increases the complexity of model learning. Therefore, the ConvLSTM expands the input into a matrix, and takes the convolution operation between units, which enables it to learn spatial changes in time series [13]. It has been widely applied in spatial prediction [14].

In summary, current recognition-based speech QBE schemes have limited accuracy, while most hashing-based speech retrieval schemes struggle to achieve QBE for different speakers. LSTM models using MFCC as input can learn from speech of arbitrary length, but they have many learning parameters and low training efficiency. Currently, CNN-based retrieval schemes have high learning efficiency, but they can only learn from fixed-length speech and perform poorly when used alone in the speech domain. Given this, we propose a deep hashing method using a time sequence speech visualization feature for speech QBE. This method combines a Chinese pronunciation model with the proposed time sequence speech visualization feature to achieve content retrieval for speech from different speakers. We have constructed 3D CNN-BiLSTM and bidirectional CovLSTM to learn from the proposed time sequence speech visualization feature. It solves CNN's problem of fixed speech length, enables keyword retrieval, reduces the number of training iterations, and enhances deep hashing's semantic representation ability and noise robustness.

The main contributions are as follows:

- 1) We proposed a novel time-series visual feature extraction method for speech data that accommodates variable-length inputs while preserving contextual relationships between sequential visualization features.
- 2) We designed a 3D CNN-BiLSTM architecture for learning deep semantic representations from the time sequence speech visualization feature, enabling the construction of content-aware deep hashing suitable for speech QBE.
- 3) We developed a bidirectional CovLSTM -based feature learning framework that accelerates model convergence and improves training efficiency.

The rest of this paper is organized as follows. Section 2 introduces the related methods and research status of speech retrieval and deep hashing. Section 3 presents the basic knowledge of this work, including the basic principles of speech retrieval based on deep hashing, the construction of deep hashing, and the Chinese pronunciation model. Section 4 gives the proposed time sequence speech visualization feature extraction, the construction of the 3D CNN-BiLSTM speech visualization feature learning model, the bidirectional CovLSTM speech visualization feature learning model, and content-based time sequence speech visualization deep hashing. Section 5 provides the details of the experiment to verify its effectiveness. Section 6 gives the conclusion and prospects.

2 Related Works

2.1 Content-based Speech Retrieval

Speech retrieval systems based on ASR usually retrieve text documents after speech recognition, improving retrieval performance through additional features [15] or improved index [16]. Their response time is short because the retrieval process is text retrieval, but the retrieval accuracy is lower for the adverse impact of WER [17]. In recent years, content-based multimedia retrieval has widely adopted hash-based retrieval schemes. Subsequently, many researchers have introduced deep learning to construct deep hashing to obtain deeper semantic features of speech to improve retrieval accuracy, such as Zhang et al. [18] using the deep hashing of the specified length of speech, which enhances the accuracy and noise robustness of speech retrieval. However, it can only retrieve the same content with the same speaker, which is more suitable for speaker identity authentication than QBE retrieval. The speed and duration of speech also significantly impact retrieval results, and requirements of speech length are strict. Yuan et al. [5] introduced attention mechanisms into deep learning models to learn binary hash codes from the bottleneck features of speech, it achieved fast QBE by calculating the Hamming distance between the query hash sequence and the speech hash sequence in the database.

2.2 Speech Deep Learning Model

Speech recognition and retrieval systems typically preprocess speech signals through frame segmentation, windowing, and transformation operations. These processes extract physical and perceptual speech features as deep learning inputs to reduce computational demands. The most used is MFCC [19]. In addition, El Haj [20] used MFCC and a Gaussian multi-way latent block model (GMWLBM) to detect the emotion of the speech. Zhao et al. [21] employed CNN to learn from the speech Log-Mel spectrum, utilizing 40-dimensional Mel coefficients per frame as input representations. This approach reduces model complexity while enabling deeper architectures that enhance contextual learning capabilities. While feature-based approaches significantly decrease data dimensionality and improve computational efficiency, they exhibit notable limitations. They are sensitive to noise, speaker-dependent physiological characteristics, pitch, tempo, etc. The features extracted from the same content will change greatly when conditions change (the speaker, audio volume, playback rate, background noise, etc.), leading to degraded model robustness. Perceptual features like MFCC and spectrograms employ Mel-scale filters to align original speech frequency with human auditory perception (Equation (1)), thereby improving noise robustness. However, they remain vulnerable to temporal variations in speech rate, pitch, and playback rate. Furthermore, standalone CNN or LSTM architectures face inherent constraints: CNNs exhibit limited temporal modeling capacity, while LSTMs suffer from high computational complexity in sequential processing.

$$M(f) = 1125 \times \ln(1 + f / 700) , \quad (1)$$

Where f is the original frequency of the speech.

In deep learning architectures, multimodal fusion approaches have demonstrated superior performance in temporal pattern recognition. Guo et al. [22] combined CNN and LSTM to learn the spatial and temporal features from the time frames of WiFi signals, improving human actions' recognition accuracy and robustness. This architecture paradigm has been adapted for speech that first learns localized spectral patterns before modeling global temporal relationships via LSTM. However, such approaches face computational challenges due to the large raw speech sampling data and learning complexity. Compared to 1D CNN-LSTM, the 2D CNN-LSTM architectures offer distinct advantages for speech processing. By representing acoustic features as time-frequency matrices (analogous to images), 2D convolutions enable efficient dimensionality reduction before LSTM-based contextual relationships modeling. This framework can improve recognition accuracy, and mitigate the influence of speaker identity [1]. Extending this concept, 3D CNN-LSTM networks introduce spatiotemporal convolution kernels that simultaneously process multiple sequential feature maps across the width, height, and temporal dimensions. This architecture preserves inter-frame patterns through 3D convolutions, making it particularly effective for video analysis applications. Akilan et al. [23] proposed a 3D CNN-LSTM model to learn from video image sequences to achieve foreground object segmentation. Permana et al. [24] used 3D CNN-LSTM to learn from RGB video, achieving error action recognition on playing the erhu musical instrument. Despite these advancements, critical

limitations persist in 3D CNN-LSTM implementations for speech processing. The inherent dimensional mismatch between 3D convolutional outputs (spatiotemporal tensors) and LSTM inputs (1D sequences) necessitates feature flattening operations. That increases the complexity of model learning and prevents the effective utilization of spatial variation features among images.

Shi et al. [25] pioneered ConvLSTM architecture, enhancing traditional LSTM units by replacing matrix multiplication with convolution operations that preserve spatial relationships in image data. This innovation effectively captures spatial-temporal patterns in visual sequences, particularly excelling at modeling spatial feature variations across time steps. Subsequently, video deep learning began to apply the ConvLSTM. Majd et al. [26] introduced the ConvLSTM to learn from consecutive frames in the video, achieving human action recognition and improving the recognition accuracy compared to the traditional LSTM model. Qiao et al. [27] used ConvLSTM to extract spatio-temporal features from videos to recognize cow behaviours, and Arbelle et al. [28] utilized ConvLSTM to achieve object segmentation in microscope videos.

2.3 Challenges and Contribution

In summary, ASR-based speech retrieval systems can achieve speech QBE with lower retrieval accuracy. Most deep hashing-based speech retrieval systems offer higher accuracy, yet fail to meet the requirements of speech QBE. Few can realize speech QBE, its retrieval precision and robustness to noise require improvements. Moreover, the deep hashing method using CNN has strict requirements on speech length, while in practical applications, speech lengths vary and are difficult to uniform. The deep hashing method using RNN (Recurrent Neural Network) and LSTM can handle temporal features, but cannot directly handle multidimensional features, and the training process is highly complex. In addition, the robustness of the MFCC is limited. Consequently, we proposed a deep hashing method using a time sequence speech visualization feature, which can extract visualized temporal features from speech data, and convert speech into time sequence images with unified size. This method is no longer constrained by speech length, the generated hashing can expand as the speech content increases. It can retrieve speeches from different speakers to achieve speech QBE. Combined with the proposed 3D CNN-BiLSTM and bidirectional CovLSTM deep model to learn from the time sequence speech visualization feature, the number of iterations required for model training is reduced, and the constructed speech deep hash has stronger robustness to noise and speaker identity.

3 Background Knowledge

3.1 Speech Retrieval based on Deep Hashing

The process of the speech retrieval system based on deep hashing is shown in Fig. 1, which includes three main components: data owner/server, query user/client, and cloud server. First, manual features of the query speech are extracted as deep model input to learn deep features. The output is then converted into deep hashing codes. The most frequently used input features for deep learning are MFCC, Filter bank (Fbank), etc., combined with RNN, LSTM, and CNN to obtain deep semantic features. The deep hash of the query speech is uploaded to the cloud for subsequent retrieval. The data owner applies identical processing to speech files in the database, generating corresponding deep hashing codes, and constructing a hash index. This creates a bijective relationship where each speech file exclusively corresponds to its deep hashing in the index. Then upload both the index and original files to the cloud. The cloud server matches the query speech's deep hashing codes with those in the hash index table using algorithms like Hamming distance, Manhattan distance, and cosine similarity measurement. After calculating the mathematical distance for successful matching, the retrieval result is fed back to the query user.

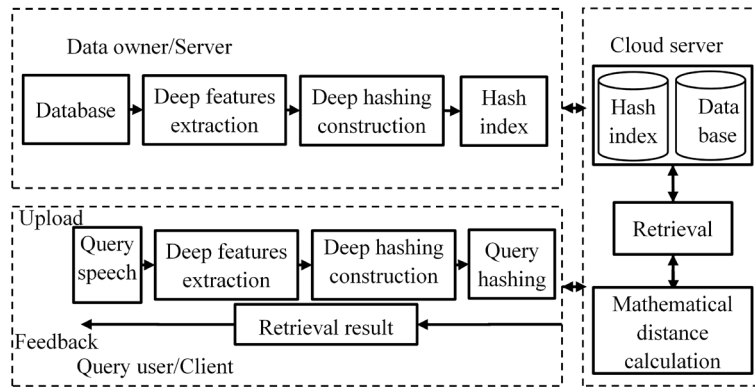


Fig. 1. The frameworks of hash-based speech retrieval

3.2 Deep Hashing Construction

Deep hashing is one of the indispensable techniques in multimedia data (graphic, audio) retrieval. This approximate nearest neighbor search method encodes multimedia data such as images and audio into compact binary codes, ensuring that similar samples share comparable hash codes while dissimilar ones exhibit distinct representations. However, traditional manual features cannot capture the semantic features of speech, so many researchers have introduced deep learning to combine hash coding with deep neural networks. The generated deep hashing preserves semantic information from raw audio and images, enhancing multimedia retrieval accuracy. Typical implementations employ RNN, LSTM, and CNN to build speech-oriented hashing frameworks.

As specialized variants of multilayer perceptrons, CNN is a deep feedforward architecture incorporating convolutional operations. The convolution kernels mimic biological visual cortex cells, processing localized spatial information through receptive fields. The CNN model excels in representation learning with hierarchical translation-invariant classification capabilities. Their multidimensional input processing is particularly suited for extracting pixel-level image features. A standard CNN architecture comprises convolutional layers (using kernels for feature extraction) and pooling layers (replacing single-point features with neighborhood statistics via downsampling to retain critical patterns). Fig. 2 illustrates the CNN-based speech deep hashing method. For speech processing, CNN typically accepts speech feature matrix as input, applying 2D convolution kernels to capture semantic features. These learned features undergo hashing functions to generate compact speech representations. Unlike images with standardized dimensions, speech signals require length normalization through a fixed-length truncation to meet CNN’s uniform input requirements.

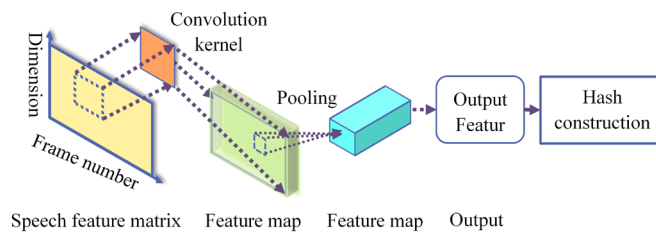


Fig. 2. CNN- based speech deep hashing construction

Unlike CNN, RNN is a model oriented to sequence that recursively evolves in the direction of sequence evolution, with all nodes being chain-connected recurrent neural networks. RNN can model natural language or other time series signals, making it more suitable for learning in areas such as speech and audio. The core of RNN is

a directed graph, each physical RNN unit can be expanded in time series, receiving the input at the current time point and the output at the previous time point at each time point. These recurrent units form a chain connection, which is also fully connected. The bidirectional RNN contains at least two layers, one for forward propagation, and the other for backward propagation, which can obtain more contextual information [7]. Let $\vec{\mathbf{a}}_t$ be the forward propagation RNN unit at time t , $\overleftarrow{\mathbf{a}}_t$ be the backward propagation RNN unit at time t . The forward algorithm calculation formula is shown in Equation (2), and the backward propagation algorithm is shown in Equation (3). Finally, the output \mathbf{y}_t can be obtained through Equation (4).

$$\vec{\mathbf{a}}_t = f(\omega_1 \times \chi_i + \omega_2 \times \overleftarrow{\mathbf{a}}_{t-1} + \vec{b}), \quad (2)$$

$$\overleftarrow{\mathbf{a}}_t = f(\omega_3 \times \chi_i + \omega_4 \times \vec{\mathbf{a}}_{t-1} + \overleftarrow{b}), \quad (3)$$

$$\mathbf{y}_t = \tanh(\omega_5 \times \vec{\mathbf{a}}_t + \omega_6 \times \overleftarrow{\mathbf{a}}_t), \quad (4)$$

Where \vec{b} and \overleftarrow{b} are the bias terms, and ω_* is the weight.

The standard RNN unit can only memorize short-term sequences, while LSTM [29] can learn the contextual relationship between sequences with long-time distance. LSTM is a gated algorithm of RNN, which contains three gates: input gate, forget gate and output gate. The three gates form a self-loop within a unit. The input gate determines the input at the current time point and the update of the internal state by the system state at the previous one; The forget gate updates the internal state at the current time point according to that at the previous one; The output gate updates the internal state by the system state. The LSTM states are fully connected, and the update method is shown in Equation (5), where F_t is the forget gate, C_t and \tilde{C}_t are memory cells, H_t is the hidden state, I_t and O_t are input and output gates, X_t is the current input, b_* is the bias term, \times represents matrix multiplication, and \circ represents Hadamard product.

$$\begin{aligned} I_t &= \sigma(\mathbf{W}_{xi} \times \mathbf{X}_t + \mathbf{W}_{hi} \times \mathbf{H}_{t-1} + \mathbf{W}_{ci} \circ \mathbf{C}_{t-1} + b_i), \\ F_t &= \sigma(\mathbf{W}_{xf} \times \mathbf{X}_t + \mathbf{W}_{hf} \times \mathbf{H}_{t-1} + \mathbf{W}_{cf} \circ \mathbf{C}_{t-1} + b_f), \\ \tilde{C}_t &= \tanh(\mathbf{W}_{xc} \times \mathbf{X}_t + \mathbf{W}_{hc} \times \mathbf{H}_{t-1} + b_c), \\ C_t &= F_t \circ \mathbf{C}_{t-1} + I_t \circ \tilde{C}_t, \\ O_t &= \sigma(\mathbf{W}_{xo} \times \mathbf{X}_t + \mathbf{W}_{ho} \times \mathbf{H}_{t-1} + \mathbf{W}_{co} \circ \mathbf{C}_{t-1} + b_o), \\ H_t &= O_t \circ \tanh(C_t). \end{aligned} \quad (5)$$

3.3 Chinese Initial Consonant and Vowel Pronunciation Model

Chinese speech recognition systems demand extensive training data to achieve sufficient accuracy. This necessity stems from Mandarin's phonetic structure: each syllable combines 23 initial consonants and 39 final vowels with four lexical tones, creating over 1,000 possible classifications. Furthermore, as a pictographic language, identical phonetic-tonal combinations may correspond to dozens of semantically unrelated Chinese characters. These linguistic complexities result in elevated error rates for text-based speech retrieval systems.

The pronunciation model PM [30] includes 21 standard consonants of modern Chinese Pinyin: b p m f d t n l g k h j q x zh ch sh r z c s, and two zero initials “y, w” included in the customary spelling. However, due to some words such as “阿”, “偶”, “饿” and so on, which only have vowels without standard consonants, the three vowels “a”, “o”, and “e” are also added to the model, with a total of 26 consonants and vowels completely covering all modern Chinese pronunciations, and representing all pronunciation words in all speech segments. In addition, there are also many silent segments, breath sounds, background noise, etc. in speech, so one type is set up to represent other speech information unrelated to pronunciation words.

The pronunciation model is not for precisely recognizing speech, instead, it should cover potential pronunciations, ensuring that deep hashing incorporates semantic meaning. This enables a more effective segmentation of variable-duration speech into meaningful segments, boosting the robustness of deep hashing against variations in tone, tempo, and speaker identity, to fulfill keyword-based fuzzy retrieval of full speech.

4 The Proposed Deep Hashing Construction Method

4.1 Time Sequence Speech Visualization Feature

Definition 1 (Speech Visualization Sequence SVS): Let the number of frames in speech s be m , and draw an RGB three-channel color spectrogram image G_i with a specified resolution for each frame i . The spectrogram sequence of speech is $SVS = \{G_1, \dots, G_i, \dots, G_m\}$.

Speech spectrograms are typically generated through preprocessing, windowing, framing, and Fast Fourier Transform (FFT). The horizontal axis represents time, and the vertical axis frequency and color depth correspond to amplitude. In practical speech retrieval systems, variable-duration speech segments produce spectrograms of inconsistent widths incompatible with CNN input requirements. While duration truncation would address this, it severely limits real-world applicability. Our solution converts speech into fixed-size spectrogram sequences whose lengths adapt to the original audio duration. This approach satisfies CNN input constraints while accommodating arbitrarily long speeches. Fig. 3 details the time sequence speech visualization feature extraction workflow.

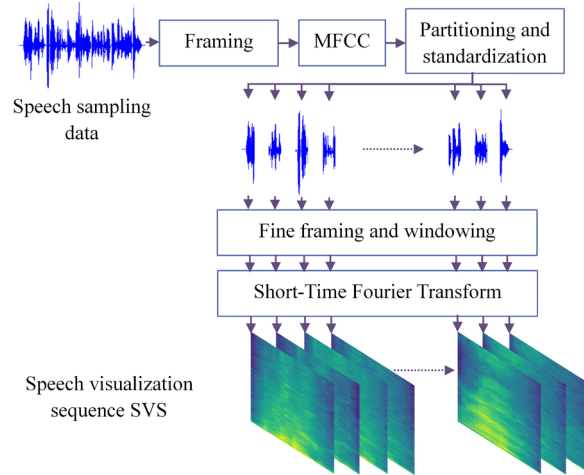


Fig. 3. Steps for extracting SVS

The specific processing steps for extracting the time sequence speech visualization feature are as follows:

Step 1: First, divide the speech s into m frames containing α sampling data. Set frame hop length be β sampling data. If the last frame of sampling data is insufficient for α , fill it with 0.

Step 2: Apply the Hanning window function to each frame to mitigate the effects of spectrum leakage.

Step 3: Divide each sampling value within the frame by the maximum value of the frame to standardize the data and reduce the influence of factors such as volume.

Step 4: Segment the normalized data of each frame according to the window size α' and overlap size γ , and perform the short-time Fourier transform to obtain the spectrogram G_i with a size of $w \times h$, $G_i = \{R, G, B, C\}$. R, G, B are the red, green, and blue color components, respectively, and C is the number of channels.

Step 5: Arrange each spectrogram in order of time to get the time sequence speech visualization feature SVS.

4.2 3D CNN -BiLSTM Time Sequence Speech Visualization Feature Learning Model

The proposed deep hash scheme applies the pronunciation model PM that contains only 26 consonants and vowels, $PM = \{[b], [p], [m], [f], [d], [t], [n], [l], [g], [k], [h], [j], [q], [x], [zh], [ch], [sh], [r], [z], [c], [s], [y], [w], [a], [o], [e]\}$. All Chinese pronunciation words must contain one of these consonants or vowels. The label set of deep learning is C , $C = \{c \mid 1 \leq \delta \leq 27\}$, $\delta \in \mathbb{N}$. When $\delta \leq 26$, $\forall c_\delta \in PM$, c_{27} is a class unrelated to meaningful pronunciation, including silence, breath sounds, background noise, etc.

The proposed 3D CNN-BiLSTM deep learning model employs three 3D convolutional layers to extract spatiotemporal features from the visualized sequence of speech. A fully connected layer then temporally unrolls these features for BiLSTM-based temporal features learning. The complete framework is illustrated in Fig. 4.

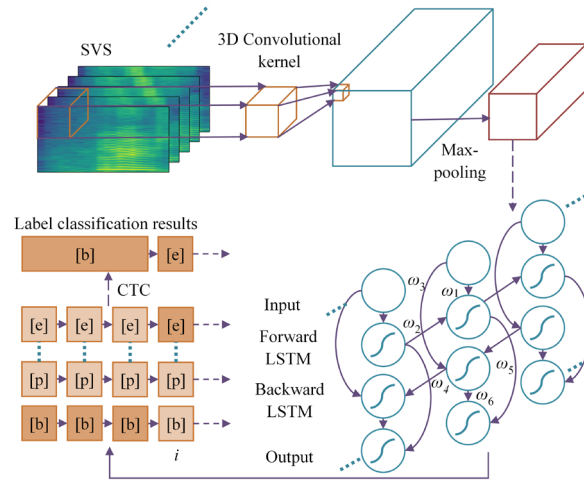


Fig. 4. The proposed 3D CNN -BiLSTM learning model

Fig. 5 illustrates the architecture of the proposed 3D CNN-BiLSTM deep model. The model extracts spatial features from the speech visualization sequence through three 3D convolutional layers. The visualization sequence of speech s is a temporal sequence of m spectrograms with dimensions $w \times h$. The 3D convolution kernels operate across both spatial dimensions and temporal frames, capturing spatiotemporal variations throughout the sequence. Dimensionality reduction pooling layers after each convolution stage preserve translational invariance to suppress the impact of speech volume.

The output of the fully connected layer is 2×512 , and the activation functions are all ReLU20 functions, as shown in Equation (6), to limit the activation range and promote sparse feature learning.

$$\text{ReLU } 20(x) = \min(\max(x, 0) 20) \in [0, 20]. \quad (6)$$

Divide the output of the fully connected layer into two sets of χ and χ' , and send them to two LSTM units for forward propagation and backward propagation operations, as shown in Equation (7).

$$\begin{aligned} \bar{\mathbf{h}}_i &= f(\omega_1 \times \chi_i + \omega_2 \times \bar{\mathbf{h}}_{i-1} + \bar{\mathbf{b}}), \\ \bar{\mathbf{h}}_i &= f(\omega_3 \times \chi_i + \omega_4 \times \bar{\mathbf{h}}_{i-1} + \bar{\mathbf{b}}), \\ \gamma_i &= \tanh(\omega_5 \times \bar{\mathbf{h}}_i + \omega_6 \times \bar{\mathbf{h}}_i), \end{aligned} \quad (7)$$

Where $\vec{\mathbf{h}}_i$ is the hidden state of the forward propagating LSTM unit at time point i , $\overleftarrow{\mathbf{h}}_i$ is the hidden state of the backward propagating LSTM unit, \vec{b} and \overleftarrow{b} are the bias terms, and ω_* is the weight, γ_i is the final output.

Use the Softmax classification layer to map 512 input features to label set C . The loss function applies the connectionist temporal classification (CTC) beam search, shown in Equation (8). Given the input data and output label set, identify valid candidate paths and output the optimal solution through maximum likelihood selection.

$$P(\mathbf{Y}|\mathbf{X}) = \sum \prod_{i=1}^m P_i(c_i | \mathbf{X}), \quad (8)$$

Where \mathbf{X} is the input SVS, \mathbf{Y} is the classification output result, and $P_i(c_i | \mathbf{X})$ is the probability distribution of the input corresponding to the label set C at the i th frame.

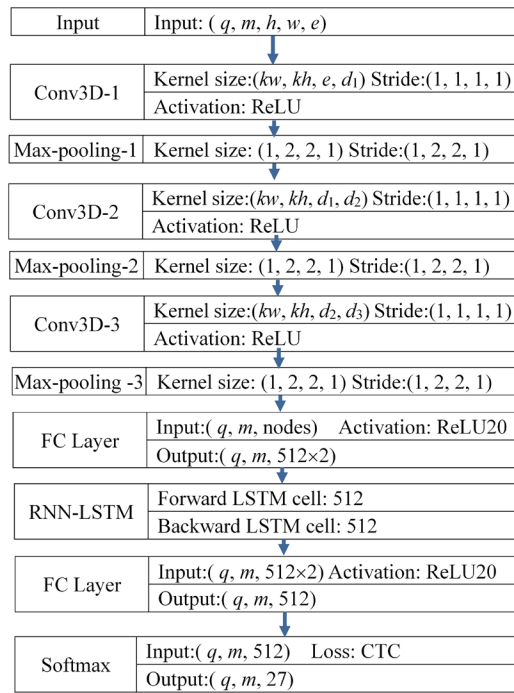


Fig. 5. The Structure of the proposed 3D CNN-BiLSTM

The length of the input SVS often exceeds the cardinality of class set C . Additionally, multiple images of SVS may belong to the same pronunciation segment, and the speaker's tempo variations, induce inconsistent duration-to-content mapping. These ambiguous alignments increase the error rate in full-speech retrieval. The CTC enables length-agnostic label prediction by permitting intermediate null outputs, effectively resolving variable-length template matching constraints.

4.3 Bidirectional ConvLSTM Time Sequence Speech Visualization Feature Learning Model

The proposed bidirectional ConvLSTM time sequence speech visualization feature learning model is shown in Fig. 6. It uses the ConvLSTM unit with the convolutional operation instead of traditional fully connected LSTM units, as shown in Equation (9), to better learn spatiotemporal features.

$$\begin{aligned}
 \mathbf{I}_t &= \sigma(\mathbf{W}_{xi} \otimes \mathbf{X}_t + \mathbf{W}_{hi} \otimes \mathbf{H}_{t-1} + \mathbf{W}_{ci} \circ \mathbf{C}_{t-1} + b_i), \\
 \mathbf{F}_t &= \sigma(\mathbf{W}_{xf} \otimes \mathbf{X}_t + \mathbf{W}_{hf} \otimes \mathbf{H}_{t-1} + \mathbf{W}_{cf} \circ \mathbf{C}_{t-1} + b_f), \\
 \mathbf{C}'_t &= \tanh(\mathbf{W}_{xc} \otimes \mathbf{X}_t + \mathbf{W}_{hc} \otimes \mathbf{H}_{t-1} + b_c), \\
 \mathbf{C}_t &= \mathbf{F}_t \circ \mathbf{C}_{t-1} + \mathbf{I}_t \circ \mathbf{C}'_t, \\
 \mathbf{O}_t &= \sigma(\mathbf{W}_{xo} \otimes \mathbf{X}_t + \mathbf{W}_{ho} \otimes \mathbf{H}_{t-1} + \mathbf{W}_{co} \circ \mathbf{C}_{t-1} + b_o), \\
 \mathbf{H}_t &= \mathbf{O}_t \circ \tanh(\mathbf{C}_t),
 \end{aligned} \tag{9}$$

Where \mathbf{F}_t is the forget gate, \mathbf{C}_t and \mathbf{C}'_t are memory cells, \mathbf{H}_t is the hidden state, \mathbf{I}_t and \mathbf{O}_t are input and output gates, \mathbf{X}_t is the current input, b_s is the bias term, and unlike standard LSTM units, the weight \mathbf{W}_{x*} is convolution kernel, \otimes represents convolution operation, and \circ represents Hadamard product.

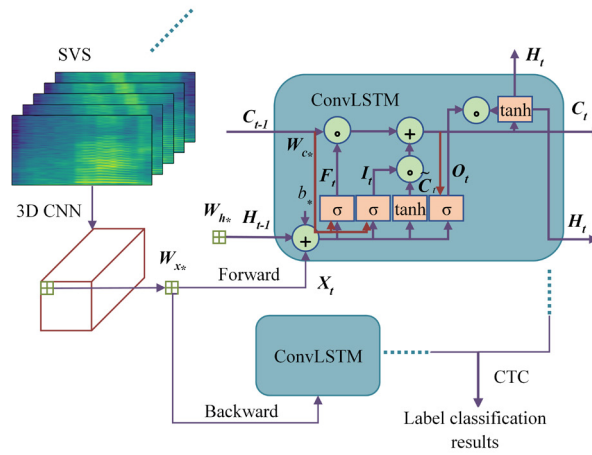


Fig. 6. The proposed bidirectional ConvLSTM learning model

The proposed bidirectional ConvLSTM time sequence speech visualization feature learning model uses two sets of ConvLSTM units for forward and backward propagation. As the output from the three-layer 3D CNN does not require flattening and can serve directly as input to ConvLSTM, the fully connected FC layer before the RNN-LSTM layer is removed. The configuration for the remaining layers in the model aligns with the 3D CNN-BiLSTM model described in the previous section, substituting the original RNN-LSTM layer exclusively with the ConvLSTM unit.

4.4 Content-based Deep Hashing Method Using Time Sequence Speech Visualization Feature

The processing steps of deep hashing construction are shown in Fig. 7, the details are as follows:

Step 1: Obtain the trained deep learning model output \mathbf{O} of speech s with m frames, $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_t, \dots, \mathbf{o}_m\}$. The \mathbf{o}_t is the feature sequence corresponding to the 27 labels the deep learning model learned in the t th frame, where $\mathbf{o}_t = \{y_1, y_2, \dots, y_{27}\}$. The larger the value of y_δ , the higher the similarity with the corresponding class in the label set C .

Step 2: Calculate $F(\mathbf{f}_t) = \max(\mathbf{f}_t)$ for each \mathbf{f}_t , and select all the consonants and vowels that can represent the meaningful word pronunciation. Retain the satisfied \mathbf{f}_t as a feature subset \mathbf{F}' , $\mathbf{F}' = \{\mathbf{f}_t | \text{Max}(\mathbf{f}_t) \neq y_{27}\}$, $1 \leq t \leq m$, y_{27} representing the non-consonant vowel class. This means removing the frames belonging to the noises, only saving the frame information corresponding to consonants and vowels.

Step 3: Merge duplicate frames. A Chinese pronunciation may last multiple frames, depending on the tempo or emotion. The same pronunciation in the query speech and the retrieved speech may have different numbers of

frames. To avoid affecting the retrieval accuracy, if two or more adjacent frames are mapped to the same consonant or vowel, these frames are considered to be the same pronunciation segment. Using Equation (10), merge multiple frames contained in a single pronunciation segment to obtain \mathbf{F}'' , $\mathbf{F}'' = \{f'_1, \dots, f'_i, \dots, f'_m\}$.

$$f'_i = \begin{cases} f_i, & \text{Max}(f_i) \neq \text{Max}(f'_{i-1}) \\ (f_i + f'_{i-1})/2, & \text{Max}(f_i) = \text{Max}(f'_{i-1}) \end{cases}, 1 \leq i \leq m \quad (10)$$

Step 4: Convert each f'_i in \mathbf{F}'' into a binary sequence. In chronological order, arrange the data y_1, y_2, \dots, y_{27} in each f'_i from smallest to largest, and the resulting sequence is $y_{(1)}, y_{(2)}, \dots, y_{(27)}$. Obtain the median $med = y_{(14)}$. The hash calculation scheme defined in Equation (11) produces speech hash sequences \mathbf{H} exhibiting Fisher-Snedecor distribution properties.

$$h(y_\delta) = \begin{cases} 0, & y_\delta \leq med \\ 1, & y_\delta > med \end{cases} \quad (11)$$

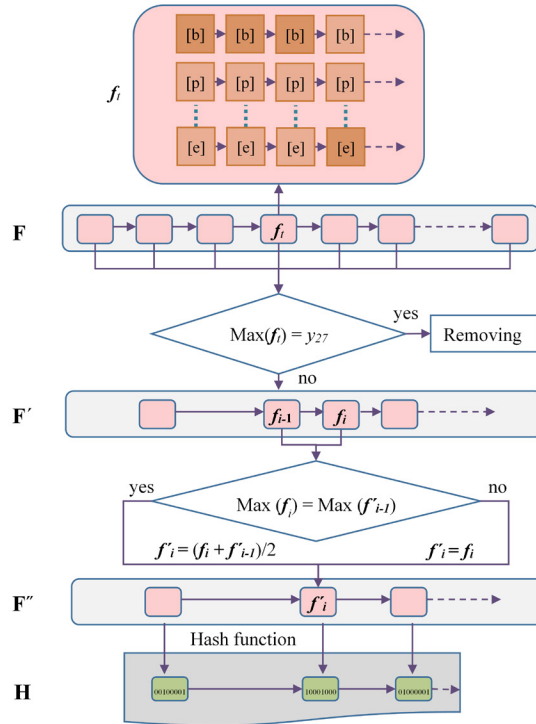


Fig. 7. Process flow for generating speech hash sequences

Instead of performing CTC on the output of deep learning and using the recognized labels, to calculate the hash, we directly use the output features of the deep model to calculate the hash code. This can reduce the impact of recognition performance on the representation ability of deep hashing to a certain extent, improve retrieval accuracy, and provide protection for information content.

Let \mathbf{H}_s and \mathbf{H}_x be the hash sequence of query speech and the one needing to match. If the matching result between \mathbf{H}_x and \mathbf{H}_s satisfies the threshold condition, add the corresponding speech file to the search result. The specific retrieval process is as follows:

Step 1: Group the two hash sequences H_x and H_s by every 27-bit hash code, each group corresponds to a label. Let H_x and H_s have k and q groups.

Step 2: Calculate the Hamming distance d_i between the h_{s1} (first group of H_s) and h_{xi} (one group of H_x), end matching when the number of groups in H_x that have not been compared is less than q .

Step 3: Compare d_i with the threshold γ . If it is greater than γ , continue to the next step. If not, return to step 2 to calculate the Hamming distance between h_{s1} and $h_{x(i+1)}$.

Step 4: Match subsequent groups in order and calculate the Hamming distance d_{i+j} between $h_{x(i+j)}$ (the j th group after h_{xi}) and h_{sj} .

Step 5: Calculate $\bar{d} = \frac{1}{j+1} \sum_0^j d_{i+j}$, where $j \leq k$.

Step 6: Compare \bar{d} with the threshold γ . If it is greater than γ , return to step 2 and continue to compare h_{s1} with the remaining groups of H_x . Otherwise, return to step 4 to match the next group.

Step 7: If the q groups of H_s have all been compared and still meet the threshold γ condition, then H_x is successfully matched.

5 Performance Evaluation and Analysis

We perform experiments on two databases. One is the THCHS30 Chinese Speech Database of Tsinghua University, with 10,875 speech samples in the A, B, and C groups (30 speakers) as training data, and 2,496 speech samples in group D (10 speakers) as the test set. Each group was about 250 news, sampled at 16 kHz and 256 kbps, and the duration of each speech was different.

The second database is the AISHELL-3 Chinese speech database, with a duration of 85 hours and 88,035 sentences. The recording was conducted in a quiet indoor environment with a sampling frequency of 44.1kHz and a sampling size of 16 bits. 218 speakers from different accent regions in China participated in the recording. We used 63,262 speeches as training data and 24,773 speeches as the test set.

All tests are programmed in Tensorflow-GPU 1.12.0 + Python 3.6 on a PC with an Intel(R) Core (TM) i7-6700 3.40 GHz CPU and 8.00 GB of main memory.

5.1 Comparison of Deep Model Training Results

We conducted a deep model's learning capacity experiment across different data scales. We tested three models, the bidirectional RNN-LSTM model [30] (baseline), the proposed 3D CNN-BiLSTM, and bidirectional ConvLSTM, on the THCHS30 and AISHELL-3, respectively. The training error rate is shown in Fig. 8. Both the proposed two models learn from the proposed time sequence speech visualization features, with parameter α containing 8,000 samples per frame and frame hop length β of 1,000 samples. The spectrogram has a length and width $w \times h$ of 155×77 for each frame in the THCHS30, while 77×38 in the AISHELL-3 due to the large amount of data.

Fig. 8(a) shows the training results on the THCHS30. Compared with the baseline model (using MFCC features as input), the proposed 3D CNN-BiLSTM and bidirectional ConvLSTM time sequence speech visualization feature learning model (using time sequence speech visualization features as input) have reduced the training error rate to below 10% after two iterations, the baseline model requires 20 iterations to achieve sufficient accuracy.

Fig. 8(b) shows the training results on the AISHELL-3 database. Due to the larger amount of speeches than THCHS30, the training error rates of the three deep models are lower. The baseline model has an error rate below 10% after 5 iterations, while the proposed two models have achieved 100% accuracy after one iteration.

The results show that the proposed time sequence speech visualization features can better represent the contextual representation of speech content. Combining the proposed 3D CNN-BiLSTM and bidirectional ConvLSTM time sequence speech visualization feature learning model can reduce the number of iterations and training data required for deep learning.

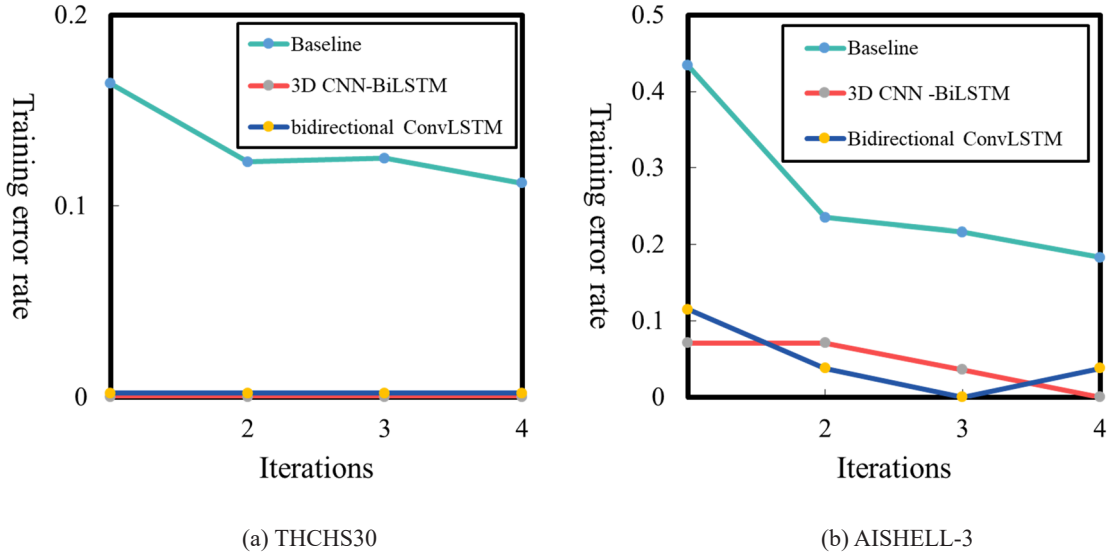


Fig. 8. The comparison of error rates in deep model training

5.2 Analysis and Comparison of Retrieval Performance of Deep Learning Models

To verify the effectiveness of the proposed time sequence speech visualization features, we use the deep hashing retrieval results based on the bidirectional RNN-LSTM model [30] as the baseline, with the MFCC features as the model input. The proposed 3D CNN-BiLSTM model and the bidirectional ConvLSTM model learn from time sequence speech visualization features. The retrieval results under the THCHS30 and the AISHELL-3 are shown in Table 1 and Table 2. For two databases, we all select 999 speeches from the training set and the test set as the query speech set, 13,356 speeches for retrieval in THCHS30, and 88,036 speeches for retrieval in AISHELL-3.

Ap is the average precision of all retrieval. $\text{Recall} = \frac{TP}{TP + FN}$, where TP is the true positive, and FN is the false negative. $TP + FN$ is 13,351 in THCHS30, due to the less similar content in the AISHELL-3 database, $TP + FN$ is 1,699.

Table 1. The retrieval results in THCHS30

Models	Threshold γ	Ap (%)	Recall (%)	Iterations
Baseline	0.320	97.68	47.60	21
3D CNN-BiLSTM	0.310	98.52	45.47	4
Bidirectional ConvLSTM	0.325	98.33	20.75	3

Table 2. The retrieval results in AISHELL-3

Models	Threshold γ	Ap (%)	Recall (%)	Iterations
Baseline	0.09	99.41	59.33	1
3D CNN-BiLSTM	0.07	99.40	58.80	1
Bidirectional ConvLSTM	0.07	99.01	58.98	1

As shown in Table 1, the baseline model needs to iterate 21 times to achieve ideal accuracy, while two proposed deep models using time sequence speech visualization features can achieve the same after 3-4 iterations only. Due to the larger number of speeches in AISHELL-3 than that in THCHS30, it can be seen from Table 2 that all three models achieve high accuracy and recall rate after only one iteration, and with the close values. The

experimental results show that the proposed time sequence speech visualization features have a strong representation ability for speech content, and can significantly reduce the number of iterations for deep model training.

5.3 Robustness Analysis of Proposed Speech Deep Hashing

We evaluated the robustness of the proposed deep speech hashing and added cafe, white, and car noise across SNR levels spanning 0-25dB to 999 speech queries in the THCHS30 database. The deep hashing was obtained from the query speech with the noise, using the deep model which had never been trained on noisy speeches, and retrieved in the original speech database. The deep hashing retrieval results based on the bidirectional RNN-LSTM model [30] were baseline. Table 3 shows the retrieval results of noise attack using the baseline model, proposed 3D CNN-BiLSTM model, and bidirectional ConvLSTM model, and compares them with the accuracy of adding white and car noise in the method of paper [31].

Table 3. Noise robustness comparison of deep hashing

Noise	Evaluation	Models	0dB	5dB	10dB	15dB	20dB	25dB
Cafe	Ap (%)	Baseline	97.27	97.73	97.99	97.99	97.84	97.69
		3D CNN-BiLSTM	98.15	98.32	98.56	98.63	98.47	98.42
		Bidirectional ConvLSTM	97.66	97.87	97.74	97.82	97.72	97.80
	Recall (%)	Baseline	32.88	42.33	44.54	44.62	45.18	46.89
		3D CNN-BiLSTM	36.51	38.13	40.38	42.18	42.38	42.39
		Bidirectional ConvLSTM	13.44	16.19	18.13	18.52	18.28	18.64
White	Ap (%)	Baseline	97.00	96.98	97.72	97.71	97.60	97.50
		3D CNN-BiLSTM	98.55	98.39	98.55	98.47	98.78	98.71
		Bidirectional ConvLSTM	97.76	97.45	97.71	97.12	97.78	97.36
		Paper [31]	58.20	62.00	72.50	88.40	-	100
	Recall (%)	Baseline	28.37	34.45	40.15	44.42	46.33	46.43
		3D CNN-BiLSTM	30.00	35.23	37.71	39.66	41.85	42.38
Car	Ap (%)	Baseline	97.96	97.70	97.73	97.81	97.68	97.93
		3D CNN-BiLSTM	98.78	98.60	98.52	98.51	98.64	98.60
		Bidirectional ConvLSTM	97.58	97.89	97.70	97.65	97.90	97.73
		Paper [31]	45.00	97.90	99.70	99.99	-	100
	Recall (%)	Baseline	46.09	47.14	47.14	47.40	46.93	47.02
		3D CNN-BiLSTM	44.17	44.37	43.99	43.20	43.46	43.30
		Bidirectional ConvLSTM	20.85	20.48	20.04	18.97	18.85	18.99

As shown in Table 3, noise attacks have little impact on accuracy using proposed deep hashing. The recall rate of the baseline model and the 3D CNN-BiLSTM model (time sequence speech visualization features as input) remains high when SNR is above 10 dB, and that of the bidirectional ConvLSTM model (time sequence speech visualization features as input) decreases slightly. After adding cafe noise, compared to the noise-free case, the recall rate of the baseline model decreases by 30.92% at 0 dB, and the decrease rate remains below 7% after 10 dB; The recall rate of the proposed 3D CNN-BiLSTM model is only 19.77% lower at 0 dB, and the reduction rate remains below 7% after 15 dB or more; Using proposed bidirectional ConvLSTM model, the recall rate decreased by 35.09% at 0 dB, and the decrease rate remained at around 10% above 15 dB. After adding white noise, compared to the noise-free case, the recall rate of the baseline model at 0 dB decreased by 40.34%, and the decrease rate remained below 7% above 15 dB; The recall rate of the proposed 3D CNN-BiLSTM model is 34.07% lower at 0 dB, and the reduction rate remains around 10% after 15 dB or more; The recall rate of the proposed bidirectional ConvLSTM model is 38.52% lower at 0 dB, and the reduction rate remained at around 20% after exceeding 15 dB. After adding car noise, compared to the noise-free case, the recall rate of the baseline model decreased by 3.16% at 0 dB, and the reduction rate remained at around 1% after exceeding 5 dB; The recall rate of the proposed 3D CNN-BiLSTM model at 0dB is 2.93% lower, and the overall reduction rate remains at around 3%; The recall rate of the proposed bidirectional ConvLSTM model from 0dB to 25dB is almost unchanged.

Compared with the method in the paper [31], our method demonstrates consistent accuracy metrics under noise attack. The proposed speech deep hashing is robust to noise, and the deep hashing of noisy speech main-

tains high retrieval accuracy and recall. Furthermore, the deep hashing generated by two proposed models using time sequence speech visualization features is more robust to noise than the deep hashing generated by the baseline model, indicating that the proposed time sequence speech visualization features can enhance noise robustness.

5.4 Effectiveness Analysis of the Proposed Time Sequence Speech Visualization Features

We perform ablation studies on the features and models to verify the effectiveness of the proposed time sequence speech visualization features. The model using MFCC features as input is M, and the model using the proposed time sequence speech visualization features is G. The model containing three layers of 3D CNN is C, and C^+ represents plus one more convolutional layer and C^- is minus a convolutional layer. BiLSTM is abbreviated as L, the bidirectional ConvLSTM model is CL. The retrieval uses 249 speech samples from A2 of the THCHS30 as query speech, and the retrieval range is 3,757 speech samples in the A set. The total number of $TP + FN$ should be 3,745. Table 4 lists the experimental results, which are the optimal values for each model.

Table 4. Results of the ablation study

Methods	Threshold γ	Iterations	Ap (%)	Recall (%)
G+C	0.30	5	99.17	25.61
G+C ⁻ +L	0.31	5	99.23	44.57
G+C ⁺ +L	0.30	8	99.78	48.25
M+C+L	0.32	3	97.31	38.66
M+C+CL	0.32	4	99.57	18.61
G+C+L	0.31	4	99.26	50.01
G+C+CL	0.33	3	99.79	25.39

Compared with the model using MFCC as the model input, the 3D CNN-BiLSTM model and bidirectional ConvLSTM model using the proposed time sequence speech visualization features have improved retrieval accuracy. In terms of recall rate, compared with the model using MFCC as the model input, the 3D CNN-BiLSTM model using the proposed time sequence speech visualization features has improved retrieval recall rate by 29.36%, and the bidirectional ConvLSTM model using the proposed time sequence speech visualization features has improved retrieval recall rate by 36.43%. The experimental results show that the proposed time sequence speech visualization features can effectively represent speech content, and the improvement in recall rate indicates that it is more robust to speaker identity and can further eliminate its adverse effects on retrieval.

Regarding model structure, the accuracy and recall rate of models that solely employ 3D CNN are notably low, particularly the recall rate, which is roughly half that of the proposed model. The results indicate that BiLSTM units can enhance the semantic learning capability of speech context. Moreover, adding a 3D CNN layer marginally improves retrieval accuracy but reduces the recall rate, which means that too many convolutional layers may lead to overfitting.

5.5 Comparison with Existing Methods

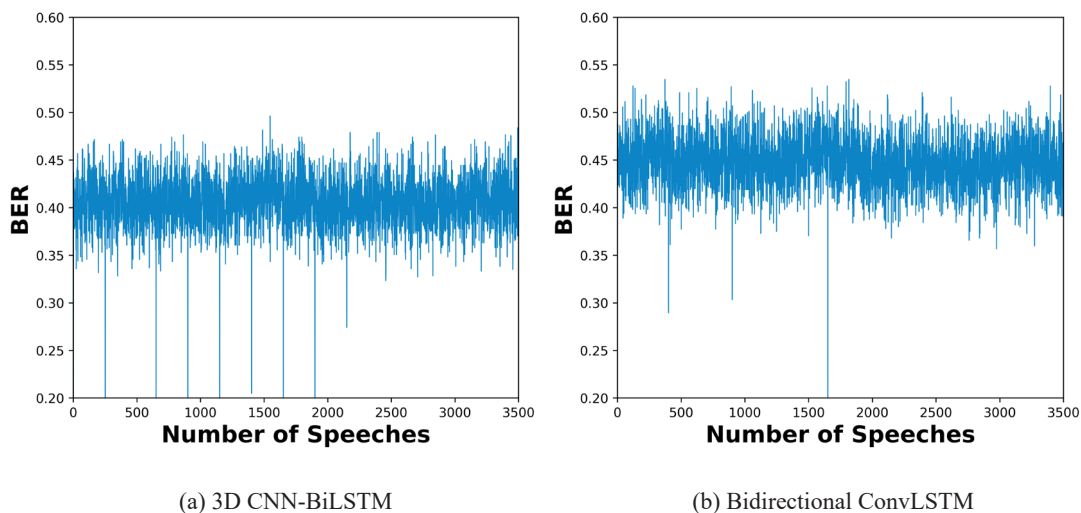
To verify the effectiveness of the deep hashing method based on the proposed time sequence speech visualization features, Table 5 presents the full-speech retrieval results of the proposed method, other deep hashing-based speech retrieval methods, and ASR-based methods [16-17]. The query speech set is 999 speeches from A2, B12, C14, and D21, the retrieval database is 13,356 speeches from THCHS30. ASR-based method [16-17] requires performing recognition first and then matching on the recognized text. For fairness, we use the baseline model (bidirectional RNN-LSTM models) and the pronunciation model PM for recognition. Retrieval based on deep hashing [18, 31] uses CNN models for training, and each speech sample generates a 512-bit hash code. The method of paper [5] can achieve QBE. The proposed method applies the results of the proposed 3D CNN-BiLSTM model with the time sequence speech visualization features as input. The number of bits generated from speeches with different lengths varies, a normal-speed speech of about 10 seconds can generate a 27×30 -bit hash code, and the retrieval threshold is 0.31. In the experiment, regardless of the speaker's identity, as long as the speech content is the same, it is considered a correct retrieval. Therefore, $TP + FN$ should be 13,351.

Table 5. Comparison of retrieval performance with existing speech retrieval methods

Models	Ap (%)	Recall (%)	Keyword search
ASR-based [16-17]	100.00	8.03	✓
Deep hashing-based [18, 31]	97.00	7.48	✗
Paper [5]	57.20	-	✓
Proposed method	98.52	45.47	✓

The ASR-based method successfully matched 1,072 speeches, all of which were true positive. However, the correct number of speeches with consistent content should be 13,351. Except for the same speech from the same speaker, the same content speeches from the other speakers could hardly be retrieved. The results showed that the ASR-based method struggles to avoid interference from speaker identity information with limited training data and is highly sensitive to the Word Error Rate (WER). The deep hashing methods [18, 31] also failed to retrieve speeches with the same content from different speakers and perform keyword searches. Although the approach in the paper [5] can retrieve keywords, its overall accuracy is relatively low. The proposed deep hashing method, leveraging time sequence speech visualization features, demonstrates superior retrieval performance. It maintains high accuracy and recall rates with fewer iterations, thereby enhancing the robustness of deep hashing against speaker identity interference.

We take one random speech from A of the THCHS30 as query speech, the retrieval range is 3,757 speech samples in the A set. We calculate the bit error rate (BER) between query speech and retrieval set, the matching results of the proposed two models are shown in Fig. 9.

**Fig. 9.** The matching results of proposed models

As shown in Fig. 9, our method can retrieve speeches of the same contents from other speakers, and the remaining BER are all greater than the threshold. The paper [18, 31] results indicate that a query speech can only retrieve the same speech from the database. The deep hashing generated by 3D CNN-BiLSTM has strong semantic representation ability, and that generated by bidirectional ConvLSTM has better discriminability. Overall, our method can achieve speech QBE and its deep hashing has better discriminability and representation ability.

We tested the ASR-based method [16-17], the proposed 3D CNN-BiLSTM method, and the Bidirectional ConvLSTM method using 10 and 15 keywords to test the keyword retrieval effectiveness. The query speech set is 999 speeches from A2, B12, C14, and D21, the retrieval database is 13,356 speeches from THCHS30. The retrieval results are shown in Fig. 10.

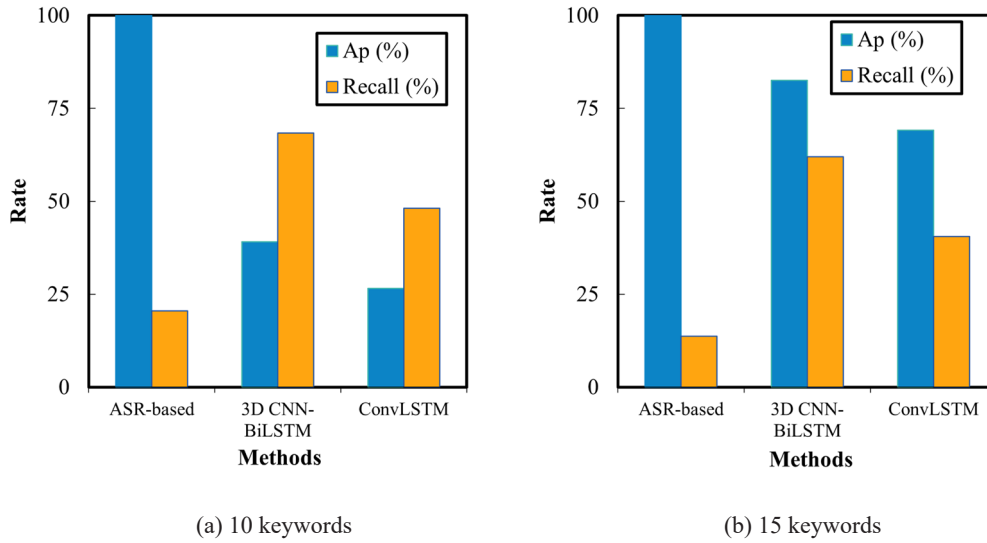


Fig. 10. The keywords search results

As shown in Fig. 10, the average precision of the ASR-based method [16-17] is close to 100%, but its recall rate is quite low due to the impact of WER. Its recall rate is only 20.52% when the number of keywords is 10 and decreases as the number of keywords increases. When the number of keywords is 10, the recall rate of the proposed 3D CNN-BiLSTM method is 68.38%, increased by 233.18% compared with the ASR-based method [16-17]. When the number of keywords is 15, the recall rate of the proposed 3D CNN-BiLSTM method is 61.95%, increased by 350.49% compared with the ASR-based method [16-17]. The recall rate of the proposed Bidirectional ConvLSTM method is increased by 134.34% (195.26%) compared with the ASR-based method [16-17] using 10 (15) keywords. The deep hashing-based method [18, 31] can only use speech queries of a specified length to retrieve speech data sets of the same length and does not support keyword retrieval. In summary, our methods enable keyword retrieval with varying numbers of keywords and significantly enhance the recall rate.

6 Conclusions

We propose the time sequence speech visualization feature to enhance deep learning performance and design two deep models, a 3D CNN-BiLSTM model and a bidirectional ConvLSTM model, to learn from the proposed feature. Experimental results demonstrate that the proposed time sequence speech visualization feature reduces the iterations required for model training, exhibits strong speaker identity robustness, and achieves superior retrieval accuracy and recall rates. Ablation studies confirm the method's effectiveness, while its noise robustness outperforms MFCC features.

This work has been proven to realize speech QBE and improve the efficiency of deep learning and the noise robustness of deep hashing. Future work will consider further optimizing the proposed time sequence speech visualization feature and its application in other fields such as speech recognition.

7 Acknowledgement

This work is supported by the *National Natural Science Foundation of China* (No. 61862041).

References

- [1] Z.Y. Wu, L.P. Yen, K.Y. Chen, Generating pseudo-relevant representations for spoken document retrieval, in: Proc. the 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
<https://doi.org/10.1109/ICASSP.2019.8683832>
- [2] C.H. Lee, Y.N. Chen, H.Y. Lee, Mitigating the impact of speech recognition errors on spoken question answering by adversarial domain adaptation, in: Proc. the 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
<https://doi.org/10.1109/ICASSP.2019.8683377>
- [3] F. Zou, X. Tang, K. Li, Y. Wang, J. Song, S. Yang, H. Ling, Hidden semantic hashing for fast retrieval over large scale document collection, *Multimedia Tools and Applications* 77(3)(2018) 3677-3697.
<https://doi.org/10.1007/s11042-017-5219-3>
- [4] L. Fan, Q.Y. Jiang, Y.Q. Yu, W.J. Li, Deep Hashing for Speaker Identification and Retrieval, in: Proc. 2019 proceedings of the Annual Conference of the International Speech Communication Association, 2019.
<http://dx.doi.org/10.21437/Interspeech.2019-2457>
- [5] Y. Yuan, L. Xie, C.C. Leung, H. Chen, B. Ma, Fast query-by-example speech search using attention-based deep binary embeddings, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28(2020) 1988-2000.
<https://doi.org/10.1109/TASLP.2020.2998277>
- [6] L. Jiao, R. Zhang, F. Liu, S. Yang, B. Hou, L. Li, New generation deep learning for video object detection: A survey, *IEEE Transactions on Neural Networks and Learning Systems* 33(8)(2021) 3195-3215.
<https://doi.org/10.1109/TNNLS.2021.3053249>
- [7] Q. Chen, G. Huang, A novel dual attention-based BLSTM with hybrid features in speech emotion recognition, *Engineering Applications of Artificial Intelligence* 102(2021) 104277.
<https://doi.org/10.1016/j.engappai.2021.104277>
- [8] Mustaqeem, S. Kwon, MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach, *Expert Systems with Applications* 167(2021) 114177.
<https://doi.org/10.1016/j.eswa.2020.114177>
- [9] Dhiraj, R. Biswas, N. Ghattamaraju, An effective analysis of deep learning based approaches for audio based feature extraction and its visualization, *Multimedia Tools and Applications* 78(17)(2019) 23949-23972.
<https://doi.org/10.1007/s11042-018-6706-x>
- [10] P. Vitolo, R. Liguori, L.D. Benedetto, A. Rubino, G.D. Licciardo, Automatic Audio Feature Extraction for Keyword Spotting, *IEEE Signal Processing Letters* 31(2024) 161-165.
<https://doi.org/10.1109/LSP.2023.3346280>
- [11] M.R. Ahmed, S. Islam, A.K.M.M. Islam, S. Shatabda, An ensemble 1D-CNN-LSTM-GRU model with data augmentation for speech emotion recognition, *Expert Systems with Applications* 218(2023) 119633.
<https://doi.org/10.1016/j.eswa.2023.119633>
- [12] Y. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, S. Chang, Modeling multimodal clues in a hybrid deep learning framework for video classification, *IEEE Transactions on Multimedia* 20(11)(2018) 3137-3147.
<https://doi.org/10.1109/TMM.2018.2823900>
- [13] W.S. Hu, H.C. Li, L. Pan, W. Li, R. Tao, Q. Du, Spatial-spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing* 58(6)(2020) 4237-4250.
<https://doi.org/10.1109/TGRS.2019.2961947>
- [14] Z. Lin, M. Li, Z. Zheng, Y. Cheng, C. Yuan, Self-attention convlstm for spatiotemporal prediction, Proc. the 34th Proceedings of the AAAI Conference on Artificial Intelligence 34(7)(2020) 11531-11538.
<https://doi.org/10.1609/aaai.v34i07.6819>
- [15] K.Y. Chen, S.H. Liu, B. Chen, H.M. Wang, A locality-preserving essence vector modeling framework for spoken document retrieval, in: Proc. the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017.
<https://doi.org/10.1109/ICASSP.2017.7953241>
- [16] A. Gupta, D. Yadav, A novel approach to perform context-based automatic spoken document retrieval of political speeches based on wavelet tree indexing, *Multimedia Tools and Applications* 80(14)(2021) 22209-22229.
<https://doi.org/10.1007/s11042-021-10800-8>
- [17] S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan, C. Glackin, A new secure and lightweight searchable encryption scheme over encrypted cloud data, *IEEE Transactions on Emerging Topics in Computing* 7(4)(2019) 530-544.
<https://doi.org/10.1109/TETC.2017.2737789>
- [18] Q.Y. Zhang, X.J. Zhao, Q.W. Zhang, Y.Z. Li, Content-based encrypted speech retrieval scheme with deep hashing, *Multimedia Tools and Applications* 81(7)(2022) 10221-10242.
<https://doi.org/10.1007/s11042-022-12123-8>
- [19] P. Vasquez-Serrano, J. Reyes-Moreno, R.C. Guido, A. Sepúlveda-Sepúlveda, MFCC Parameters of the Speech Signal: An Alternative to Formant-Based Instantaneous Vocal Tract Length Estimation, *Journal of Voice: Official Journal of the*

- Voice Foundation 39(6)(2025) 1431-1439.
<https://doi.org/10.1016/j.jvoice.2023.05.012>
- [20] A. El Haj, Emotions recognition in audio signals using an extension of the latent block model, *Speech Communication* 161(2024) 103092.
<https://doi.org/10.1016/j.specom.2024.103092>
- [21] Z. Zhao, Q. Li, Z. Zhang, N. Cummins, H. Wang, J. Tao, B.W. Schuller, Combining a parallel 2D CNN with a self-attention Dilated Residual Network for CTC-based discrete speech emotion recognition, *Neural Networks* 141(2021) 52-60.
<https://doi.org/10.1016/j.neunet.2021.03.013>
- [22] L. Guo, H. Zhang, C. Wang, W. Guo, G. Diao, B. Lu, C. Lin, L. Wang, Towards CSI-based diversity activity recognition via LSTM-CNN encoder-decoder neural network, *Neurocomputing* 444(2021) 260-273.
<https://doi.org/10.1016/j.neucom.2020.02.137>
- [23] T. Akilan, Q.J. Wu, A. Safaei, J. Huo, Y. Yang, A 3D CNN-LSTM-based image-to-image foreground segmentation, *IEEE Transactions on Intelligent Transportation Systems* 21(3)(2020) 959-971.
<https://doi.org/10.1109/TITS.2019.2900426>
- [24] A. Permana, T.K. Shih, A. Musdholifah, A.K. Sari, Error Action Recognition on Playing The Erhu Musical Instrument Using Hybrid Classification Method with 3D-CNN and LSTM, *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 17(3)(2023) 313-324.
<https://doi.org/10.22146/ijccs.76555>
- [25] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, W. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: *Proc. the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.
- [26] M. Majd, R. Safabakhsh, A motion-aware ConvLSTM network for action recognition, *Applied Intelligence* 49(7)(2019) 2515-2521.
<https://doi.org/10.1007/s10489-018-1395-8>
- [27] Y. Qiao, Y. Guo, K. Yu, D. He, C3D-ConvLSTM based cow behaviour classification using video data for precision livestock farming, *Computers and electronics in agriculture* 193(2022) 106650.
<https://doi.org/10.1016/j.compag.2021.106650>
- [28] A. Arbelle, S. Cohen, T.R. Raviv, Dual-task ConvLSTM-UNet for instance segmentation of weakly annotated microscopy videos, *IEEE Transactions on Medical Imaging* 41(8)(2022) 1948-1960.
<https://doi.org/10.1109/TMI.2022.3152927>
- [29] H. Naeem, A.A. Bin-Salem, A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images, *Applied Soft Computing* 113(2021) 107918.
<https://doi.org/10.1016/j.asoc.2021.107918>
- [30] Y. Hu, Q. Zhang, Q. Zhang, Y. Jia, A novel hashing-inverted index for secure content-based retrieval with massive encrypted speeches, *Multimedia Systems* 30(1)(2024) 22.
<https://doi.org/10.1007/s00530-023-01229-0>
- [31] Q. Zhang, J. Bai, F. Xu, A retrieval method for encrypted speech based on improved power normalized cepstrum coefficients and perceptual hashing, *Multimedia Tools and Applications* 81(11)(2022) 15127-15151.
<https://doi.org/10.1007/s11042-022-12560-5>