



Few-Shot Speaker Recognition Using Self-Supervised Speech Representations and a Transformer Encoder

Wei Li^{*ID}, Zhongming Yang

Computer Engineering Technical College, Guangdong Polytechnic of Science and Technology, Zhuhai 519000, China

Corresponding Author Email: livay_21@163.com

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.430435>

ABSTRACT

Received: 8 March 2026
Revised: 27 July 2026
Accepted: 17 August 2026
Available online: 31 August 2026

Keywords:

few-shot speaker recognition, self-supervised speech representation, Transformer encoder, Mockingjay, WavLM

Self-supervised speech representations (S3Rs) have been widely investigated for speaker recognition because they can capture informative acoustic characteristics from speech signals without relying on extensive manual annotations. However, their applicability to few-shot speaker recognition, where only a limited number of labeled speech samples are available for each speaker, remains insufficiently explored. This study proposes a few-shot speaker recognition framework that combines S3Rs with a Transformer encoder to investigate whether representations learned from unlabeled speech data can support speaker identification under limited-data conditions. Two pretrained speech representations, Mockingjay and WavLM, are separately employed as input features, while the Transformer encoder serves as the core component of the classification model to learn discriminative information from the extracted representations. The proposed framework is evaluated using both representations, and its recognition performance is compared with that obtained using conventional Mel-spectrogram features. The experimental results indicate that the framework achieves recognition accuracy values of 94.80% with Mockingjay representations and 92.29% with WavLM representations. The comparison further shows that S3Rs provide more effective features for few-shot speaker recognition than conventional Mel-spectrogram features under the evaluated experimental conditions. These findings suggest that combining pretrained speech representations with Transformer-based feature learning is a feasible approach to speaker recognition when labeled training samples are limited.

1. INTRODUCTION

Speaker recognition [1-3] is an important research area in speech signal processing and biometric identification. It aims to determine or verify a person's identity by analyzing the acoustic characteristics of speech signals. Compared with other biometric identification methods, such as fingerprint recognition and facial recognition, speaker recognition can be performed without requiring physical contact or specialized image acquisition equipment. Speech signals can also be collected through commonly available microphones and communication devices, making voice-based identity recognition suitable for a range of practical applications. Typical examples include voiceprint authentication for personal smart devices, secure access control in telephone banking, speaker identification in forensic investigations, and caller identity verification in customer service systems. As voice-controlled devices and remote communication services become increasingly common, reliable speaker recognition has become an important component of intelligent human-computer interaction and information security systems.

From the perspective of identity recognition objectives, speaker recognition can be divided into two main categories: speaker identification [4, 5] and speaker verification [6, 7]. Speaker identification aims to determine the identity of an

unknown speaker from a predefined set of enrolled speakers. The system analyzes an input speech utterance and assigns it to one of the registered speaker identities according to the extracted acoustic characteristics. In contrast, speaker verification determines whether a speech utterance belongs to a claimed speaker. Rather than selecting an identity from multiple candidates, the verification system evaluates the similarity between the input speech and the reference information associated with the claimed identity. Although these two tasks share several feature extraction and representation learning techniques, their decision procedures and evaluation criteria are different. Speaker identification is commonly evaluated using classification-based measures, whereas speaker verification generally relies on similarity scores and verification error measures.

Speaker recognition systems can also be categorized according to the types of information used for identification. Speech-based recognition methods rely exclusively on acoustic information [4, 7-9], while multimedia-based methods [10] combine speech signals with additional information, such as facial images or video sequences. Multimedia speaker recognition may use visual information to estimate the number of speakers appearing in a recording. Another approach constructs a joint audio-visual representation by combining speaker embeddings extracted

from speech with facial embeddings obtained from images. For example, an integrated multimedia vector, referred to as an M-vector, can be constructed by concatenating face embeddings and speech-based x-vectors with an appropriate weighting factor. Such methods make use of complementary information from different modalities. Nevertheless, the availability and quality of visual information may vary considerably across applications. In telephone conversations, audio recordings, and many voice-controlled systems, only speech signals are available. Speech-based speaker recognition therefore remains particularly relevant for applications in which visual data cannot be obtained.

Despite considerable progress in speaker recognition, the performance of existing systems may deteriorate when speech recordings are affected by environmental noise, reverberation, transmission distortion, or differences in recording devices [11-13]. The acoustic characteristics extracted from a speaker's voice may change under different recording conditions, even when the speech is produced by the same person. Such variations can make it difficult to distinguish speaker-specific information from factors unrelated to identity. In addition, speaker recognition systems can be affected by spoofing attacks involving replayed, synthesized, or converted speech [14, 15]. Conventional systems trained mainly on genuine speech samples may not adequately distinguish authentic speech from deliberately manipulated signals. Although noise robustness and spoofing detection are separate research problems, they indicate the broader difficulty of developing speaker representations that remain reliable under varying input conditions.

The performance of a speech-based speaker recognition system depends substantially on the quality of the extracted speech features and the ability of the classification model to distinguish among different speakers. Traditional approaches commonly employ acoustic features such as Mel-frequency cepstral coefficients, filter-bank energies, and Mel-spectrograms. These features describe different aspects of the spectral characteristics of speech and have been widely adopted in speech processing applications. However, conventional acoustic representations are primarily designed to characterize signal properties rather than to isolate speaker identity. They may retain information associated with phonetic content, speaking style, and recording conditions, which can interfere with speaker classification. Consequently, the selection of input features and the construction of effective classification models have long been important research directions in speaker recognition.

With the development of deep learning, speaker recognition research has increasingly focused on learning acoustic representations directly from speech data. Neural network architectures can extract complex characteristics from speech signals and construct representations that are more suitable for downstream recognition tasks. Nevertheless, the training of these models often requires a substantial amount of labeled speech data covering multiple speakers and recording conditions. In practical applications, collecting and annotating large speech datasets can be expensive and time-consuming. The difficulty becomes more pronounced when only a small number of speech samples can be collected for each target speaker. Under such conditions, the classification model may learn characteristics associated with individual training utterances rather than stable speaker-specific patterns, resulting in poor generalization to previously unseen speech samples.

Speaker recognition under limited labeled-data conditions is commonly investigated within the framework of few-shot speaker recognition [16, 17]. In this setting, the recognition system is expected to identify speakers using only a small number of labeled reference utterances. Few-shot speaker recognition [4] is particularly relevant to applications in which it is impractical to collect extensive enrollment recordings from each user. For instance, a newly registered user of a voice-controlled device may provide only a few short speech samples during enrollment. Similarly, forensic recordings and historical audio materials may contain limited speech segments from a speaker of interest. A recognition system operating in these circumstances must extract sufficient speaker-discriminative information from the available samples while reducing its dependence on large labeled datasets.

Few-shot speaker recognition presents several difficulties that are less pronounced in conventional speaker recognition tasks. First, a limited number of reference utterances may not adequately represent the variability of a speaker's voice across different phonetic contexts and speaking conditions. Second, differences in recording environments or speech duration can produce substantial variation between the reference samples and the speech to be classified. Third, when the training samples are insufficient, a classification model with a large number of parameters may overfit the available data. Finally, the acoustic differences between certain speakers can be relatively small, making it difficult to construct well-separated speaker representations. These issues suggest that few-shot recognition requires not only an appropriate classification strategy but also input features that preserve meaningful speaker information when labeled training data are scarce.

One direction for addressing the dependence on labeled speech data involves learning general-purpose speech representations from large quantities of unlabeled recordings. Self-supervised learning (SSL) allows a neural network to acquire useful information from speech signals without requiring explicit speaker identity annotations for every training sample. Instead of relying entirely on manually labeled datasets, self-supervised models learn representations through training objectives derived from the structure of the input data. Such representations may contain information about phonetic characteristics, temporal relationships, speaker identity, and other acoustic properties. The resulting features can subsequently be transferred to downstream speech processing tasks using a relatively limited amount of task-specific labeled data.

Self-supervised speech representations (S3Rs) have already been investigated for conventional speaker recognition tasks [18, 19]. These representations offer an alternative to conventional acoustic features by using information learned during pretraining on unlabeled speech signals. Their potential applicability to speaker identification is particularly relevant when the amount of labeled training data is limited. Nevertheless, the suitability of a pretrained representation for a particular recognition task depends on the information retained during pretraining and the ability of the downstream classifier to distinguish among speaker identities.

Recent studies have also explored different neural network architectures and representation learning strategies for few-shot speaker identification. For example, a channel-attention convolutional recurrent neural network using three-dimensional log-Mel spectrogram features was developed to address feature extraction and overfitting problems in few-shot

speaker identification [20]. The method combines convolutional and recurrent processing with an attention mechanism to learn speaker-related information from limited speech samples. A subsequent study investigated a low-complexity speaker embedding module based on feature segmentation, transformation, and reconstruction [21]. This approach considered the computational requirements of few-shot speaker identification and demonstrated the importance of designing speaker representations that are suitable for resource-constrained applications. These investigations illustrate the continuing interest in improving few-shot speaker recognition through the design of specialized feature extraction and classification architectures.

Related research has examined different strategies for learning speaker embeddings without relying on extensive identity annotations. Tao et al. [22] investigated self-supervised training of speaker encoders using multimodal diverse positive pairs, demonstrating an alternative approach to constructing speaker representations without explicit speaker labels. Ge et al. [23] proposed an isomorphic graph attention network for pooling S3Rs, focusing on the aggregation of speaker-related information. Han et al. [24] investigated a cluster-aware self-distillation framework for self-supervised speaker verification, incorporating strategies to address unreliable pseudo-labels. These studies suggest that self-supervised representation learning offers a practical direction for reducing the dependence of speaker recognition systems on manually annotated speech data.

Other investigations have concentrated on the quality of pseudo-labels and the reliability of speaker embeddings learned without direct supervision. Fathan and Alam [25] analyzed the influence of clustering methods on self-supervised speaker verification and reported that the quality of generated pseudo-labels can substantially affect downstream recognition performance. Gao et al. [26] further investigated pseudo-label correction and clustering-based learning for speaker verification. Liu et al. [27] examined the selection of reliable pseudo-labels through confidence distribution modeling in semi-supervised and self-supervised speaker recognition. A recent study and review of SSL for speaker recognition [28] also discussed the characteristics and limitations of different representation learning frameworks. Collectively, these investigations indicate that the quality of learned speech representations and their adaptation to a particular recognition task remain important research considerations.

Among the available self-supervised speech models, Mockingjay and WavLM provide two possible sources of pretrained speech representations. Mockingjay adopts a masked acoustic modeling strategy to learn contextual information from speech sequences, while WavLM combines masked speech prediction with denoising-oriented pretraining to capture information relevant to different speech processing tasks [18, 19]. Although their training strategies differ, both models can generate contextual speech representations that may contain information useful for distinguishing speakers. Unlike conventional Mel-spectrogram features, these representations are learned from speech data through pretraining objectives rather than being determined solely by predefined signal transformations. However, the information contained in a pretrained representation is not necessarily optimized for few-shot speaker identification. Its suitability depends on whether the downstream recognition model can extract and use speaker-discriminative characteristics from a

small set of labeled utterances.

Yi et al. [29] further investigated the application of S3Rs to speaker segmentation using a WavLM-based approach. This investigation provides another example of the use of pretrained speech representations in speaker-related processing tasks, although speaker segmentation and speaker identification involve different recognition objectives.

In addition to the input representation, the architecture of the classifier plays an important role in speaker recognition. Speech signals contain temporal variations arising from changes in phonetic content, articulation, and speaking style. A classification model must therefore process sequences of acoustic features and identify information that is relevant to speaker identity. Transformer encoders provide a mechanism for modeling relationships among different positions in an input sequence through self-attention. Unlike architectures that process temporal information primarily through local operations or sequential recurrent updates, self-attention allows the model to establish relationships between feature representations at different temporal positions. This property makes Transformer encoders suitable candidates for processing contextual speech representations. Nevertheless, their actual performance in few-shot speaker identification must be established experimentally, particularly because a limited number of labeled samples may restrict the effective training of the classifier.

The combination of S3Rs and Transformer-based classification offers a potential approach to the limited-data problem. Pretrained representations can provide acoustic information learned from speech data before the downstream recognition stage, while a Transformer encoder can process relationships within the resulting feature sequences. In principle, these two components serve complementary purposes: the pretrained model provides the input representation, and the classifier learns to distinguish the speaker identities represented in the labeled training samples. The recognition performance of this combination, however, cannot be inferred directly from the success of self-supervised models in conventional speaker recognition. Few-shot identification involves different constraints on the number of labeled samples and may require the classifier to operate with substantially less speaker-specific information.

Although self-supervised representations have been studied in speaker recognition, the comparative behavior of different pretrained speech representations when coupled with a Transformer encoder for few-shot speaker identification still warrants further investigation. In particular, it is useful to determine whether such representations provide more informative input features than conventional Mel-spectrograms when the number of labeled speaker samples is limited. It is also necessary to examine whether the choice of pretrained representation affects the recognition performance of the same classification architecture. Addressing these questions can provide a clearer understanding of the relationship between representation learning and Transformer-based classification in few-shot speaker identification.

Motivated by these considerations, this study proposes a few-shot speaker recognition framework that combines S3Rs with a Transformer encoder. Two pretrained representations, Mockingjay and WavLM, are considered separately as input features, and a Transformer encoder is employed as the principal component of the speaker classification model. The study evaluates the recognition performance obtained using these representations and compares it with the performance

achieved using conventional Mel-spectrogram features. The objective is to investigate the applicability of pretrained speech representations to few-shot speaker identification and to examine their interaction with a Transformer-based classifier. Since the task addressed in this paper is speaker identification, the term speaker recognition in the following sections refers specifically to speaker identification unless otherwise stated.

The main contributions of this study are summarized as follows:

(1) A few-shot speaker identification framework is developed by combining pretrained S3Rs with a Transformer encoder. The framework investigates the use of speech features learned through self-supervised pretraining for classification under limited labeled-data conditions.

(2) Two S3Rs, Mockingjay and WavLM, are examined within the same Transformer-based classification framework. Their respective recognition results are analyzed to investigate the influence of the input representation on few-shot speaker identification performance.

(3) The proposed framework is evaluated against a conventional Mel-spectrogram-based approach. The experimental comparison provides evidence regarding the suitability of S3Rs for few-shot speaker recognition and identifies differences in recognition performance between the selected representations.

The remainder of this paper is organized as follows. Section 2 reviews related studies on speaker recognition, few-shot learning, and self-supervised speech representation learning. Section 3 describes the proposed framework, including the extraction of S3Rs and the Transformer-based classification model. Section 4 presents the experimental setup, recognition results, and corresponding analysis. Finally, Section 5 summarizes the findings and discusses possible directions for future research.

2. RELATED WORK

The performance of a few-shot speaker recognition system is closely related to two aspects: the quality of the speech representations used to characterize individual speakers and the ability of the classification model to distinguish speaker-related information from variations caused by speech content and recording conditions. Traditional acoustic features provide useful spectral information, but their effectiveness may be limited when only a small number of labeled speech samples are available. Recent developments in SSL have introduced alternative approaches to extracting speech representations from large amounts of unlabeled data. At the same time, Transformer-based architectures have provided new methods for modeling relationships within sequential acoustic features.

This section reviews the two principal components relevant to the proposed framework. Section 2.1 discusses self-supervised speech representation learning, with particular attention to Mockingjay and WavLM. Section 2.2 introduces the Transformer encoder and explains its main architectural components and their relevance to speaker recognition.

2.1 Self-supervised speech representation

SSL has attracted considerable attention in speech processing because it allows models to learn useful

representations from speech data without requiring manually assigned labels for every training sample. In conventional supervised learning, the training process generally relies on labeled speech recordings associated with specific phonetic categories, speaker identities, or other target information. Obtaining such annotations can be expensive, particularly when large datasets are required. In contrast, SSL constructs learning objectives from the input speech itself, allowing neural networks to discover structural and contextual information through pretraining on unlabeled recordings.

A variety of self-supervised speech representation models have been developed using different learning objectives and network architectures. Representative examples include Mockingjay [18], WavLM [19], wav2vec 2.0 [30], autoregressive predictive coding (APC) [31], and contrastive predictive coding (CPC) [32]. These models differ in how they process speech signals, construct training targets, and learn relationships between acoustic observations. Nevertheless, they share the general objective of obtaining representations that can subsequently be used in downstream speech processing tasks. Such representations have been applied to automatic speech recognition, speaker recognition, speech classification, and other applications involving acoustic signals.

The main distinction between conventional acoustic feature extraction and self-supervised representation learning lies in how the features are obtained. Traditional features, such as Mel-spectrograms and Mel-frequency cepstral coefficients, are computed using predefined signal processing procedures. They describe the spectral properties of speech but are not directly optimized to represent contextual relationships within an utterance. Self-supervised models, on the other hand, learn feature transformations during pretraining. The resulting representations may encode relationships among neighboring or distant speech segments, depending on the architecture and training objective. However, the presence of contextual information does not necessarily mean that every representation is equally suitable for identifying speakers. Different pretraining methods may preserve different amounts of speaker-specific and linguistic information.

One important category of SSL methods is based on predictive learning. In autoregressive predictive coding, a model is trained to predict future acoustic observations from preceding speech segments [31]. By requiring the model to estimate information that has not yet been observed, predictive learning encourages the extraction of patterns that persist across consecutive frames. CPC adopts a different strategy, in which the model learns to distinguish representations corresponding to actual future observations from negative samples [32]. The contrastive objective encourages the model to capture information useful for predicting the temporal structure of the signal. Although these approaches differ in their optimization procedures, both demonstrate how speech representations can be learned without direct supervision from speaker identity labels.

Another category of self-supervised approaches relies on masked prediction. Instead of predicting only future observations, masked prediction methods conceal or modify portions of the input sequence and train the model to recover information associated with the affected regions. This strategy encourages the representation model to make use of contextual information from the available speech segments. It also provides a way to learn relationships among acoustic observations without requiring explicitly annotated speaker

labels.

Mockingjay [18] is a representative model based on masked acoustic modeling. It uses Mel-spectrogram features as input and learns contextualized speech representations through a bidirectional Transformer-based encoder. Unlike unidirectional predictive models that rely primarily on preceding observations, Mockingjay can use information from both earlier and later frames to predict the acoustic content of masked positions. This bidirectional processing allows the model to learn relationships across the input sequence and to construct representations that reflect the surrounding acoustic context.

The encoding network of Mockingjay consists of multiple Transformer encoder layers. Each layer contains a multi-head self-attention mechanism and a position-wise feed-forward network, together with residual connections and normalization operations. The self-attention mechanism allows the model to establish relationships among frames at different positions in the speech sequence. The feed-forward network subsequently transforms the contextualized representations, while residual connections and normalization support information propagation through the encoder layers. The output of the final encoder layer provides a contextual representation for each input time step.

During Mockingjay pretraining, a proportion of the input frames is selected for modification. In the original masking procedure, approximately 15% of the frames are selected, and different masking operations are applied according to predefined probabilities. These operations include replacing selected frames with zero-valued inputs, substituting them with randomly selected frames, or leaving them unchanged. The network is then trained to reconstruct the original acoustic information at the selected positions using the available context. Because the model does not receive an unmodified version of every selected frame, it must learn relationships within the speech sequence rather than simply reproducing the input.

After pretraining, the masking procedure is not required when Mockingjay is used as a speech representation extractor. An input utterance can be processed by the pretrained encoder to obtain a sequence of contextual feature vectors. These vectors can then serve as input to a downstream model designed for a particular recognition task. For speaker identification, the downstream classifier must determine which characteristics of the extracted representations are useful for distinguishing individual speakers. Although Mockingjay was not originally designed exclusively for speaker recognition, its contextual representations provide a possible alternative to conventional acoustic features.

An important consideration is that the information required for speaker recognition differs from that required for automatic speech recognition. Speech recognition focuses mainly on linguistic content, while speaker recognition requires characteristics associated with the identity of the speaker. A representation that performs well in one task may not necessarily provide the most discriminative information for another. Mockingjay representations therefore need to be evaluated within the specific recognition framework rather than assumed to be superior to conventional acoustic features. This consideration is particularly important in few-shot speaker recognition, where only a small amount of labeled information is available for learning speaker-specific decision boundaries.

WavLM [19] is another self-supervised speech

representation model developed to support a broad range of speech processing tasks. Unlike Mockingjay, which operates on Mel-spectrogram features, WavLM processes speech waveforms using a convolutional feature encoder followed by Transformer-based contextual modeling. The convolutional encoder transforms the input waveform into latent speech representations, while the Transformer layers capture relationships among the resulting acoustic features. This architecture allows WavLM to learn speech representations without depending on a predefined Mel-spectrogram input representation.

The pretraining strategy of WavLM combines masked speech prediction with denoising-oriented learning. During training, the input speech may be modified through masking, noise addition, or the introduction of overlapping speech. The model is trained to predict discrete targets corresponding to the original speech at masked positions. This procedure encourages the network to learn information from speech signals under different acoustic conditions. By incorporating additional training conditions beyond clean speech reconstruction, WavLM aims to obtain representations that can support both linguistic and non-linguistic speech processing tasks.

WavLM is relevant to speaker recognition because speaker identity information must be extracted from signals that may also contain phonetic variation, background interference, and changes in speaking conditions. A model pretrained on varied speech signals may learn representations containing information relevant to these factors. However, pretrained features can retain both speaker-related and speaker-independent characteristics. Consequently, an additional classification model is required to learn how the available representations should be used for a particular set of speakers.

WavLM has been investigated in several downstream applications, including automatic speech recognition, speaker verification, and other speech analysis tasks. Its performance across different tasks suggests that the pretrained representations contain information that can be adapted to multiple recognition objectives. Nevertheless, the effectiveness of these representations in few-shot speaker identification depends on the available enrollment samples, the classification architecture, and the experimental conditions. A strong result in conventional speaker verification does not automatically imply an equivalent result in speaker identification with limited labeled samples.

Mockingjay and WavLM are selected in this study because they represent two different approaches to self-supervised speech representation learning. Mockingjay learns contextual information from Mel-spectrogram inputs through masked acoustic reconstruction, whereas WavLM learns representations from speech waveforms using masked prediction and denoising-oriented pretraining. Their differences provide an opportunity to investigate how the choice of pretrained representation influences the performance of a common downstream classification model.

In the proposed framework, the two representations are considered separately rather than combined into a single feature vector. This arrangement makes it possible to examine the recognition performance associated with each representation under the adopted experimental setting. A comparison with conventional Mel-spectrogram features is also included to determine whether the use of pretrained speech representations provides an advantage when only a limited number of labeled utterances are available. The

effectiveness of these representations is assessed through the experimental results presented in Section 4.

2.2 Transformer encoder

The Transformer is a neural network architecture designed to process sequential information using attention mechanisms. Unlike recurrent neural networks, which commonly update hidden representations through a sequence of recurrent operations, the Transformer uses self-attention to model dependencies among elements of an input sequence. This allows relationships between different positions to be represented without relying exclusively on a recurrent processing structure. As a result, Transformer-based models have been widely investigated for sequential learning tasks, including natural language processing and speech signal analysis.

The Transformer architecture contains encoder and decoder components in its original sequence-to-sequence formulation. However, many classification and representation learning tasks can be addressed using the encoder alone. The Transformer encoder accepts a sequence of feature vectors and produces a corresponding sequence of contextualized representations. Each output vector contains information derived from the associated input position and, through self-attention, information from other positions in the sequence. This characteristic makes the encoder suitable for processing speech representations in which information is distributed across multiple time steps.

Figure 1 illustrates the basic architecture of the Transformer encoder considered in this study. The architecture includes input embedding processing, positional information addition, multi-head self-attention, residual connection and normalization operations, and a position-wise feed-forward network. These components work together to transform the input feature sequence into representations that incorporate contextual relationships. The following paragraphs describe their respective functions.

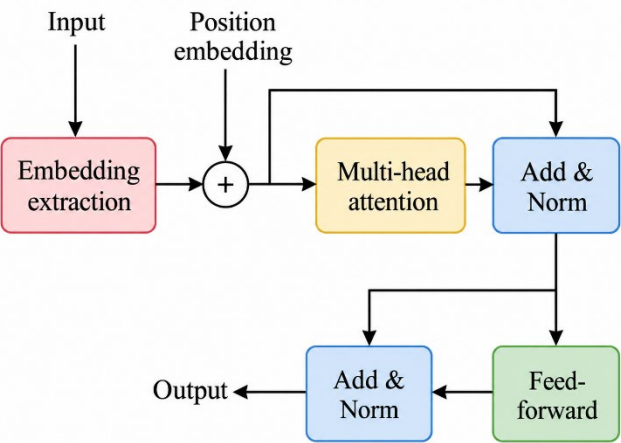


Figure 1. The architecture of Transformer encoder

The input embedding stage prepares the features for processing by the Transformer encoder. In general, each input element is represented as a vector with a predefined feature dimension. When the original input feature dimension differs from the hidden dimension required by the encoder, a projection operation can be employed to obtain a compatible

representation. In speech processing, the input may consist of acoustic feature vectors or representations obtained from a pretrained speech model. The embedding stage therefore provides the feature sequence on which subsequent attention operations are performed.

Since the self-attention mechanism does not inherently distinguish the order of input elements, positional information is introduced to describe their relative or absolute locations within the sequence. In a conventional Transformer encoder, positional embeddings or positional encodings can be added to the input representations. The resulting vectors contain both feature information and an indication of sequence position. This is important for speech processing because acoustic characteristics are not independent of their temporal arrangement. The order of speech frames contributes to the formation of phonetic patterns and other time-dependent information within an utterance.

The multi-head self-attention module is the principal component responsible for modeling relationships across the input sequence. In self-attention, each input representation is transformed into query, key, and value vectors. Attention weights are determined from the relationships between queries and keys, and these weights are used to aggregate information from the corresponding value vectors. Consequently, the representation at a particular time step can incorporate information from other positions in the speech sequence. The attention mechanism does not simply assign different weights to feature dimensions; instead, it models relationships among sequence positions according to the learned query and key representations.

Multi-head attention extends this mechanism by performing several attention operations in parallel using different learned projections. Each attention head can capture relationships under its own representation subspace. The outputs of the individual heads are then concatenated and transformed to form the output of the multi-head attention module. This design allows the encoder to examine different relationships within the same input sequence. In speech processing, such relationships may involve nearby acoustic frames or more distant segments of an utterance. The specific patterns learned by the attention heads depend on the training data and optimization objective.

After the multi-head self-attention operation, a residual connection and normalization stage is applied. A residual connection combines the input of a sublayer with its transformed output, allowing information from earlier representations to pass through the network. This operation can reduce the difficulty of training deep networks by providing a direct path for information and gradient propagation. Layer normalization is then used to regulate the distribution of the intermediate representations. Together, these operations contribute to the stability of the encoder and help preserve relevant information during successive feature transformations.

The feed-forward network is another important component of the Transformer encoder. It is applied independently to the representation at each sequence position, using the same network parameters across positions. In a conventional Transformer encoder, the feed-forward network contains two linear transformations separated by a nonlinear activation function. The intermediate hidden dimension may differ from the encoder dimension, allowing the network to transform the information obtained from the attention module. Unlike self-attention, which exchanges information across positions, the

position-wise feed-forward network processes the feature representation at each position separately.

A second residual connection and normalization stage follows the feed-forward network. This stage combines the input to the feed-forward sublayer with its transformed output and performs normalization according to the adopted encoder configuration. In the conventional post-normalization architecture, normalization follows the residual addition. Other Transformer implementations may place normalization before the attention or feed-forward operation. Although these arrangements differ in computational order, both are intended to support stable learning and information propagation through the encoder.

The encoder can contain one or more stacked Transformer layers. When multiple layers are used, the output of one layer becomes the input to the next. Successive layers allow the model to construct increasingly transformed contextual representations from the original input sequence. The number of layers, the number of attention heads, and the hidden feature dimension determine important aspects of the encoder architecture. These parameters influence the representational capacity and computational requirements of the model. Their selection is particularly relevant when the available labeled training dataset is small.

For a standard Transformer encoder without temporal pooling or sequence-reduction operations, the number of sequence positions remains unchanged between the input and output. The hidden feature dimension is also maintained across the attention and feed-forward sublayers when the appropriate output projections are used. Therefore, an input sequence of a given length and encoder dimension produces an output sequence with the same length and hidden dimension. However, this property applies to the encoder layers themselves and does not imply that the dimensionality of the original acoustic features must be identical to the encoder hidden dimension. Input projection, pooling, or subsequent classification layers may change feature dimensions outside the encoder.

In speaker recognition, a Transformer encoder can be used to process sequential speech features and learn relationships among representations extracted from different portions of an utterance. Speech signals contain variations caused by phonetic content, articulation, and changes in acoustic conditions. Some of these variations may be unrelated to speaker identity, while others may carry information useful for distinguishing speakers. By processing the feature sequence through self-attention and feed-forward transformations, the encoder provides a mechanism for learning contextual relationships that may contribute to speaker classification.

For few-shot speaker recognition, the use of a Transformer encoder presents both opportunities and challenges. Its attention mechanism can process information distributed across a speech sequence, which is relevant when individual utterances contain limited speaker-specific evidence. However, the encoder must learn classification-related parameters from the available training samples, and insufficient labeled data may increase the risk of overfitting. The use of pretrained speech representations as input provides an alternative to relying entirely on features learned from a small speaker-labeled dataset. The representations are obtained from a model trained beforehand, while the Transformer-based classifier is used to learn distinctions among the target speakers.

The present study adopts a Transformer encoder as the core component of the classification framework. S3Rs extracted

using Mockingjay or WavLM are supplied to the downstream model, which processes the input features to support speaker identification. The role of the Transformer encoder is to learn relationships within the supplied representations and provide information for the subsequent classification stage. This arrangement allows the study to examine whether contextual feature processing contributes to few-shot speaker recognition when combined with different pretrained speech representations.

The combination of SSL representations and a Transformer encoder does not guarantee improved performance for every recognition task. Its effectiveness depends on the information preserved by the pretrained features, the design of the classification model, and the characteristics of the available speech samples. Accordingly, the framework is evaluated experimentally rather than assuming that pretrained representations or attention-based processing will necessarily outperform conventional approaches. The detailed model configuration and recognition procedure are presented in Section 3, while Section 4 reports the experimental results and corresponding comparisons.

3. THE PROPOSED METHOD

This section describes the proposed few-shot speaker recognition framework, which combines S3Rs with a Transformer encoder-based (TE-based) classifier. The framework is designed to investigate whether pretrained speech representations can provide effective input features for speaker identification when only a limited number of labeled speech samples are available. Rather than relying exclusively on conventional acoustic features and a classifier trained from labeled speech data, the proposed method uses representations obtained from pretrained self-supervised speech models. These representations are subsequently processed by a Transformer-based classifier to distinguish among the target speakers.

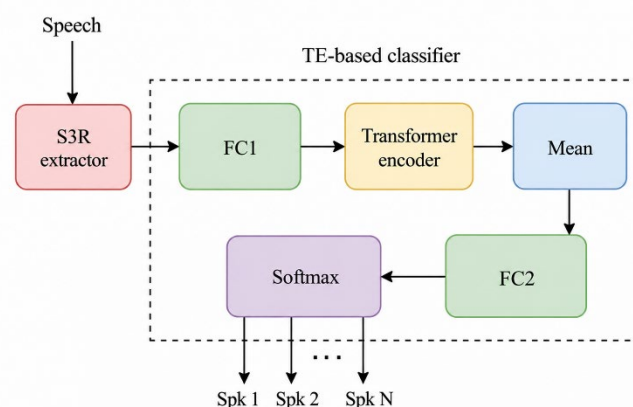


Figure 2. Framework of the proposed speaker recognition method based on self-supervised speech representations and a Transformer encoder

As illustrated in Figure 2, the proposed framework consists of two principal parts: an S3R extractor and a TE-based classifier. The S3R extractor converts an input speech utterance into a sequence of feature vectors, while the TE-based classifier processes these vectors and produces classification probabilities corresponding to the enrolled

speakers. The two parts are connected sequentially, with the output of the representation extractor serving as the input to the classification network.

The main functions of the two parts are described below.

(1) S3R extractor: This part extracts contextual speech representations from the input utterance using a pretrained self-supervised speech model. Two alternative extractors are considered in this study, namely Mockingjay and WavLM. Each extractor produces a sequence of acoustic representations that can be supplied to the subsequent classification network.

(2) TE-based classifier: This part receives the extracted speech representations and learns to distinguish the identities of the speakers included in the classification task. It consists of a feature projection layer, a Transformer encoder, a temporal averaging operation, a second fully connected layer, and a softmax operation. These components perform feature transformation, contextual feature learning, sequence-level aggregation, and speaker classification.

The proposed method separates the extraction of pretrained speech representations from the learning of speaker-specific classification information. This arrangement makes it possible to examine different S3R models within a common downstream classification framework. In particular, Mockingjay and WavLM representations are evaluated separately, allowing their recognition performance to be compared using the same general classification architecture.

The overall processing procedure begins with an input speech utterance. The S3R extractor first converts the utterance into a sequence of feature vectors. These vectors are then projected into the feature space required by the TE-based classifier through the first fully connected layer. The Transformer encoder processes the projected sequence and produces contextualized representations. Subsequently, temporal mean pooling aggregates the sequence-level information into a fixed-length utterance-level feature vector. This vector is passed through the second fully connected layer to produce speaker-specific classification scores, which are converted into probabilities by the softmax operation.

The two principal parts of the framework perform different but connected functions. The S3R extractor provides pretrained speech representations, while the TE-based classifier learns to associate the extracted features with the target speaker identities. The detailed implementation and processing procedures of these two parts are described in Sections 3.1 and 3.2, respectively.

3.1 S3R extractor part

The S3R extractor is responsible for converting an input speech utterance into a sequence of learned acoustic representations. In contrast to conventional feature extraction methods, which obtain acoustic characteristics through predefined signal processing operations, self-supervised representation learning uses a pretrained neural network to transform speech signals into contextual feature vectors. These representations are learned from speech data through self-supervised pretraining objectives and can subsequently be used as inputs to different downstream recognition models.

The motivation for using an S3R extractor in the proposed framework is closely related to the limited availability of labeled speech samples in few-shot speaker recognition. When only a small number of utterances are available for each speaker, training a feature extraction network entirely from the available labeled data may be difficult. A pretrained self-

supervised model offers an alternative source of acoustic representations learned before the speaker classification stage. The downstream classifier can then operate on these representations rather than relying exclusively on feature patterns learned from a limited speaker-labeled dataset.

In this study, the S3R extractor can be implemented using either the Mockingjay model or the WavLM model. The two models are considered separately because they differ in their input processing procedures and pretraining strategies. The choice of extractor determines the form of the speech representation supplied to the TE-based classifier, while the subsequent classification procedure follows the same general framework.

For the Mockingjay-based implementation, the input speech is first represented using Mel-spectrogram features. These features are processed by the pretrained Mockingjay encoder to obtain contextualized speech representations. The encoder uses information from different positions within the input sequence to construct a feature vector for each time step. Unlike the masked input sequences employed during self-supervised pretraining, speech representations for downstream recognition are extracted without applying the pretraining masking procedure.

The resulting Mockingjay representations contain information derived from the spectral characteristics of the input utterance and the contextual relationships learned by the pretrained encoder. Although the original Mel-spectrogram describes acoustic information at individual time steps, the extracted representation also incorporates information from the surrounding speech sequence. This contextual processing provides an alternative representation of the utterance for the downstream speaker identification task.

For the WavLM-based implementation, the speech waveform is processed by the pretrained WavLM model. WavLM uses a convolutional feature encoder to generate latent acoustic representations, followed by Transformer-based contextual processing. The representations obtained from the pretrained model are then used as input to the TE-based classifier. Since WavLM is pretrained using masked prediction and denoising-oriented learning objectives, its extracted features may contain information related to different acoustic characteristics and temporal relationships within the speech signal.

The use of WavLM differs from that of Mockingjay primarily in the representation extraction stage. Mockingjay operates on Mel-spectrogram inputs, whereas WavLM processes speech waveforms through its own feature encoding architecture. Nevertheless, the output of either extractor can be regarded as a sequence of feature vectors associated with the input utterance. This common representation format allows both models to be incorporated into the proposed speaker classification framework.

For an input utterance, the S3R extractor produces a feature sequence containing representations at successive temporal positions. The number of extracted feature vectors depends on the input utterance and the temporal resolution of the selected representation model. Similarly, the dimensionality of each vector is determined by the output configuration of the pretrained extractor. Therefore, Mockingjay and WavLM do not necessarily produce feature sequences with identical temporal lengths or feature dimensions.

The TE-based classifier is designed to process these representations through an initial feature transformation stage. Consequently, the representations extracted from different

self-supervised models can be supplied to the same general downstream architecture, provided that their input dimensions are appropriately accommodated by the first fully connected layer.

An important distinction should be made between representation extraction and speaker classification. The S3R extractor provides acoustic feature vectors, but these vectors do not directly represent the final predicted speaker identities. Speaker recognition requires an additional decision process that maps the extracted features to the speaker classes defined in the recognition task. This function is performed by the TE-based classifier described in Section 3.2.

The two S3R extractors are evaluated independently in the proposed framework. This arrangement allows the study to investigate how representations obtained from different self-supervised pretraining procedures affect few-shot speaker identification. The extracted features are passed to the subsequent classification stages, where they undergo dimensionality transformation, contextual processing, temporal aggregation, and speaker probability estimation.

3.2 TE-based classifier part

The TE-based classifier is the second major component of the proposed few-shot speaker recognition framework. Its primary function is to transform the representations generated by the S3R extractor into classification results corresponding to the target speakers. Since the extractor produces a sequence of feature vectors rather than a single speaker identity label, the classifier must process the temporal information contained in the sequence and obtain a fixed-length representation suitable for classification.

As illustrated in Figure 2, the TE-based classifier consists of five modules: the first fully connected layer (FC1), the Transformer encoder, the mean operation, the second fully connected layer (FC2), and the softmax operation. The modules are arranged sequentially, and each performs a different function in the classification process.

The FC1 module performs the initial feature transformation. It receives the sequence of representations generated by the S3R extractor and converts their feature dimensions to the dimension used by the Transformer encoder. In the proposed architecture, FC1 reduces the dimensionality of the extracted S3R features before they are passed to the subsequent module.

This transformation serves two purposes. First, it allows feature vectors generated by different S3R extractors to be processed using a common encoder input dimension. Second, reducing the feature dimension can decrease the amount of computation required by the subsequent Transformer encoder. Since the input representations may contain a relatively large number of feature dimensions, the projection operation provides a more compact representation for contextual feature processing.

FC1 is applied to the feature vectors associated with the temporal positions of the input utterance. Therefore, the operation transforms the feature dimension while retaining the sequential organization of the extracted representations. The output of FC1 remains a sequence of feature vectors, with its temporal structure preserved for processing by the Transformer encoder.

The Transformer encoder is the principal feature learning module of the TE-based classifier. It receives the projected S3R feature sequence and processes the relationships among representations at different temporal positions. Through the

self-attention mechanism, each position can incorporate information from other positions within the input sequence. The resulting representations therefore contain contextual information derived from the sequence rather than information associated solely with individual feature vectors.

The attention mechanism is particularly relevant to speech processing because the acoustic characteristics of an utterance vary over time. Different segments may contain different phonetic content, and the extent to which a segment provides information useful for distinguishing speakers may also vary. By modeling relationships among temporal positions, the Transformer encoder provides a mechanism for learning contextual patterns from the extracted speech representations.

The multi-head self-attention mechanism allows several sets of relationships to be modeled through different learned projections. The attention outputs are processed together and transformed through the feed-forward sublayers of the encoder. Residual connections and normalization operations support information propagation through the network. As a result, the output of the Transformer encoder consists of a sequence of feature vectors that has undergone contextual feature transformation.

In the proposed framework, the input and output of the Transformer encoder maintain the same sequence length and hidden feature dimension. This property applies to the encoder itself, after the input representations have been transformed by FC1. Although the encoder changes the numerical values and contextual information contained in the feature vectors, it does not change their overall sequence shape.

It should be noted that the Transformer encoder does not directly produce the final speaker classification results. Its role is to transform the projected speech representations into contextualized features that can subsequently be aggregated and classified. The sequence-level output must therefore be converted into a fixed-length representation before it can be supplied to the final classification layer.

The mean module performs this conversion through temporal averaging. It computes the mean value of the Transformer output feature vectors along the temporal dimension. For each feature dimension, the values corresponding to the different time steps are averaged to obtain a single value. The resulting representation contains one aggregated value for each hidden feature dimension.

This operation converts a sequence of contextualized frame-level representations into an utterance-level feature vector. Unlike the input to the mean module, which contains multiple temporal positions, the output is a single fixed-length vector representing the processed utterance. Consequently, the classifier can obtain an utterance-level representation even when the input sequences contain different numbers of temporal positions.

Temporal averaging also provides a relatively simple aggregation strategy. It does not require an additional trainable attention-pooling network or a separate sequence modeling module. Instead, information from the entire Transformer output sequence contributes to the final aggregated representation. However, the mean operation assigns equal averaging weights to the temporal positions, and its effectiveness depends on the quality of the features produced by the preceding Transformer encoder.

The distinction between temporal feature processing and utterance-level aggregation is important in the proposed architecture. The Transformer encoder operates on the sequence of feature vectors and learns contextual relationships

across positions, whereas the mean module reduces the temporal dimension and generates a fixed-length representation. These operations serve different purposes and together establish the connection between sequential speech feature learning and speaker classification.

The aggregated feature vector is subsequently passed to FC2. This module maps the utterance-level representation to a vector whose number of output nodes equals the number of speaker classes in the recognition task. Each output node corresponds to one enrolled speaker, and its value represents the classification score assigned to that speaker before probability normalization.

The output dimension of FC2 is therefore determined by the number of target speakers. For a task involving a fixed set of speaker identities, the corresponding classification layer contains one output node for each identity. When the number of speaker classes changes, the required output dimension of FC2 must also be adjusted accordingly. This configuration is consistent with the speaker identification objective adopted in the present study, where the input utterance is assigned to one of the predefined speaker classes.

Although FC1 and FC2 are both fully connected layers, they operate on representations with different structural characteristics. FC1 processes the feature vectors at individual temporal positions and transforms their feature dimensions before contextual learning. FC2, in contrast, processes the fixed-length utterance-level representation obtained after temporal averaging and maps it to the speaker classification scores.

The softmax module is applied after FC2 to convert the classification scores into a probability distribution over the target speaker classes. Each output probability is nonnegative, and the probabilities associated with all speaker classes sum to one. The resulting values indicate the relative classification probabilities assigned by the model to the different speakers.

During inference, the predicted speaker identity is determined by the output node with the highest probability. Thus, an input utterance is assigned to the speaker class associated with the maximum softmax output. The classifier operates within the predefined speaker set and produces a speaker identification result based on the information learned during training.

The TE-based classifier is trained using cross-entropy loss. During the training stage, an input utterance is processed by the S3R extractor and the subsequent classification modules to obtain the predicted speaker probabilities. These probabilities are compared with the corresponding ground-truth speaker label through the cross-entropy objective. The loss measures the discrepancy between the predicted probability distribution and the target speaker identity.

The training process adjusts the parameters of the TE-based classifier to reduce the classification loss. Through this process, the classifier learns how to transform the extracted S3R features into representations that distinguish among the target speakers. The FC1 layer learns the initial feature projection, the Transformer encoder learns contextual transformations, and FC2 learns the mapping between the aggregated utterance-level representation and the speaker classes.

Once training is completed, the resulting classifier can be used to identify speakers from input utterances according to the learned classification parameters. Each test utterance is processed through the same sequence of modules used during training. The S3R extractor first generates the pretrained speech representations, FC1 transforms their feature

dimension, the Transformer encoder processes the contextual relationships, and the mean operation produces a fixed-length utterance-level representation. Finally, FC2 generates the speaker classification scores, and softmax converts these scores into probabilities.

The proposed framework is intended for few-shot speaker identification, in which the classifier is trained using a limited number of labeled speech samples for each speaker. The pretrained S3R extractor provides acoustic representations learned before the downstream classification task, while the TE-based classifier learns to distinguish the available speaker identities using those representations. This division between representation extraction and classification is particularly relevant when the labeled dataset is relatively small.

Nevertheless, the effectiveness of the framework depends on the information contained in the extracted representations and the ability of the classifier to learn appropriate decision boundaries from the available samples. The use of a pretrained representation does not eliminate the difficulties associated with limited labeled data, and the Transformer encoder may still be affected by the size and distribution of the training dataset. Therefore, the proposed method is evaluated experimentally using different input representations rather than assuming that the combination will necessarily produce superior recognition results.

In summary, the TE-based classifier transforms sequential S3Rs into utterance-level speaker classification probabilities through five successive processing modules. FC1 performs feature projection, the Transformer encoder learns contextual relationships, the mean operation aggregates temporal information, FC2 maps the aggregated representation to the speaker classes, and softmax produces the final probability distribution. Together with the S3R extractor, these modules form the complete few-shot speaker recognition framework illustrated in Figure 2. The recognition performance of the proposed method is examined in Section 4.

4. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the experimental evaluation of the proposed speaker recognition framework using S3Rs and a TE-based classifier. The experiments were conducted on the VCTK speech corpus to examine the recognition performance of different pretrained speech representations and classification architectures. Mockingjay and WavLM were evaluated as alternative input representations, and their results were compared with those obtained using conventional Mel-spectrogram features. In addition, a fully connected layer-based classifier was introduced as a baseline to investigate the influence of the Transformer encoder on recognition performance.

The following subsections describe the dataset, experimental configuration, recognition results, and comparative analysis.

4.1 Datasets introduction

The proposed method was evaluated using the Centre for Speech Technology Research Voice Cloning Toolkit (CSTR VCTK) corpus, a speech dataset containing recordings from English speakers with different accents. The experimental dataset included 109 speakers, each contributing approximately 400 spoken sentences. The speech recordings

contain variations in phonetic content and speaker characteristics, making the corpus suitable for investigating speaker identification using different acoustic feature representations.

In this study, the task was formulated as closed-set speaker identification. The objective was to classify each evaluation utterance into one of the 109 speaker identities included in the experimental dataset. Each speaker was represented in the training, development, and evaluation subsets, allowing the classification model to learn speaker-specific characteristics from the training recordings and identify speakers from separate evaluation utterances.

The dataset was divided into three subsets: training, development, and evaluation. The approximate ratio among the three subsets was 4:1:1. This partitioning strategy was applied at the utterance level, with recordings from each speaker distributed among the three subsets. The training set contained 28,695 utterances, while the development and evaluation sets each contained 7,306 utterances. The total number of utterances used in the experiments was 43,307. The distribution of utterances across the three subsets is summarized in Table 1.

Table 1. Distribution of utterances in the VCTK experimental dataset

Subset	Utterance Number
Training set	28,695
Development set	7,306
Evaluation set	7,306

The training set was used to optimize the parameters of the speaker classification model. During training, the development set was used to evaluate the intermediate models and select the model with the best development-set performance. The selected model was subsequently evaluated on the independent evaluation subset to obtain the reported recognition results.

This division separates model training and model selection from the final performance evaluation. Since the same speaker identities were included in all three subsets, the experiments assessed the ability of the classifier to recognize previously registered speakers from utterances not used for model training. The evaluation therefore represents a closed-set speaker identification task rather than recognition of entirely unseen speaker identities.

The use of the three subsets also makes it possible to compare different representation and classification configurations under a common experimental dataset. In the following experiments, the performance of the proposed framework was examined using Mockingjay and WavLM representations, with additional comparisons involving Mel-spectrogram features and an alternative classifier architecture.

4.2 Experimental setup

The experiments were implemented using PyTorch and conducted on a computing platform equipped with an NVIDIA RTX A5000 GPU. The evaluation focused on speaker identification performance using different speech representations and classification architectures.

Recognition accuracy was used as the evaluation metric, defined as the proportion of evaluation utterances whose predicted speaker identities matched their corresponding ground-truth labels. For each input utterance, the classifier

generated scores for the 109 target speaker classes, and the speaker associated with the highest output probability was selected as the predicted identity.

Two pretrained self-supervised speech models, Mockingjay and WavLM, were used to obtain the input feature representations. The pretrained models were employed as feature extractors without fine-tuning their parameters during the downstream speaker classification experiments. Accordingly, the classification model was trained using speech representations generated by the pretrained extractors.

WavLM consists of a convolutional feature encoder followed by a Transformer-based contextual encoder. The convolutional feature encoder contains seven temporal convolutional layers with 512 channels. Their convolutional strides are configured as (5, 2, 2, 2, 2, 2, 2), while the corresponding kernel sizes are (10, 3, 3, 3, 3, 2, 2). The Transformer-based component processes the resulting latent acoustic features and incorporates contextual information across the sequence. The model also employs convolution-based relative positional information, with a kernel size of 128 and 16 groups. Further details of the WavLM architecture are provided in reference [19].

Mockingjay employs a Transformer encoder to learn contextualized representations from Mel-spectrogram features. The model configuration considered in this study contains six Transformer encoder layers. Each layer has a hidden dimension of 768, a feed-forward network with 3,072 hidden units, and 12 attention heads. Through masked acoustic modeling during pretraining, Mockingjay learns relationships among speech frames and produces contextual feature representations. Additional architectural details can be found in reference [18].

In the representation extraction stage, Mockingjay and WavLM were applied separately to the input utterances. The extracted Mockingjay representations had a feature dimension of 768, while the WavLM representations had a feature dimension of 1,024. These representations served as alternative inputs to the TE-based classification network.

The TE-based classifier consisted of two fully connected layers, a Transformer encoder, a temporal mean pooling operation, and a softmax output layer. The first fully connected layer (FC1) projected the input features into a 512-dimensional space. This operation reduced the feature dimensions of Mockingjay and WavLM representations from 768 and 1,024, respectively, to a common dimension of 512.

The projected features were subsequently processed by the Transformer encoder to learn contextual relationships within the input sequence. Temporal mean pooling was then applied to aggregate the sequence representations into a fixed-length utterance-level feature vector. The second fully connected layer (FC2) mapped this vector to 109 output nodes, corresponding to the 109 speakers in the experimental dataset. Finally, the softmax operation converted the output scores into speaker classification probabilities.

The TE-based classifier was trained using the cross-entropy objective, with the ground-truth speaker identities serving as classification labels. The pretrained speech representation extractors were not fine-tuned during this process. Thus, the experimental framework examined the effectiveness of the extracted speech representations when used with a separately trained downstream classifier.

The same general TE-based classification architecture was adopted for the comparisons involving Mockingjay, WavLM, and Mel-spectrogram input features. The comparative

experiments were designed to examine the influence of the input feature representation and the classification architecture on recognition performance.

4.3 Experimental results

Table 2 presents the recognition results obtained on the VCTK evaluation set using Mockingjay and WavLM representations with the proposed TE-based classifier. The performance of the two representations was evaluated in terms of speaker identification accuracy.

Table 2. Experimental results on the evaluation set of VCTK using S3R and TE-based classifier in terms of accuracy

S3R	Classifier	Accuracy (%)
Mockingjay representation	TE-based	94.80
WavLM representation	TE-based	92.29

As shown in Table 2, the proposed framework achieved an accuracy of 94.80% using Mockingjay representations and 92.29% using WavLM representations. Both configurations achieved recognition accuracy exceeding 90% on the VCTK evaluation set, indicating that the extracted representations could be used to distinguish among the 109 speaker classes under the adopted experimental conditions. Mockingjay achieved a recognition accuracy 2.51 percentage points higher than that obtained using WavLM. Since both representations were processed using the same general TE-based classification architecture, the results indicate that the selection of input speech representation influenced the recognition performance of the downstream classifier.

The difference may be related to the distinct pretraining objectives and feature extraction procedures of the two models. Mockingjay learns contextual information from Mel-spectrogram features, whereas WavLM processes speech waveforms through convolutional feature extraction and Transformer-based contextual learning. These differences may affect the characteristics of the information retained in the resulting feature representations. However, the experimental results alone do not establish that Mockingjay contains more speaker identity information than WavLM.

The results demonstrate that both pretrained speech representations can be employed in the proposed speaker identification framework. Among the two representations considered, Mockingjay produced the higher recognition accuracy on the evaluation set. This finding also indicates that recognition performance depends not only on the classifier architecture but also on the characteristics of the speech representations supplied to it.

4.4 Comparison with Mel-spectrogram

To investigate the influence of self-supervised speech representation learning on recognition performance, the proposed framework was compared with a baseline using conventional Mel-spectrogram features. The comparison was conducted using the same general TE-based classification architecture, with the input feature representation varied among Mel-spectrogram, Mockingjay, and WavLM.

Table 3 presents the recognition accuracy obtained using the three feature representations on the VCTK evaluation set.

As shown in Table 3, the TE-based classifier achieved an accuracy of 89.09% when conventional Mel-spectrogram

features were used as input. In comparison, the accuracy increased to 94.80% with Mockingjay representations and 92.29% with WavLM representations. These results correspond to improvements of 5.71 and 3.20 percentage points, respectively, over the Mel-spectrogram baseline.

Table 3. Comparison of Mel-spectrogram and S3Rs on the VCTK evaluation set

Feature	Classifier	Accuracy (%)
Mel-spectrogram	TE-based	89.09
Mockingjay representation	TE-based	94.80
WavLM representation	TE-based	92.29

The results indicate that both S3Rs produced higher speaker identification accuracy than the conventional Mel-spectrogram features under the adopted experimental conditions. The improvement was more pronounced for Mockingjay, while WavLM also showed a measurable increase over the baseline.

One possible explanation is that S3Rs incorporate contextual information acquired during pretraining on speech data. Unlike conventional Mel-spectrogram features, which primarily describe local spectral characteristics, pretrained representations contain transformed information derived from the learning objectives and architectures of the respective models. Such information may be useful for the downstream classifier when distinguishing among speakers.

Nevertheless, the observed differences should not be interpreted as direct proof that the pretrained representations contain a greater quantity of speaker identity information. The relative performance may also depend on differences in feature dimensionality, pretraining strategies, and the interactions between the extracted representations and the classifier. The present comparison establishes the relative recognition performance of the evaluated configurations rather than identifying the precise representation properties responsible for the differences.

Overall, the findings show that the use of Mockingjay and WavLM representations was associated with higher recognition accuracy than the Mel-spectrogram baseline. These results support the use of pretrained S3Rs for speaker identification under the experimental conditions considered in this study.

4.5 Comparison with FC-based classifier

To examine the influence of the Transformer encoder on classification performance, the proposed TE-based classifier was compared with an alternative fully connected layer-based (FC-based) classifier.

The FC-based classifier was constructed by replacing the Transformer encoder module in the proposed framework with a fully connected layer. The remaining general processing structure was retained, allowing the two architectures to be compared using the same input speech representations. The framework of the FC-based classifier is illustrated in Figure 3.

For the FC-based classifier, the first fully connected layer (FC1) and the intermediate fully connected layer were configured with 512 output nodes. The final fully connected layer (FC2) contained 109 output nodes, corresponding to the number of speakers in the experimental dataset.

Both Mockingjay and WavLM representations were evaluated using the FC-based and TE-based classification architectures. Table 4 presents the corresponding recognition

results on the VCTK evaluation set.

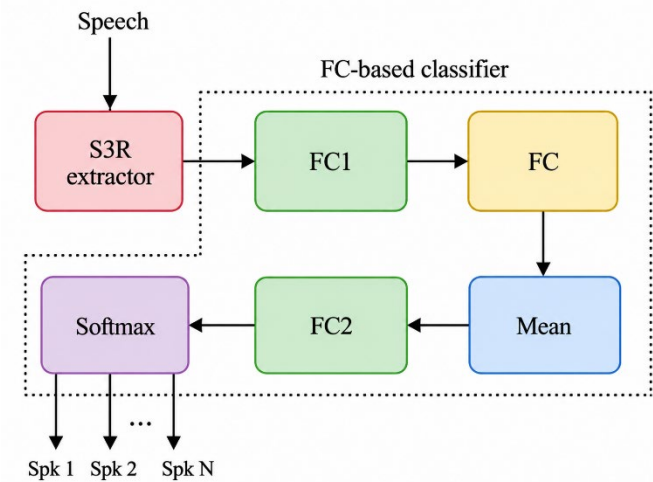


Figure 3. The framework of the FC-based classifier

Table 4. Comparison with FC-based classifier on evaluation set using S3R in terms of accuary

Feature	Classifier	Accuracy (%)
Mockingjay representation	FC-based	93.44
WavLM representation	FC-based	89.71
Mockingjay representation	TE-based	94.80
WavLM representation	TE-based	92.29

As shown in Table 4, the FC-based classifier achieved an accuracy of 93.44% with Mockingjay representations, whereas the TE-based classifier achieved 94.80%. The improvement associated with the TE-based classifier was therefore 1.36 percentage points.

When WavLM representations were used, the FC-based classifier achieved an accuracy of 89.71%, while the TE-based classifier reached 92.29%. The corresponding improvement was 2.58 percentage points, indicating a larger difference between the two classification architectures for WavLM than for Mockingjay.

These results show that the TE-based classifier outperformed the adopted FC-based baseline for both input representations. The improvement may be associated with the ability of the Transformer encoder to model relationships among features at different temporal positions through self-attention. Unlike a conventional position-wise fully connected transformation, the Transformer encoder can integrate information across the input sequence before temporal aggregation.

However, the observed performance differences cannot be attributed exclusively to the self-attention mechanism. The two classifiers differ in their internal architectures, and their relative performance may also depend on model capacity and parameter configurations. Therefore, the findings should be interpreted as a comparison between the specific classifier implementations considered in this study.

A comparison of the two speech representations across the classification architectures provides an additional observation. Mockingjay achieved higher recognition accuracy than WavLM with both classifiers. Under the FC-based architecture, Mockingjay exceeded WavLM by 3.73 percentage points. With the TE-based classifier, the difference decreased to 2.51 percentage points.

This result indicates that the relative ranking of the two input representations remained unchanged across the evaluated classifiers, although the size of the performance difference varied. The larger improvement obtained by WavLM when replacing the FC-based classifier with the TE-based classifier also suggests that the influence of the classification architecture may differ according to the characteristics of the input representations.

Taken together, the results indicate that the proposed TE-based classifier achieved higher speaker identification accuracy than the adopted FC-based baseline on the VCTK evaluation set. Combined with the comparisons in Sections 4.3 and 4.4, these findings demonstrate the importance of both input representation selection and classifier architecture in the evaluated speaker identification framework.

5. CONCLUSIONS

This study proposed a few-shot speaker recognition framework that combines S3Rs with a TE-based classifier. Two pretrained speech representations, Mockingjay and WavLM, were separately employed to extract acoustic features from input utterances. The resulting representations were processed by a classification network consisting of two fully connected layers, a Transformer encoder, a temporal mean pooling module, and a softmax layer. The first fully connected layer projected the extracted representations into a lower-dimensional feature space, while the Transformer encoder learned contextual relationships within the feature sequences. Temporal mean pooling subsequently aggregated the learned features into fixed-length utterance-level representations for speaker classification.

The experimental results demonstrated the feasibility of applying pretrained S3Rs to few-shot speaker identification. The proposed framework achieved recognition accuracy values of 94.80% and 92.29% when using Mockingjay and WavLM representations, respectively. Under the evaluated experimental conditions, Mockingjay produced higher recognition accuracy than WavLM. Furthermore, the comparison with conventional Mel-spectrogram features indicated that S3Rs provided more effective input features for the proposed classification framework. These findings suggest that pretrained speech representations can support speaker discrimination when only limited labeled training samples are available.

Nevertheless, the reported results are specific to the experimental settings considered in this study. Further investigation is required to assess the generalization of the framework across different speech datasets, recording conditions, and numbers of labeled samples per speaker. Future work may also examine the computational requirements of the classifier and the performance of additional pretrained speech representation models under few-shot recognition conditions.

ACKNOWLEDGMENT

This work is partly supported by Special Project for Key Areas in Guangdong Provincial Regular Higher Education Institutions (New Generation Information Technology) (Grant No.: 2023ZDZX1061), Teaching Reform Project of Association of Fundamental Computing Education in Chinese

Universities (Grant No.: 2024-AFCEC-296), 2025 University-level Teaching Quality Engineering Project (Education and Teaching Reform Research and Practice Project) of Guangdong Polytechnic of Science and Technology (Grant No.: JG202506), and Guangdong Polytechnic of Science and Technology 2023 University-level Demonstrative Curriculum for Curriculum-based Ideological and Political Education: RPA Application Technology (Grant No.: J32007002004).

REFERENCES

- [1] Vu, H.L., Dat, P.T., Nhi, P.T., Hao, N.S., Trang, N.T.T. (2025). VoxVietnam: A large-scale multi-genre dataset for Vietnamese speaker recognition. In 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, pp. 1-5. <https://doi.org/10.1109/ICASSP49660.2025.10890124>
- [2] Gan, C., Tu, Y., Jin, Z., Mak, M., Lee, K.A. (2025). Grouped knowledge distillation with adaptive logit softening for speaker recognition. In 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, pp. 1-5. <https://doi.org/10.1109/ICASSP49660.2025.10887814>
- [3] Huang, Y., Li, Y., Ren, Y., Tu, W., Yang, Y. (2025). FreqSense: Universal and low-latency adversarial example detection for speaker recognition with interpretability in frequency domain. In 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, pp. 1-5. <https://doi.org/10.1109/ICASSP49660.2025.10888789>
- [4] Li, Y., Chen, H., Cao, W., Huang, Q., He, Q. (2023). Few-shot speaker identification using lightweight prototypical network with feature grouping and interaction. *IEEE Transactions on Multimedia*, 25: 9241-9253. <https://doi.org/10.1109/TMM.2023.3253301>
- [5] Chen, Z., Wu, S., Li, X., Ai, Z., Xu, S. (2025). Open-set speaker identification through efficient few-shot tuning with speaker reciprocal points and unknown samples. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 33: 3347-3362. <https://doi.org/10.1109/TASLPRO.2025.3587591>
- [6] Jia, Y., Zhang, Y., Weiss, R., et al. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in Neural Information Processing Systems*, 31: 4480-4490.
- [7] Greenberg, C.S., Mason, L.P., Sadjadi, S.O., Reynolds, D.A. (2020). Two decades of speaker recognition evaluation at the National Institute of Standards and Technology. *Computer Speech & Language*, 60: 101032. <https://doi.org/10.1016/j.csl.2019.101032>
- [8] Yang, Z., Li, D., Cai, Y., Wang, Z., Yang, H. (2025). EX-Vector: Emotional X-Vector transfer learning for speaker recognition with emotion domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 33: 1415-1426. <https://doi.org/10.1109/TASLPRO.2025.3540666>
- [9] RamaSaiAvinash, K., Madhav, K.V., EmaldaRoslin, S., Nandhitha, N., Muthiah, A. (2025). Deep learning techniques for automated speaker recognition system. In 2025 2nd International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS), Bangalore, India, pp. 1-6. <https://doi.org/10.1109/ICCAMS65118.2025.11233917>
- [10] Yang, J., Chen, F., Cheng, Y., Lin, P. (2023). Integration of audio-visual information for multi-speaker multimedia speaker recognition. *Digital Signal Processing*, 145: 104315. <https://doi.org/10.1016/j.dsp.2023.104315>
- [11] Chinmai, H., Devaiah, D., Chiranth, V., Dwivedi, H.V., Shraddha, C. (2025). Speaker recognition in noisy environment using ECAPA-TDNN. In 2025 Annual International Conference on Data Science, Machine Learning and Blockchain Technology (AICDMB), Mysuru, India, pp. 1-4. <https://doi.org/10.1109/AICDMB64359.2025.11277609>
- [12] Katav, A., Moshe, Y., Cohen, I. (2025). A framework for robust speaker verification in highly noisy environments leveraging both noisy and enhanced audio. In 2025 33rd European Signal Processing Conference (EUSIPCO), Palermo, Italy, pp. 41-45. <https://doi.org/10.23919/EUSIPCO63237.2025.11226545>
- [13] Cho, S., Wee, K. (2025). Multi-noise representation learning for robust speaker recognition. *IEEE Signal Processing Letters*, 32: 681-685. <https://doi.org/10.1109/LSP.2025.3530879>
- [14] Wu, Z., De Leon, P.L., Demiroglu, C., et al. (2016). Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4): 768-783. <https://doi.org/10.1109/TASLP.2016.2526653>
- [15] Gomez-Alanis, A., Peinado, A.M., Gonzalez, J.A., Gomez, A.M. (2019). A gated recurrent convolutional neural network for robust spoofing detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12): 1985-1999. <https://doi.org/10.1109/TASLP.2019.2937413>
- [16] Wang, J., Wang, K., Law, M.T., Rudzicz, F., Brudno, M. (2019). Centroid-based deep metric learning for speaker recognition. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp. 3652-3656. <https://doi.org/10.1109/ICASSP.2019.8683393>
- [17] Anand, P., Singh, A.K., Srivastava, S., Lall, B. (2019). Few shot speaker recognition using deep neural networks. *arXiv preprint arXiv:1904.08775*. <https://doi.org/10.48550/arXiv.1904.08775>
- [18] Liu, A.T., Yang, S., Chi, P., Hsu, P., Lee, H. (2020). Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 6419-6423. <https://doi.org/10.1109/ICASSP40776.2020.9054458>
- [19] Chen, S., Wang, C., Chen, Z., et al. (2022). WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505-1518. <https://doi.org/10.1109/JSTSP.2022.3188113>
- [20] Saritha, B., Laskar, M.A., Monsley, A.K., Laskar, R.H., Choudhury, M. (2024). CACRN-Net: A 3D log Mel spectrogram based channel attention convolutional recurrent neural network for few-shot speaker identification. *Computers & Electrical Engineering*, 115: 109100. <https://doi.org/10.1016/j.compeleceng.2024.109100>

- [21] Li, Y., Huang, Q., Xing, X., Xu, X. (2025). Low-complexity speaker embedding module with feature segmentation, transformation and reconstruction for few-shot speaker identification. *Expert Systems with Applications*, 280: 127542. <https://doi.org/10.1016/j.eswa.2025.127542>
- [22] Tao, R., Lee, K.A., Das, R.K., Hautamäki, V., Li, H. (2023). Self-supervised training of speaker encoder with multi-modal diverse positive pairs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31: 1706-1719. <https://doi.org/10.1109/TASLP.2023.3268568>
- [23] Ge, Z., Xu, X., Guo, H., Wang, T., Yang, Z. (2024). Speaker recognition using isomorphic graph attention network based pooling on self-supervised representation. *Applied Acoustics*, 219: 109929. <https://doi.org/10.1016/j.apacoust.2024.109929>
- [24] Han, B., Chen, Z., Qian, Y. (2024). Self-supervised learning with cluster-aware-DINO for high-performance robust speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32: 529-541. <https://doi.org/10.1109/TASLP.2023.3331949>
- [25] Fathan, A., Alam, J. (2024). An analytic study on clustering driven self-supervised speaker verification. *Pattern Recognition Letters*, 179: 80-86. <https://doi.org/10.1016/j.patrec.2024.01.024>
- [26] Gao, C., Li, Y., Li, Y., Li, X. (2025). Improving pseudo labels quality with infomap and label correction for self-supervised speaker verification. *Pattern Recognition Letters*, 198: 86-92. <https://doi.org/10.1016/j.patrec.2025.10.005>
- [27] Liu, J., Wang, X., Meng, J., Li, B. (2026). SpeakerMatch: Matching reliable pseudo-labels in semi-supervised and self-supervised speaker recognition with confidence distribution. *Signal Processing*, 239: 110263. <https://doi.org/10.1016/j.sigpro.2025.110263>
- [28] Lepage, T., Dehak, R. (2026). Self-supervised learning for speaker recognition: A study and review. *Speech Communication*, 176: 103333. <https://doi.org/10.1016/j.specom.2025.103333>
- [29] Yi, J., Gu, Y., Xu, Y. (2025). Empowering speaker segmentation with self-supervised learning. *Electronics Letters*, 61(1): e70321. <https://doi.org/10.1049/ell2.70321>
- [30] Baeovski, A., Zhou, Y., Mohamed, A., Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33: 12449-12460.
- [31] Chung, Y., Hsu, W., Tang, H., Glass, J. (2019). An unsupervised autoregressive model for speech representation learning. In *Proceedings of Interspeech 2019*, Graz, Austria, pp. 146-150. <https://doi.org/10.21437/Interspeech.2019-1473>
- [32] van den Oord, A., Li, Y., Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*. <https://doi.org/10.48550/arXiv.1807.03748>