Neural Networks 141 (2021) 72–86

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2021 Special Issue

Deep joint learning for language recognition

Lin Li [a,*], Zheng Li [a], Yan Liu [a], Qingyang Hong [b,*]

[a] *School of Electronic Science and Engineering, Xiamen University, China*
[b] *School of Informatics, Xiamen University, China*

ARTICLE INFO

Article history:
Available online 26 March 2021

Keywords:
Deep joint learning
Language recognition
Language identification
Multi-feature learning
Multi-task learning

ABSTRACT

Deep learning methods for language recognition have achieved promising performance. However, most of the studies focus on frameworks for single types of acoustic features and single tasks. In this paper, we propose the deep joint learning strategies based on the Multi-Feature (MF) and Multi-Task (MT) models. First, we investigate the efficiency of integrating multiple acoustic features and explore two kinds of training constraints, one is introducing auxiliary classification constraints with adaptive weights for loss functions in feature encoder sub-networks, and the other option is introducing the Canonical Correlation Analysis (CCA) constraint to maximize the correlation of different feature representations. Correlated speech tasks, such as phoneme recognition, are applied as auxiliary tasks in order to learn related information to enhance the performance of language recognition. We analyze phoneme-aware information from different learning strategies, like joint learning on the frame-level, adversarial learning on the segment-level, and the combination mode. In addition, we present the Language-Phoneme embedding extraction structure to learn and extract language and phoneme embedding representations simultaneously. We demonstrate the effectiveness of the proposed approaches with experiments on the Oriental Language Recognition (OLR) data sets. Experimental results indicate that joint learning on the multi-feature and multi-task models extracts instinct feature representations for language identities and improves the performance, especially in complex challenges, such as cross-channel or open-set conditions.

1. Introduction

Language recognition is usually referred to Language Identification (LID), which evaluates whether a target language is spoken from auditory speech (Castaldo et al., 2010). LID and Automatic Speaker Verification (ASV) are two important tasks in speech processing. Since there are certain similarities between LID and ASV, some techniques are shared between them, e.g., i-vector models (Dehak, Kenny, Dehak, Dumouchel and Ouellet, 2011; Dehak, Torres-Carrasquillo, Reynolds and Dehak, 2011), and deep neural networks (Heigold, Moreno, Bengio, & Shazeer, 2016; Huang, Li, Yu, Deng, & Gong, 2013). The i-vector models (Dehak, Torres-Carrasquillo et al., 2011; Martinez, Plchot, Burget, Glembek, & Matejka, 2011) have prevailed as the state-of-the-art LID systems over the last decade. Different from Gaussian Mixture Model (GMM) based approaches, in i-vector models, utterances are compressed into fixed-length embeddings, and the cosine distance scoring along with Linear Discriminant Analysis (LDA) serves as the backend system. However, cosine distance scoring does not perform sufficiently. For better backend performance, Logistic Regression (LR) is usually applied as a classifier to model i-vectors in succession.

Recently, since neural networks have achieved outstanding performance over a wide range of applications, a great deal of research has focused on exploring Deep Neural Networks (DNN) in LID to improve performance in varying conditions. The first work in this area was the DNN i-vector (Lei, Scheffer, Ferrer, & McLaren, 2014) in which the posterior probability from the output layer of the DNNs in an ASR model replaced Gaussian components in Universal Background Model (UBM) for the i-vector computation. Another significant step forward in LID was the Bottleneck Feature (BNF) based i-vector (Richardson, Reynolds, & Dehak, 2015) in which the outputs of a bottleneck layer in an ASR model were used as features to be appended with acoustic features, then the combined features formed the input vectors for the i-vector model. Later, end-to-end systems (Kozhirbayev, Yessenbayev, & Karabalayeva, 2017; Lopez-Moreno et al., 2014) were employed to identify languages. However, it required larger datasets than the conventional i-vector models. More recently, motivated by the x-vector (Snyder, Garcia-Romero, Mccree, Sell and Khudanpur, 2018) in ASV, DNN based language embedding has become a popular approach, which aggregates

* Corresponding authors.
E-mail addresses: lilin@xmu.edu.cn (L. Li), qyhong@xmu.edu.cn (Q. Hong).

https://doi.org/10.1016/j.neunet.2021.03.026
0893-6080/© 2021 Elsevier Ltd. All rights reserved.

frame-level features into utterance-level representations (Sadjadi et al., 2018; Snyder, Garcia-Romero et al., 2018). Exploring a better loss function is also a hot topic. The center loss and angular softmax loss based methods are two widely applied loss functions, thus, many improvements have been presented with those two loss functions (Cai, Chen, & Li, 2018) to increase discrimination capabilities.

Meanwhile, there has been research in utilizing different learning strategies for language recognition. The cGAN-classifiers (Shen, Lu, Li, & Kawai, 2017) utilize real samples as conditional information to the generator network, while the discriminator network outputs the classified output related to labels.

Most of the existing research on LID has focused on schemas of single feature, single network framework, and single-speech task. Although studying individual schema improves model efficiency, the joint information from features, frameworks, and tasks is not shared, which has the potential to improve the performance of LID. Therefore, the motivation of this study is exploring joint learning from shared information in language recognition.

To encourage the development of LID technologies and to tackle real challenges existing in LID tasks, the OLR Challenge has been organized annually since 2016, attracting dozens of teams around the world (Li et al., 2020; Tang, Wang, & Chen, 2018; Tang, Wang, Chen, & Chen, 2017; Tang, Wang, & Song, 2019; Wang, Li, Tang, & Chen, 2016). We made significant achievements in previous OLR Challenges, including OLR Challenge 2018 and 2019. Additionally, we are one of the organizers of the OLR Challenge 2020, so exploring better performance on OLR datasets is meaningful in this study.

In this paper, we introduce deep joint learning to boost the language recognition system, aiming at utilizing relative information from features, frameworks, or tasks to improve the performance of LID. The proposed joint learning models consider different configurations for acoustic feature extraction and employ phoneme-aware information to optimize instinct feature representations for language identities. This goes beyond existing approaches that primarily use only one kind of acoustic feature and one classification task.

First, we analyze the integration structure of multiple acoustic features, and show the effectiveness of deep joint learning in which each subnet benefits from each other. We introduce the auxiliary classification constraints with adaptive weights of loss functions for feature encoder sub-networks. We also introduce CCA constraint learning to assist deep joint learning of multiple features, which enhances the correlation of multiple features, by maximizing the correlation between features.

In addition, we introduce multi-task learning (1) with frame-level phonetic information, (2) with adversarial learning of segment-level phonetic information, and (3) with the combination of frame-level joint learning and segment-level adversarial learning with phonetic information. Furthermore, we present the Language-Phoneme embedding extraction structure. For multi-task joint learning, the phonetic subnet works as an auxiliary support for the main task, i.e., the LID.

Experimental results show the superiority of deep joint learning, where language embedding extraction benefits from shared information correlated with different kinds of acoustic features or different tasks. The aforementioned methods are effective and more robust than baseline systems.

The main contributions of this paper are summarized as follows:

- This paper systematically analyzes deep joint learning of multiple acoustic features, and multiple speech tasks, for LID tasks, in different test conditions.
- The multi-loss constraint with adaptive weights makes the joint learning of multi-feature more flexible. The CCA constraint is proposed to extract correlated feature representations for multiple acoustic features, which improves the correlation between multiple features and multiple frameworks.
- Different strategies for joint learning on multi-tasks are discussed, like frame-level MT learning, segment-level adversarial learning, and the combined structures. A Language-Phoneme joint network is proposed to learn discriminative language and phonetic information simultaneously.

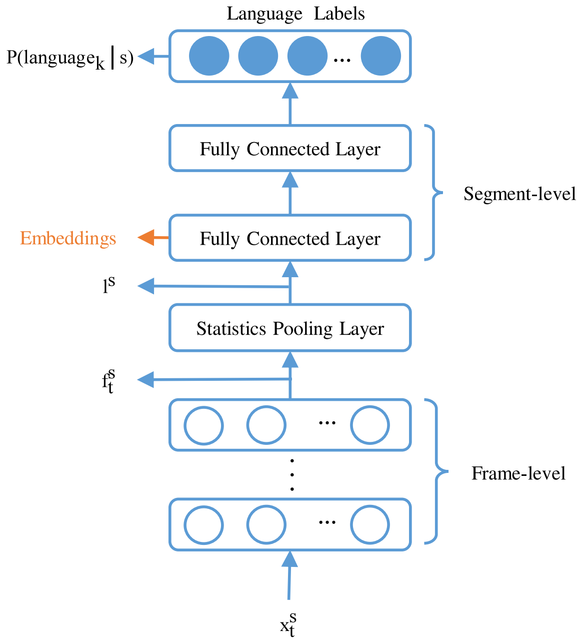


Fig. 1. The x-vector architecture.

The rest of the paper is organized as follows. The related works are introduced in Section 2. Section 3 and Section 4 present deep joint learning for multiple acoustic features and multiple tasks, respectively. In Section 5, the experimental settings are illustrated. Section 6 provides the experiments, results, and analyses. The conclusion is given in Section 7.

2. Related works

2.1. X-vector

The architecture for the language embedding extraction used in this paper is the x-vector (Snyder, Garcia-Romero, Sell, Povey and Khudanpur, 2018), as shown in Fig. 1. In the x-vector, the statistics pooling layer computes the mean and standard deviation, so that frame-level speech is aggregated into the segment level. For an input segment S with T frames, the feature $X^S = \{x_1^S, \ldots, x_t^S, \ldots, x_T^S\}$, $F_f\left(\cdot, \Theta_f\right)$ refers to frame level activations, l^S refers to the segment vector, and $F_s\left(\cdot, \Theta_s\right)$ refers to segment activations.

Then the Cross Entropy (CE) loss is expressed as:

$$L = CE\left(F_s\left(l^S; \Theta_s\right), y^S\right) \tag{1}$$

$$l^S = P\left(f_1^S, \ldots, f_T^S\right) \tag{2}$$

$$f_t^S = F_f\left(x_t^S; \Theta_f\right), (t = 1, 2, \ldots, T) \tag{3}$$

where y^S is the ground truth label of the utterance S, and $P(\cdot)$ indicates the statistical pooling operation from the 1st to the Tth frame of the utterance input X^S.

After training, the output of the penultimate layer at the segment level is extracted as the embeddings, which are named as x-vectors. After the LDA, the scoring model, like cosine distance, Probability Linear Discriminant Analysis (PLDA) or LR is further applied to the back-end processing to generate the scores.

Typically, two different x-vector architectures are widely used in the community: the Time Delay Neural Network (TDNN) based x-vector (Snyder, Garcia-Romero et al., 2018) and the ResNet based x-vector (Cai et al., 2018). The recently proposed extended x-vector architecture (E-TDNN) significantly outperforms the basic x-vector (TDNN) in most cases (Villalba et al., 2019), which uses a slightly wider temporal context in the TDNN layers, and it interleaves dense layers between TDNN layers, which leads to a deeper x-vector model. Since the OLR 2020 challenge chose an E-TDNN based x-vector as the baseline model for language embedding extraction, in this paper, the x-vector models we implemented are based on the E-TDNN, as well.

2.2. Multiple acoustic features

In LID evaluations, to improve the performance of the final submitted system, it is common for participants to fuse several subsystems at the score-level (Li et al., 2020; Sadjadi et al., 2018) with some score-level fusion toolkits (Brümmer, 2007; Brümmer & de Villiers, 2013; Rodriguez-Fuentes et al., 2013). The subsystems may utilize the same deep network structure but with different acoustic features.

Besides score-level fusion, studies were conducted on methods of combination or extension for acoustic features. Murty and Yegnanarayana (2006) proposed the residual phase feature as an additional feature for the MFCC feature. The BNF extraction model was introduced in Richardson et al. (2015). The tandem feature (Ravanelli, Do, & Janin, 2014) was proposed by splicing the BNF with basic acoustic features. In Li, He, Zhang, and Liu (2010), two acoustic features were concatenated directly to create a new feature vector with the LDA for reducing dimensions. In Zhao, Li, and Zhang (2019), an end-to-end framework was presented with an auxiliary feature learning branch, in which the feature learnt from the auxiliary branch is concatenated with the original acoustic feature in the model.

2.3. Multiple speech tasks

In speech signal processing fields, phonetic content, speaker identity, and language identity are the three most significant forms of encoded information. One of the important parts of speech recognition is to recognize phonetic content. Speaker recognition aims to recognize speaker identity, and language recognition refers to recognizing the language spoken in auditory speech. In speech recognition tasks, speaker adaptation techniques have been adopted to represent speaker-specific information (Karafiat, Burget, Matejka, Glembek, & Cernocky, 2012; Saon, Soltau, Nahamoo, & Picheny, 2013). The usage of phonetic information for speaker and language recognition tasks has also been studied.

There is a variety of multi-task learning strategies in speaker recognition. In Kenny, Gupta, Stafylakis, Ouellet, and Alam (2014) and Lei et al. (2014), by substituting the GMM with an ASR model's output posteriors, the performance of the DNN i-vector system was significantly improved. Recently, in Wang et al. (2019), the authors introduced phonetic information in a speaker embedding learning framework. In Liu, He, Liu, and Johnson (2018), speaker and phoneme classification networks were trained jointly to compose a multi-task learning framework. In Yang, Wang, Gong, Qian, and Yu (2020), the authors proposed a framework for speaker-text factorized embedding for speaker recognition. In order to factorize input speech into speaker-text embeddings, two embeddings of speakers and texts were integrated into a single representation in a higher level model, and all loss functions of sub-nets were jointly optimized.

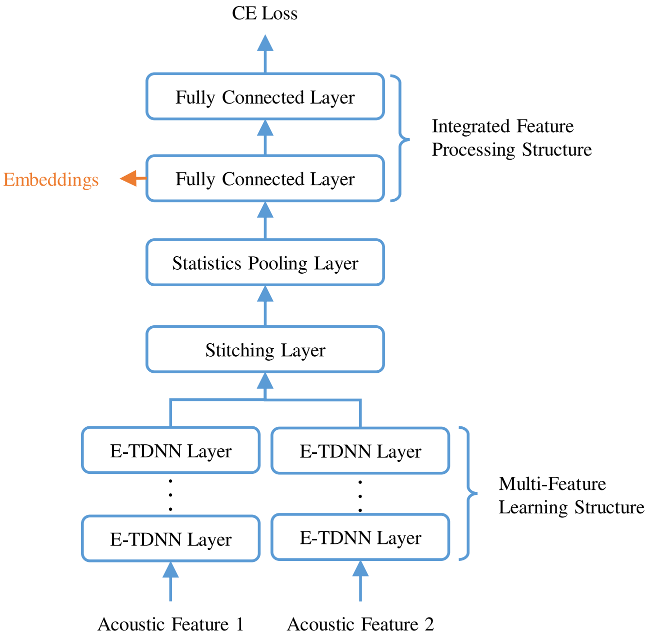


Fig. 2. The frame-level multi-feature learning structure.

There is still a lot of research space for joint learning of phoneme information in the field of language recognition. In Tian, He, Liu, and Liu (2016), the senone-based Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) was investigated to model long-range correlations in speech signals. In Tang, Wang, Chen, Li, and Abel (2018), a phonetic temporal model for LID was proposed in which phonetic feature outputs of a phone-aware DNN were applied as the inputs of an LSTM-RNN system. In our previous work (Zhao et al., 2019), we introduced the frame-level phonetic multi-task learning framework into language recognition tasks and achieved significant improvements. However, to the best of our knowledge, there is still insufficient research on deep joint learning of multiple speech tasks for language recognition.

3. Deep joint learning of multiple acoustic features

3.1. Multi-feature learning

In our previous work, the frame-level multi-feature integration method was studied (Li, Lu, Zhou, Li, & Hong, 2019) with the application of speaker verification, as shown in Fig. 2. For the purpose of utilizing the correlation of acoustic features, we have analyzed and proposed the multiple acoustic features integration structure. Based on such a structure, two kinds of acoustic features are used to train a deep neural network model of language recognition simultaneously, and two shallow-layer features are integrated as a fused feature before the pooling operation. This structure represents the characteristics of deep joint learning. It is a joint neural network model, and the structure of each branch is basically similar. Therefore these homogeneous components share correlative information with each other. Let $x^S_{t,1}$, $x^S_{t,2}$ denote the two kinds of acoustic feature vectors, such as MFCC and

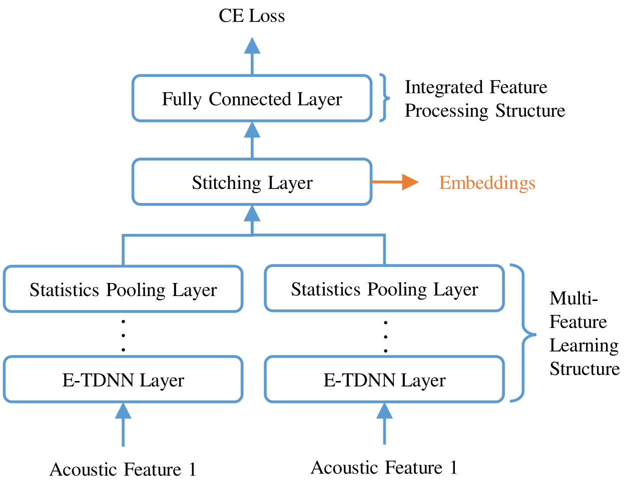


Fig. 3. The segment-level multi-feature learning structure.

PLP, from the same speech S, and f_t^S represents the frame-level integrated feature:

$$f_t^S = F_{f,3}\left(cat\left(F_{f,1}\left(x_{t,1}^S;\Theta_1\right), F_{f,2}\left(x_{t,2}^S;\Theta_2\right)\right);\Theta_3\right) \tag{4}$$

where $cat(\cdot)$ denotes the concatenating operation. $F_{f,1}(\cdot;\Theta_1)$ is the pre-projection of acoustic feature 1 given the network parameters Θ_1, and the same for $F_{f,2}(\cdot;\Theta_2)$. $F_{f,3}(\cdot;\Theta_3)$ indicates the transform in the integration block with network parameters Θ_3.

As illustrated above, multiple features can be integrated at the frame level and be jointly trained within an x-vector model, which yields significant improvements in ASV tasks (Li et al., 2019). In this study, we further investigate the possibility of assembling features at higher levels, such as the segment-level, as shown in Fig. 3. This architecture integrates multiple features after the statistics pooling layers, where the language discriminative information is learned independently with each feature, and then is integrated at the segment level. If we let l^S represent the segment-level integrated features, the computation can be written as:

$$l^S = F_{s,3}\left(cat\left(P\left(F_{f,1}\left(x_{t,1}^S;\Theta_1\right)\right), P\left(F_{f,2}\left(x_{t,2}^S;\Theta_2\right)\right)\right);\Theta_3\right) \tag{5}$$

where $x_{t,1}^S$ represents the tth input acoustic feature 1 for the T-frame utterance S, and the same for $x_{t,2}^S$; $F_{f,1}(\cdot;\Theta_1)$ is the pre-projection of feature 1 given the network parameters Θ_1, and the same for $F_{f,2}(\cdot;\Theta_2)$; $P(\cdot)$ refers to the statistical pooling operation, which computes the mean and standard deviation accumulated from the 1st to Tth frame, and $F_{s,3}(\cdot;\Theta_3)$ is the segment level computation in fully connected layers with network parameters Θ_3.

3.2. Multi-loss learning with adaptive weights

In our previous work, as mentioned in Section 2, different acoustic features are encoded with two parallel feature extraction branches and trained under a CE loss in the integrated feature processing structure. However, using a CE loss to lead the training only emphasizes the language discriminative ability of integrated feature, and it omits the classification ability of feature branches in multi-feature learning structures, which is also important for language extraction.

Thus, we propose a multi-loss learning structure for multi-feature learning. As shown in Fig. 4, the proposed method introduces a branch constraint loss structure with additional CE losses and adaptive weights. In Fig. 4, the learnt adaptive weight δ_i is the output value from an additional output layer, in which there is only one node. The branch constraint loss $\pounds_i$ with a parameter δ_i is learned for feature i, by using the weight uncertainty approach in Cipolla, Gal, and Kendall (2018), which is written mathematically as:

$$\pounds_i = \frac{1}{2\delta_i^2} L_i + log\left(\delta_i\right) \tag{6}$$

where L_i refers to the classification loss function of feature $i (i = 1, 2)$.

From formula (6), it is obvious that, if the condition $\delta_i^2 > 0.5$ is matched, which is easy to achieve in our experiments, the losses of feature branches are reduced. Meanwhile, the constraint $log(\delta_i)$ ensures the parameter δ_i^2 does not increase without limit.

The total loss L_{total} can be written as:

$$L_{total} = \pounds_1 + \pounds_2 + L_c \tag{7}$$

where $\pounds_1$ and $\pounds_2$ are the weighted branch CE losses, and L_c indicates the CE loss in the main branch.

Adaptive weight is introduced to balance the losses in the subnetworks for feature extraction, and to control those losses in a relatively low level compared with the main loss.

3.3. CCA constraint learning

3.3.1. Canonical correlation analysis

Canonical Correlation Analysis is an algorithm for learning representations of two views of data, so that the predictability and correlation of each view is the highest, simultaneously (Hardoon, Mourão Miranda, Brammer, & Shawe-Taylor, 2007; Ngiam et al., 2011; Vinokourov, Shawe-Taylor, & Cristianini, 2002). CCA has been used for unsupervised data learning when multiple views are available. If we let $(X_1, X_2) \subset \mathbb{R}$ denote vectors from two perspectives of a data representation, the objective of CCA can be written as:

$$\left(w_1^*, w_2^*\right) = \underset{(w_1,w_2)}{argmax}\, corr\left(w_1' X_1, w_2' X_2\right) \tag{8}$$

where w_1', w_2' are the transpositions of w_1, w_2, and $\left(w_1^*, w_2^*\right)$ represents pairs of linear projections from two views, such that the correlation of $\left(w_1' X_1, w_2' X_2\right)$ is the maximum.

With the usage of DNN, deep CCA (DCCA) (Andrew, Arora, Bilmes, & Livescu, 2013) replaces linear projections with nonlinear transformations, shown in Fig. 5. If Θ_1 represents all parameters of the network for view 1, and Θ_2 indicates all parameters of the network for view 2, the objection is:

$$\left(\Theta_1^*, \Theta_2^*\right) = \underset{(\Theta_1,\Theta_2)}{argmax}\, corr\left(F_1\left(X_1;\Theta_1\right), F_2\left(X_2;\Theta_2\right)\right) \tag{9}$$

where $F_1(X_1;\Theta_1)$ is the nonlinear transform for X_1 with network parameters Θ_1 and the same for $F_2(X_2;\Theta_2)$. To reach $\left(\Theta_1^*, \Theta_1^*\right)$, the gradient of the correlation objective, which is estimated on the training data, is used to update parameters of the DNNs.

3.3.2. CCA constraint loss

As mentioned above, one of CCA's advantages is to eliminate noisy and redundant features in uncorrelated dimensions from two views. Meanwhile, one significant purpose of the multi-feature learning structures is to extract the instinct representations from different kinds of features. By connecting the commonalities of CCA and multi-feature learning structures, we propose a new CCA constrained multi-feature learning approach, as shown in Figs. 6 and 7. In brief, this approach introduces a correlation constraint on the outputs of different feature layers, which can be formulated as a regularization term L_s in the training objective with the CE main loss function L_c:

$$L = L_c + \alpha L_s \tag{10}$$

where α controls the strength of the regularization.

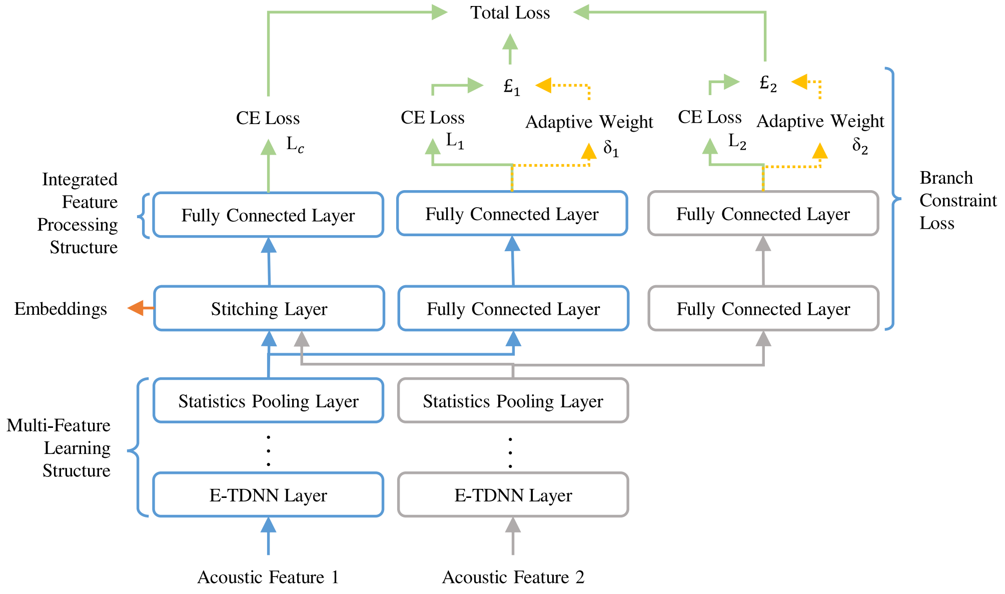


Fig. 4. The multi-loss learning structure with adaptive weights.

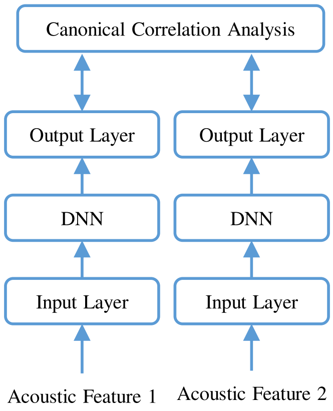


Fig. 5. The deep CCA structure.

Since the randomness and fluctuation in each frame between different features may be significant, we only apply the CCA constraint loss on the segment level, or the utterance level.

The formula of the CCA constraint loss is as follows:

$$L_s = -corr\left(P\left(F_{f,1}\left(x_{t,1}^S;\Theta_1\right)\right), P\left(F_{f,2}\left(x_{t,2}^S;\Theta_2\right)\right)\right) \tag{11}$$

where $x_{t,i}^S\,(i=1,2)$ indicates the tth frame in the segment S of speech, and this segment has T frames.

No matter the frame-level computation or the segment-level computation, the regularization L_s encourages all segment-level representations in a feature view to correlate to the other feature view. Therefore, this new multi-feature learning approach is referred to as multi-feature learning with CCA constraint.

3.3.3. *CCA constraint layer*

Although using two feature views $P\left(F_{f,1}\left(\cdot\right)\right)$ and $P\left(F_{f,2}\left(\cdot\right)\right)$ as the inputs for the CCA constraint computation is logically practicable, some disadvantages emerge: (1) it leads to high computational costs for CCA because of the high dimensions of the feature views, e.g. 512 dimensions; (2) it would generate the 'NaN problem' in the CCA calculation when the activation functions is Rectified Linear Unit (ReLU); (3) the direct utilization of $P\left(F_{f,1}\left(\cdot\right)\right)$ and $P\left(F_{f,2}\left(\cdot\right)\right)$ to compute the CCA loss during training may introduce constraints that are too strong for the networks, which contributes to the low quality of the training, even with the control of α.

To deal with the above problems, a corresponding CCA constraint layer is proposed, and the internal structure of the CCA constraint layer is shown in Fig. 8. The CCA constraint layer consists of a normal fully connected layer with ReLU, a fully connected layer with a Tanh activation function, which has lower dimensions, such as 32 dimensions, for each view. The CCA layer is placed between the feature view subnets and the computation of the CCA constraint loss, as shown in Figs. 6 and 7. After training, the CCA constraint layer is removed, so the model size is the same as the original multi-feature learning model.

4. Deep joint learning of multiple tasks

4.1. *Multi-task learning with phonetic information*

In this paper, we study the multi-task learning architecture (Liu et al., 2018) based on the E-TDNN framework (Villalba et al., 2019), as shown in Fig. 9.

The phonetic multi-task learning framework contains three modules, including the shared E-TDNN feature encoder M_e at the frame-level, the phoneme classifier M_p at the frame-level and the language classifier M_c at the segment-level. Given a segment S of T frames $X^S = \{x_1^S, \ldots, x_t^S, \ldots, x_T^S\}$, the total loss of multi-task learning is composed of the language classification loss L_c and the phoneme classification loss L_p with an empirical control factor β, written as:

$$L_{total} = L_c + \beta L_p \tag{12}$$

$$L_c = CE\left(M_c\left(M_e\left(x_t^S\right)\right), y^S\right) \tag{13}$$

$$L_p = \frac{1}{T}\sum_{t=1}^{T} CE\left(M_p\left(M_e\left(x_t^S\right)\right), y_i^p\right) \tag{14}$$

where CE(A, B) indicates the CE loss computed between the two distributions A and B. y^S denotes the segment-level language label, and y^p is the frame-level phoneme label.

In Wang et al. (2019), researchers proposed a segment-level phonetic label, which was the primary kernel of segment-level phonetic multi-task learning. The procedural flow for segment-level phonetic multi-task learning is shown in Fig. 10. Given a segment S with T frames, the corresponding segment-level phoneme label y^P is defined as:

$$y^P = \{y_1, y_2, \ldots, y_c \ldots, y_C\} \tag{15}$$

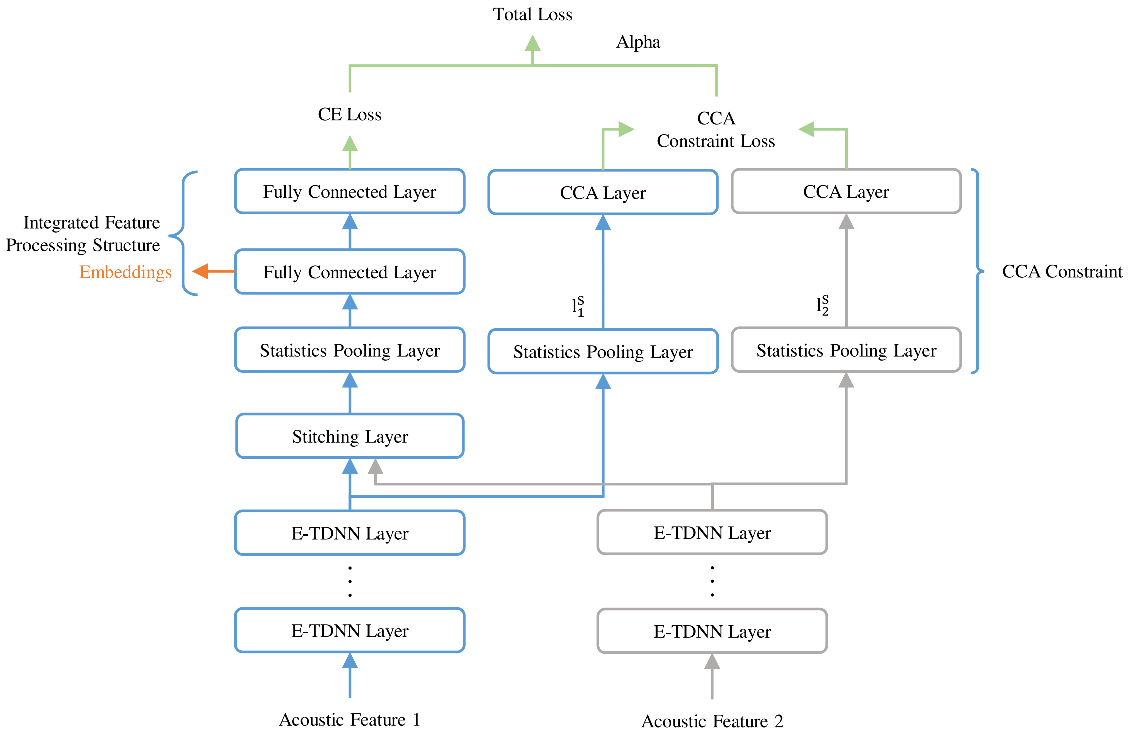


Fig. 6. The frame-level multi-feature learning structure with CCA constraint.

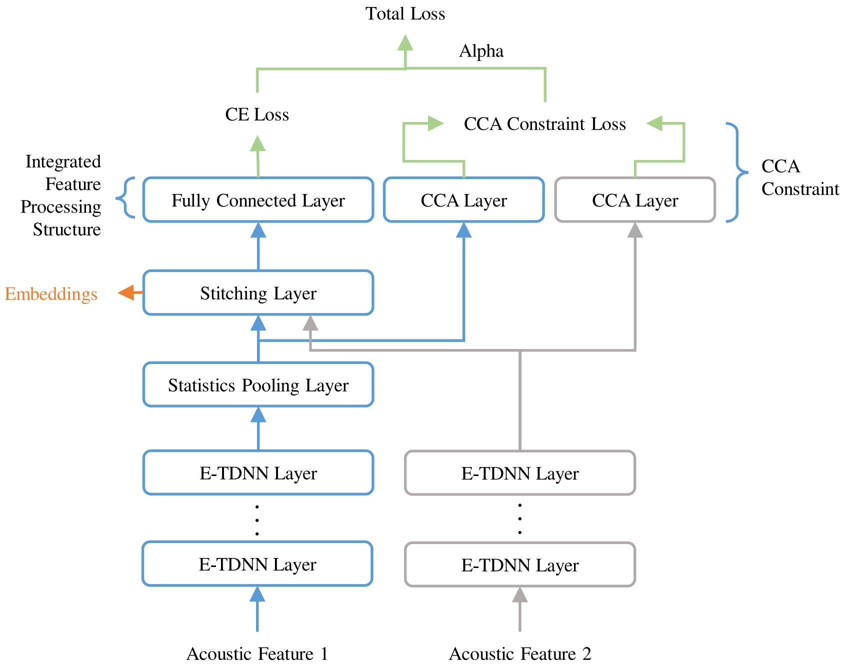


Fig. 7. The segment-level multi-feature learning structure with CCA constraint.

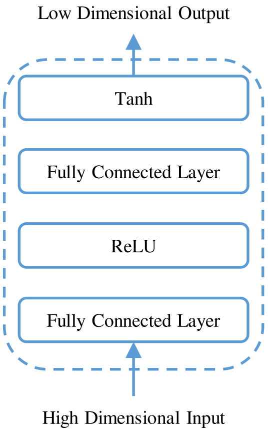


Fig. 8. The internal structure of the CCA constraint layer.

$$y_c = \frac{T_c}{T}, (c = 1, \ldots, C) \tag{16}$$

where C is the total number of the phoneme set, and T_c denotes the number of the c-th phoneme observed in segment S.

4.2. Adversarial learning for phonetic information suppression

As shown in Fig. 11, segment-level phonetic information can also be suppressed by a Gradient Reversal Layer (GRL). The GRL is added to the sub-network for phoneme classification, which aims to reduce the influence of the segment-level phonetic information.

In the GRL, the forward propagation computation remains the same as in the normal hidden layers, while reversing the value of the gradient in the backward propagation.

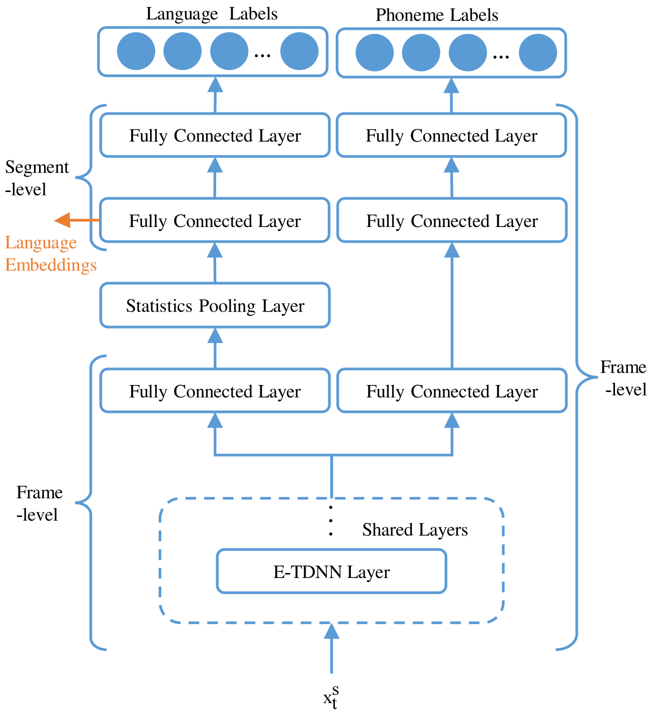


Fig. 9. The frame-level phonetic multi-task learning structure.

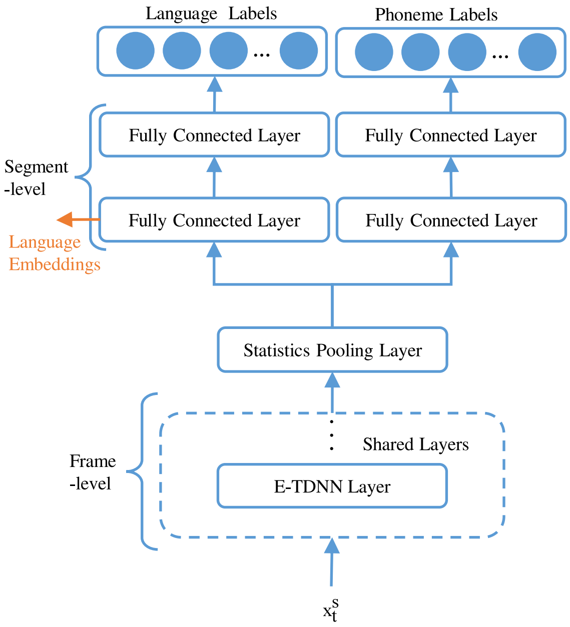


Fig. 10. The segment-level phonetic multi-task learning structure.

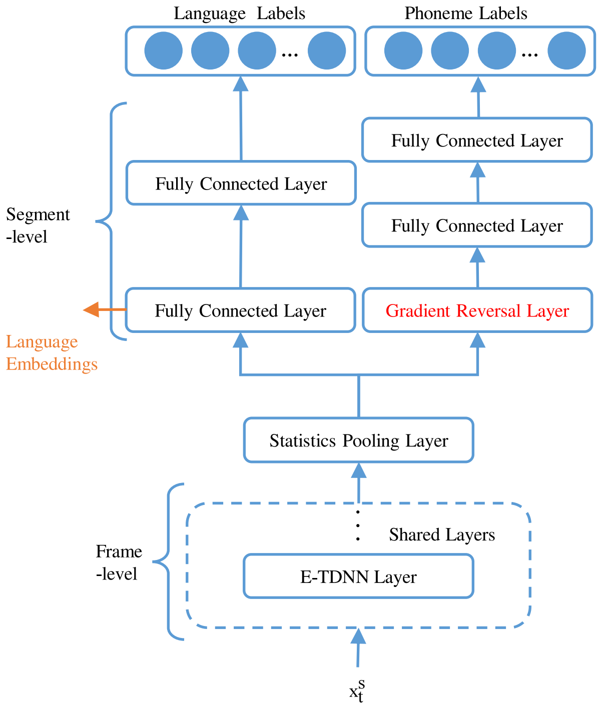


Fig. 11. The adversarial learning structure for phonetic information suppression.

4.3. Deep joint learning for multi-task and adversarial learning

We combine the previous structures, frame-level multi-task learning with phonetic information and segment-level adversarial learning for phonetic information suppression, to create a single, fused network with improved performance, as shown in Fig. 12.

4.4. Multi-task learning with language-phoneme embedding

Inspired by Yang et al. (2020), with the help of phonetic information in the LID training, we introduce a framework of

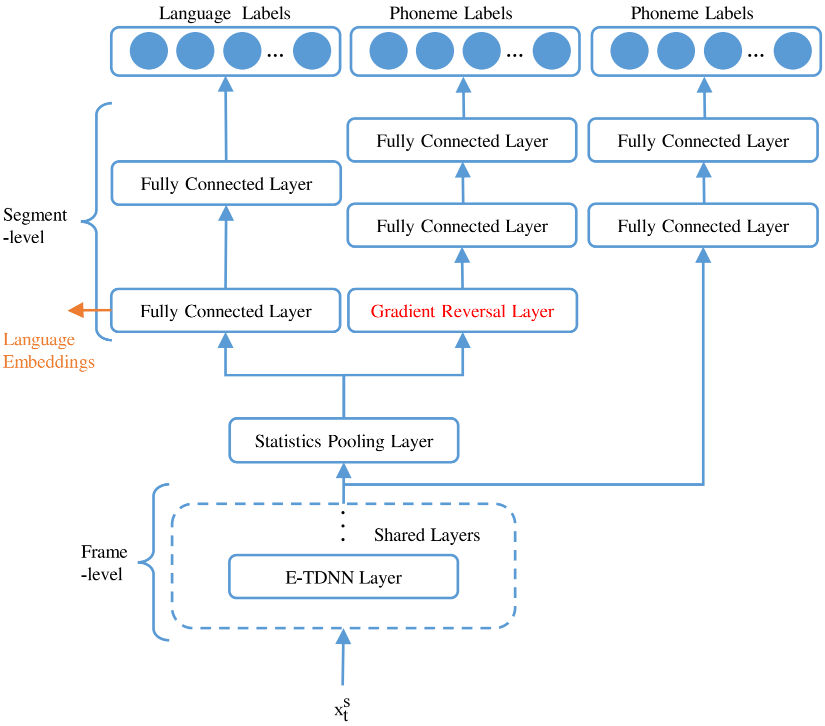


Fig. 12. The deep joint learning of multi-task and adversarial learning structure.

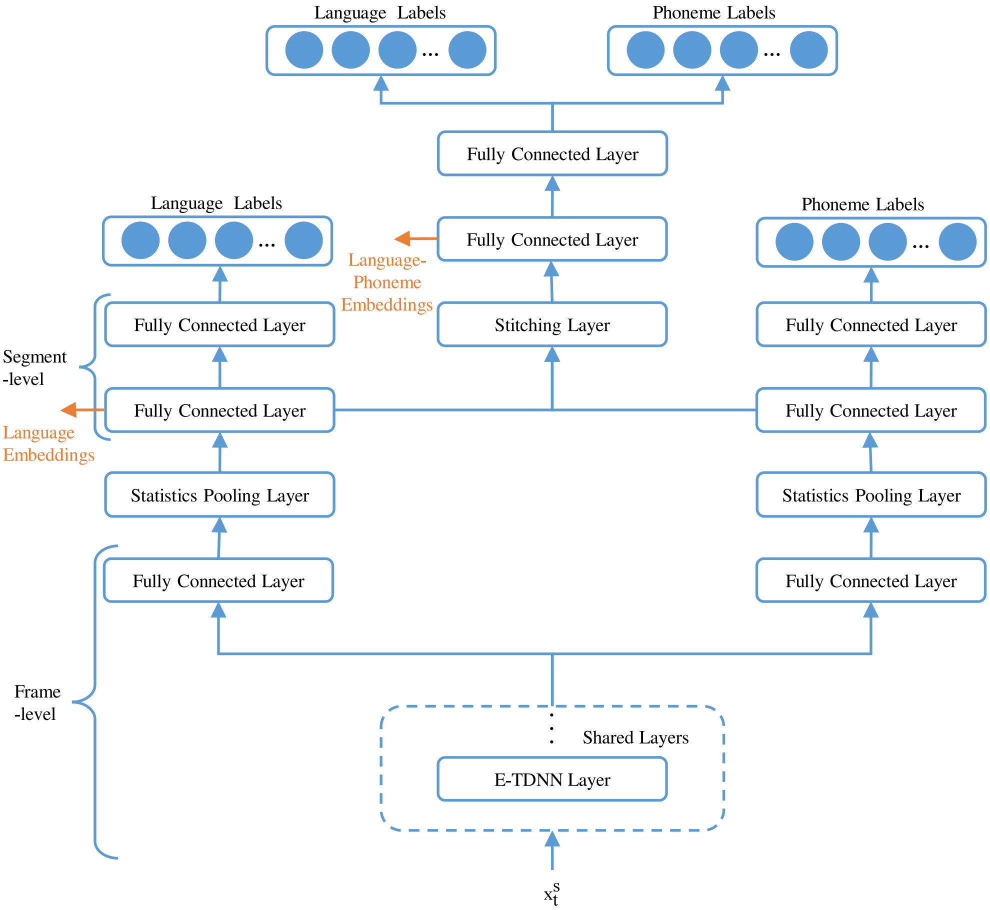


Fig. 13. The Language-Phoneme embedding learning structure.

Language-Phoneme embedding extraction for language recognition, as shown in Fig. 13. The Language-Phoneme learning structure is used to learn language and phonetic information simultaneously, given that the phoneme distributions of different languages are significantly distinctive. Thus, the Language-Phoneme embedding is used as input in the back-end processing of language identification.

5. Experimental settings

5.1. Data sets

The language identification database used in our experiments is the OLR dataset (Li et al., 2020; Tang et al., 2018, 2017, 2019; Wang et al., 2016) for both training and testing. Table 1 details the content of the dataset.

5.1.1. Training set

In the OLR challenges, additional training materials were prohibited and participants were only permitted to use several specified datasets, including AP16-OL7, AP17-OL3, AP17-OLR-test, AP18-OLR-test, and THCHS30 (plus the accompanying resources) (Wang & Zhang, 2015). The AP16-OL7 database incorporates seven languages, including Mandarin, Cantonese, Indonesian, Japanese, Russian, Korean, and Vietnamese, which are all oriental languages. Each language in the database has about 10 h of speech signals recorded in reading style. The AP17-OL3 database contains three languages, Kazakh, Tibetan, and Uyghur, which are minority languages in China. The data volume for each language in AP17-OL3 contains about 10 h of speech signals recorded in reading style. The THCHS30 contains 30 h of Chinese ASR training speech, with lexicon and dictionary files.

The standard training set in the Evaluation Plan of OLR Challenge 2020 consists of AP16-OL7, AP17-OL3, AP17-OLR-test (Tang et al., 2017; Wang et al., 2016). Thus, this standard training set was used as the training data for all models in this study, including ten oriental languages, with about 140.2 h of speech.

Before training, we carried out data augmentation, including speed and volume perturbation, to increase the amount and diversity of the training data. For speed perturbation, we applied a speed factor of 0.9 or 1.1 to slow down or speed up the original recording. For volume perturbation, the random volume factor was applied. Finally, two augmented copies of the original recording were added to the original data set to obtain a 3-fold combined training set.

5.1.2. Test set

To better investigate of the proposed methods, three kinds of test sets were used, which correspond to three particular LID conditions, including short utterance LID, cross-channel LID, and open-set dialect identification.

For short utterance LID, the AP18-OLR-test-short was used as the test set, with about 5.8 h of speech. This data set is a close-set identification set, in which the language of the utterances is also one of the ten target languages, as in the training set, but the test utterances are only 1 s long.

For cross-channel LID, the AP19-OLR-test-channel was used, with about 8.9 h of speech. This test data is recorded in different channels and is also a close-set identification set, including six known target languages: Cantonese, Indonesian, Japanese, Russian, Korean, and Vietnamese.

For open-set dialect identification, the AP20-OLR-dev-task2 was used, with about 10.21 h of utterances in total. The AP20-OLR-dev-task2 contains the dialect test subset of AP19-OLR-dev-task3, which contains three target dialects: Hokkien, Sichuanese, and Shanghainese, and the test subset of AP19-OLR-test-task3, which contains three nontarget (interfering) languages: Catalan, Greek, and Telugu.

5.1.3. OLR 2020 challenge

The AP20-OLR-test in the latest OLR 2020 Challenge (Li et al., 2020) was also introduced to investigate the performance of the multi-task learning systems, which were based on THCHS30 alignment. The AP20-OLR-test contained three tasks: (1) cross-channel LID, (2) dialect identification, and (3) noisy LID. Table 2 details the AP20-OLR-test dataset.

5.2. Phoneme labels for multi-task learning

THCHS30 (Wang & Zhang, 2015) is an allowed-to-use data set in the OLR Challenges, which contains the lexicon and dictionary files for 30 h of Chinese ASR training speech. Thus, based on the THCHS30 training data, according to the Kaldi's recipe (Povey et al., 2011), a *tri4b* (tri-phone) GMM-based ASR model was trained, including 3,604 Probability Density Function Identifications (PDF-IDs) . For the purpose of accurate phoneme alignment, we used the PDF-IDs in each frame to represent the corresponding phoneme label. Finally, the phoneme labels for the OLR training set were decoded from the trained ASR model.

To investigate the influence of ASR models trained on different languages and training sizes, we also used Librispeech (Panayotov, Chen, Povey, & Khudanpur, 2015), which is a corpus of read English speech, to train an English based ASR model to obtain phoneme labels for the OLR training set. Due to the larger amount of training speech, according to the Kaldi's recipe (Povey et al., 2011), the *tri6b* (tri-phone) GMM-based ASR model was trained on Librispeech with 5704 PDF-IDs. The decoding procedure for the OLR training set was the same as that of THCHS30. It should be noted that neither the THCHS30 nor Librispeech based ASR models are multi-lingual ASR model. Thus, the phoneme labels of the multi-lingual LID training set are the virtual phoneme labels.

Table 3 details the THCHS30 and Librispeech based ASR models.

5.3. Experimental settings

All models were trained on 16 kHz data. There are two kinds of acoustic features: one is 20-dimensional PLP with 3-dimensional pitch, and the other is 20-dimensional MFCC with 3-dimensional pitch. All acoustic features had frame-length of 25 ms, frame shifts of 10 ms, and mean normalization over a sliding window of up to three seconds. Voice Active Detection (VAD) was used to filter out non-speech frames, and different acoustic features shared the same VAD based on MFCC, for the frame alignment.

The back-end process was the same for all three test sets when the embeddings were extracted, as mentioned in the baseline system on OLR 2020 Challenge (Li et al., 2020). LDA was trained based on the enrollment set and was employed to promote language-related information. The dimensionality of the LDA projection space was set to 100. After LDA projection and centering, LR was trained based on the enrollment set and was used to compute the score of a trial on a particular language.

The proposed models for language embedding extraction were implemented on the open-source speaker and language recognition toolkit ASV-subtools (Tong, Zhao, Zhou, Lu, Li, Li, & Hong, 2021), which is developed based on PyTorch (Paszke et al., 2017). The front-end models were optimized with Adam optimizer, with a mini-batch size of 512, and a chunk size of 100 in each segment, while the acoustic feature extraction and the back-end process were executed on Kaldi (Povey et al., 2011). In multi-feature learning models with CCA constraint, the control factor α mentionded in Equation (10) was set as 0.1. In multi-task learning, the control factor β mentionded in Equation (12) was set as 0.2 for all models.

Table 1
Dataset details.

	Task	Languages	No. of utterances	Hours	Channel
Training set	Training	Mandarin, Cantonese, Indonesian, Japanese, Russian, Korean, Vietnamese, Kazakh, Tibetan, and Uyghur	282,855	140.2	Mobile
Test set	Short utterance	Mandarin, Cantonese, Indonesian, Japanese, Russian, Korean, Vietnamese, Kazakh, Tibetan, and Uyghur	21,456	5.8	Mobile
	Cross-channel	Cantonese, Indonesian, Japanese, Russian, Korean, and Vietnamese	10,800	8.9	Unknown
	Open-set	Hokkien, Sichuanese, Shanghainese, Catalan, Greek, and Telugu	6900	10.21	Mobile

Table 2
The details of the AP20-OLR-test from the OLR 2020 Challenge.

Task	Language	No. of speakers	No. of male	No. of female	Duration (h)	No. of utterances	Channel	Recording environment
1	Cantonese	6	3	3	3.11	2394	Cross channel	Quiet
	Indonesian	6	3	3	3.2	1800		
	Japanese	6	3	3	2.21	2254		
	Russian	6	3	3	3.87	1800		
	Korean	6	3	3	2.3	1800		
	Vietnamese	6	3	3	3.25	1800		
2	Sichuanese	6	3	3	2.31	1800	Mobile channel	Quiet
	Shanghainese	6	3	3	2.26	1800		
	Hokkien	6	3	2	3.23	1998		
	Malay	6	3	2	3.72	2000		
	Mandarin	6	3	3	2.57	1800		
	Thai	6	2	3	1.83	2000	Cross channel	
3	Cantonese	6	2	3	2.71	1965	Mobile channel	Noisy
	Japanese	6	3	3	1.99	1862		
	Russian	6	7	6	3.61	1944		
	Korean	6	7	6	2.26	1925		
	Mandarin	6	3	3	2.53	1800		

5.4. Evaluation metrics

In this paper, we use two metrics to evaluate the language recognition systems, which are *Cavg* and the Equal Error Rate (EER), respectively.

5.4.1. Cavg

In most language recognition challenges, such as LRE and OLR, *Cavg* is chosen as the principle evaluation metric. We first define the pair-wise loss that composes the missing and false alarm probabilities for a particular target/non-target language pair:

$$C\left(L_t, L_n\right) = P_{Target} P_{Miss}\left(L_t\right) + \left(1 - P_{Target}\right) P_{FA}\left(L_t, L_n\right) \tag{17}$$

where L_t and L_n are the target and non-target languages, respectively; P_{Miss} and P_{FA} are the missing and false alarm probabilities, respectively. P_{Target} is the prior probability for the target language, which is set to 0.5 in the OLR evaluations. Then the principal metric *Cavg* is defined as the average of the above pair-wise performance:

$$C_{avg} = \frac{1}{N} \sum_{L_t} \left\{ P_{Target} P_{Miss}\left(L_t\right) + \sum_{L_n} P_{Nontarget} P_{FA}\left(L_t, L_n\right) \right\} \tag{18}$$

where N is the number of languages, and $P_{Nontarget} = (1 - P_{Target})/(N - 1)$. For the open-set testing condition, all of the interfering languages are treated as one unknown language in the computation of *Cavg*.

5.4.2. Equal error rate

In the classification task, two basic metrics are considered, which are False Rejected Ratio (FRR), and False Accepted Ratio (FAR). The FRR and FAR are written as:

$$P_{FRR} = \frac{Number\ of\ target\ trials\ rejected}{Number\ of\ Total\ target\ trials} \tag{19}$$

$$P_{FAR} = \frac{Number\ of\ nontarget\ trials\ accepted}{Number\ of\ Total\ nontarget\ trials} \tag{20}$$

If we use P_{FRR} as the vertical axis, and P_{FAR} as the horizontal axis, a continuous curve of FAR corresponding to FRR is obtained, named the Detect Error Trade-off (DET).

The EER is defined as the value in the DET curve, where $P_{FRR} = P_{FAR}$, and EER is also a widely-used metric for evaluating language recognition systems.

6. Experimental results

6.1. Multiple acoustic feature learning

The experimental results on deep joint learning of multiple acoustic features are listed in Table 4 and reported in terms of *Cavg* and EER.

We first reproduced the baseline systems, according to the AP20-OLR Challenge (Li et al., 2020), with two kinds of acoustic features, as reported in system No. 1 and system No. 2 in Table 4. From the comparison, it is apparent that the baseline system based on MFCC features slightly outperformed the one with PLP features, in three tasks.

Then, score-level fusion on baseline systems No. 1 and No. 2 was investigated, with the equal fusion weight. Score-level fusion brought improvements to the short-utterance task, while the

Table 3
The Corpus for the training of ASR models.

Corpus	Language	Hours	ASR model	PDF-IDs	OLR Challenge
THCHS30	Chinese	30	*tri4b* GMM	3604	Permitted
Librispeech	English	1000	*tri6b* GMM	5704	Not permitted

performance of the cross-channel task and the open-set dialect task were corrupted. System No. 4 is the system used the direct-concatenated acoustic features as input, which outperformed the score-level fusion system on the cross-channel task, which denotes the benefit of multiple acoustic features. However, the score-level fusion system obtained the best performance on the short-utterance task with a *Cavg* value of 0.0484 and an EER value of 5.09%.

System No. 5 and No. 6 adopted the multiple acoustic feature learning models, as introduced in Section 3.1, with the frame-level and the segment-level implementations, respectively. It is observed that, in both implementations, the introduction of multiple acoustic learning boosted the system performance on the short-utterance task and the cross-channel task, compared with the baseline systems. The frame-level multi-feature learning slightly outperformed the segment-level model on the cross-channel task, while the segment-level model was better on the short-utterance task. The reason behind this may be attributed to the fact that the frame-level multi-feature learning focuses on more general and global representations, while the segment-level multi-feature learning structure explores more specic information among the languages. In the comparison of multiple acoustic feature learning systems and score-level fusion systems, the score-level fusion system achieved better performance at short-utterance tasks, while the robustness of multiple feature learning was revealed on the cross-channel task. This may be due to the fact that there is limited language related information in short utterances, which might be further compressed with multiple acoustic feature learning but insufficient for representing language identities.

The proposed multi-loss learning with adaptive weights system is reported in system No. 7. Compared with system No. 6, which was the single-loss learning model, the presented multi-loss learning strategy achieved stable improvements on all three tasks. In addition, the training loss curves of the multi-loss learning (system No. 7) and the basic single-loss learning systems (system No. 6) are shown in Fig. 14(a). From Fig. 14(a), it can be observed that with the deep joint learning of multi-loss, and with the adaptive weights, the loss curve was lower and more stable than the single-loss system. Furthermore, the accuracy curves of the multi-loss learning and the basic single-loss learning systems are shown in Fig. 14(b), and it is obvious that the multi-loss system obtained stable and higher accuracy during the whole training stage.

The results of multi-feature learning models with CCA constraints are listed in system No. 8 and No. 9. In both the frame-level and the segment-level multi-feature learning models, the CCA constraint learning led to constant improvements, in two tasks. The best performance of multi-feature learning systems on the cross-channel task was obtained by frame-level multi-feature learning with CCA constraints, with a *Cavg* value of 0.2253 and an EER value of 22.60%. Meanwhile, the best performance of multi-feature learning systems on the open-set dialect identification task was achieved by segment-level multi-feature learning with CCA constraints, with a *Cavg* value of 0.0743 and an EER value of 9.87%.

6.2. Multiple task learning systems

The experimental results on deep joint learning for multiple task learning are listed in Tables 5, 6, and 7, and reported in terms of *Cavg* and EER.

6.2.1. Investigation of phoneme labels

According to the OLR Challenge regulations, multi-lingual phoneme labels were prohibited, thus, we used the permitted data set THCHS30 to train an ASR model and then to obtain the virtual phoneme labels for the multi-lingual language training set. Meanwhile, regardless of the competition constraints, we also investigated the influence of virtual phoneme labels achieved from an English ASR model for multi-task learning, shown in Table 5. The E-TDNN x-vector baseline system and the frame-level multi-task learning method with different languages' ASR models are compared in three tasks. We observe that, whether the Chinese or English ASR model was used to get the phoneme labels, the introduction of phonetic information into the LID model training reliably improved the performance of LID systems. But the performance of Librispeech based frame-level multi-task learning system outperformed that of THCHS30 based system, on the cross-channel task and the open-set task. We assume this is due to the larger amount of ASR training data for Librispeech. The more accurate ASR decoding model and more PDF-IDs were achieved, to introduce more accurate phonetic information, which is crucial to multi-task learning.

Thus, to investigate the potential of multi-task learning methods, we also used phoneme labels from the decoding process of the Librispeech ASR model to train multi-task learning models in the rest of experiments in this paper.

6.2.2. Multi-task learning systems

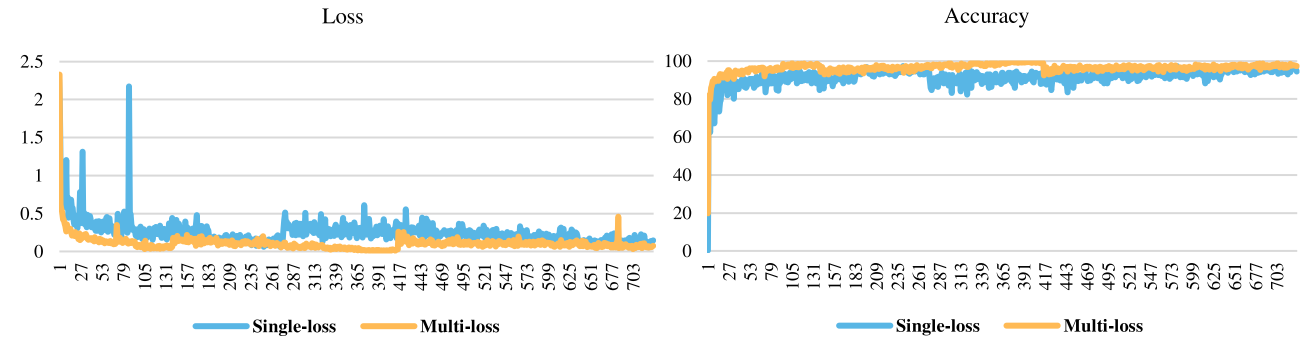
With the THCHS30 and Librispeech ASR model based phoneme labels, the results of different multi-task learning systems in this paper are listed in Tables 6 and 7. An additional comparison on the latest OLR 2020 test sets is shown in Table 8.

Whether frame-level or segment-level multi-task learning, the introduction of phonetic information into the LID model training drastically improved the system performance. Furthermore, frame-level phonetic information contributed more than segment-level phonetic information in multi-task learning, and the performance gap between the frame-level and the segment-level systems was significant on the systems with THCHS30 alignment.

Adversarial training at segment level with a GRL obtained greater performance improvement compared with segment-level multi-task learning in most cases.

When we further combined frame-level phonetic multi-task learning and the segment-level adversarial training, as reported in system No. 9 and No. 10, the best performance on the short-utterance task and the cross-channel task was achieved with Librispeech alignment, as shown in Table 7, obtaining EER values of 3.81% on the short-utterance task and 15.04% on the cross-channel task, and *Cavg* values of 0.0380 on the short-utterance task and 0.1517 on the cross-channel task, respectively.

The Language-Phoneme embedding extraction model was studied with (1) the usage of Language embeddings (system No. 11); (2) the usage of Language-Phoneme embeddings (system No. 12). Both of the aforementioned systems showed improvements, compared with the baseline systems, but the usage of the Language-Phoneme embeddings achieved greater improvements, and the improvement was significant on the open-set task, revealing the generalization of this approach.



(a) The loss curves.

(b) The accuracy curves.

Fig. 14. A comparison of the loss and accuracy curves between the multi-loss and single-loss training.

Table 4
Results of multiple acoustic feature learning (*Cavg*/EER%).

No.	System	Feature	Test sets		
			Short-utterance	Cross-channel	Open-set
0	AP20-OFFICIAL-BASELINE	MFCC	–	0.2696/26.94	0.0849/12.40
1	Baseline	MFCC	0.0540/5.64	0.2694/26.94	0.0851/11.73
2	Baseline	PLP	0.0538/5.80	0.2778/27.91	0.0880/13.20
3	Score-level fusion	MFCC&PLP	**0.0484/5.09**	0.2723/27.31	0.0945/10.08
4	Concat-features	MFCC&PLP	0.0528/5.54	0.2659/26.53	0.0943/13.40
5	MF learning Frame-level	MFCC&PLP	0.0541/5.51	0.2407/24.10	0.0884/12.07
6	MF learning Segment-level	MFCC&PLP	0.0529/5.62	0.2457/24.61	0.0867/12.71
7	MF learning Segment-level Multi-loss Adaptive weights	MFCC&PLP	0.0502/5.31	0.2385/24.38	0.0767/10.03
8	MF learning Frame-level CCA constraint	MFCC&PLP	0.0523/5.21	**0.2253/22.60**	0.0796/10.73
9	MF learning Segment-level CCA constraint	MFCC&PLP	0.0538/5.33	0.2374/23.83	**0.0743/9.87**

Table 5
The comparisons of different ASRs based multiple task learning (*Cavg*/EER%).

No.	System	Feature	Test sets		
			Short utterance	Cross channel	Open-set
1	Baseline	MFCC	0.0540/5.64	0.2694/26.94	0.0851/11.73
2		PLP	0.0538/5.80	0.2778/27.91	0.0880/13.20
3	Frame-MT (THCHS30)	MFCC	0.0411/4.19	0.1785/18.07	0.0730/10.40
4		PLP	**0.0392/3.95**	0.1909/19.47	0.0736/8.93
5	Frame-MT (Librispeech)	MFCC	0.0411/4.19	**0.1633/16.57**	0.0706/9.40
6		PLP	0.0405/4.15	0.1815/18.47	**0.0700/10.27**

With phoneme labels from the ASR model trained on THCHS30, which is an allowed-to-used data set in the OLR 2020 Challenge, the multi-task learning systems are further investigated on the latest OLR 2020 test sets (Li et al., 2020), listed in Table 8. The trend of the cross-channel task was similar with that in Table 6, in which the frame-level multi-task learning system obtained the best performance, with a *Cavg* value of 0.0768 and an EER value of 8.34%. In the dialect recognition task, which is also an open-set testing task, the robustness of the combination of frame-level and adversarial learning of segment-level phonetic information was revealed, with a *Cavg* value of 0.1581 and an EER value of 18.97%. In the noisy LID task, the adversarial learning of segment-level phonetic information system was the best, with a *Cavg* value of 0.0571 and an EER value of 5.56%.

6.3. Comparison and discussion

Take the cross-channel task for example. Based on the t-Stochastic Neighbor Embedding (t-SNE), the two-dimensional distribution of embeddings from the baseline system (No. 1), the best multi-feature learning system (No. 8 in Table 4), and the best multi-task learning system (No. 3 in Table 6) are shown in Fig. 15. In Fig. 15, every color represents one kind of language embeddings. From the comparison of the distributions of embeddings, it could be observed that with multi-feature learning, the embeddings of languages were more discriminative. However, with multi-task learning, the improvements were more significant on the distributions of embeddings. In other words, the embeddings from the same language got closer, and the distance between

Table 6
The results of multiple task learning with THCHS30 alignment ($Cavg$/EER%).

No.	System	Feature	Test sets		
			Short-utterance	Cross-channel	Open-set
0	AP20-OFFICIAL-BASELINE	MFCC	–	0.2696/26.94	0.0849/12.40
1	Baseline	MFCC	0.0540/5.64	0.2694/26.94	0.0851/11.73
2		PLP	0.0538/5.80	0.2778/27.91	0.0880/13.20
3	Frame-MT	MFCC	0.0411/4.19	**0.1785/18.07**	0.0730/10.40
4		PLP	**0.0392/3.95**	0.1909/19.47	0.0736/8.93
5	Segment-MT	MFCC	0.0565/5.83	0.1918/19.54	0.0813/9.73
6		PLP	0.0534/5.40	0.205/20.69	0.082/11.33
7	Segment-GRL-MT	MFCC	0.0448/4.62	0.2170/21.74	0.0776/11.00
8		PLP	0.0411/4.20	0.2051/20.69	0.0829/11.27
9	Frame-Segment-GRL-MT	MFCC	0.0407/4.12	0.2102/21.14	0.074/10.13
10		PLP	0.043/4.28	0.2058/20.69	0.0733/10.27
11	Concat-Embedding-MT (Language embeddings)	MFCC	0.0533/5.49	0.1901/19.12	0.0803/10.4
12	Concat-Embedding-MT (Language-Phoneme embeddings)	MFCC	0.0516/5.38	0.1912/19.51	**0.0634/9.20**

Table 7
Results of multiple task learning with Librispeech alignment ($Cavg$/EER%).

No.	System	Feature	Test sets		
			Short-utterance	Cross-channel	Open-set
0	AP20-OFFICIAL-BASELINE	MFCC	–	0.2696/26.94	0.0849/12.40
1	Baseline	MFCC	0.0540/5.64	0.2694/26.94	0.0851/11.73
2		PLP	0.0538/5.80	0.2778/27.91	0.0880/13.20
3	Frame-MT	MFCC	0.0411/4.19	0.1633/16.57	0.0706/9.40
4		PLP	0.0405/4.15	0.1815/18.47	**0.0700/10.27**
5	Segment-MT	MFCC	0.0426/4.42	0.2148/21.67	0.0786/9.40
6		PLP	0.0434/4.51	0.2273/22.84	0.0782/10.47
7	Segment-GRL-MT	MFCC	0.0418/4.24	0.2007/20.16	0.0763/9.00
8		PLP	0.0418/4.23	0.2173/21.76	0.0773/10.05
9	Frame-Segment-GRL-MT	MFCC	**0.0380/3.81**	**0.1517/15.04**	0.0882/9.40
10		PLP	0.0388/3.97	0.1784/18.00	0.0807/10.27
11	Concat-Embedding-MT (Language embeddings)	PLP	0.0418/4.32	0.2895/29.04	0.0728/9.47
12	Concat-Embedding-MT (Language-Phoneme embeddings)	PLP	0.0389/3.95	0.2140/21.49	**0.0728/8.80**

Table 8
Results of multiple task learning with THCHS30 alignment on OLR 2020 Challenge ($Cavg$/EER%).

No.	System	Feature	OLR 2020 test sets (AP20-OLR-test)		
			Cross-channel	Dialect	Noisy
0	AP20-OFFICIAL-BASELINE	MFCC	0.1321/14.58	0.1752/19.74	0.0715/7.14
1	Baseline	MFCC	0.1327/14.26	0.1891/22.31	0.0716/7.22
2	Frame-MT	MFCC	**0.0768/8.34**	0.1587/19.49	0.0612/6.13
3	Segment-MT	MFCC	0.1288/16.05	0.1992/23.24	0.0670/6.62
4	Segment-GRL-MT	MFCC	0.1235/14.04	0.1560/19.15	**0.0571/5.56**
5	Frame-Segment-GRL-MT	MFCC	0.1019/11.22	**0.1581/18.97**	0.0581/5.94
6	Concat-Embedding-MT (Language-Phoneme embeddings)	MFCC	0.1294/15.10	0.1741/20.29	0.0636/6.46

different languages' embeddings became larger, compared with both the best MF system and the baseline system.

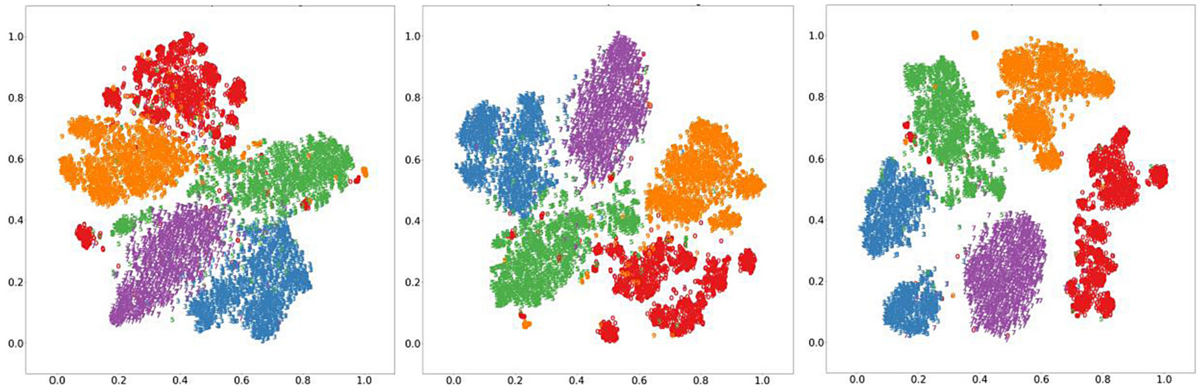
Table 9 gives the comparison of the best systems for our proposals on the OLR datasets with the top primary systems reported on the OLR18 Challenge (Short-utterance LID) (Tang et al., 2018) and the OLR19 Challenge (Cross-channel LID) (Tang et al., 2019). From the comparison, it is found that both the best MF and MT systems were comparable with the third ranked primary systems on the OLR Challenges. It should be noted that the primary systems reported on the OLR Challenges were fusion systems with many sub-systems. For the frame level multi-task learning model (frame-MT) with phoneme labels from THCHS30 based ASR model, which uses the dataset that is permitted in OLR Challenges, the performance was also comparable with the Top 1 primary systems.

From the comparison above, the effectiveness of the proposed deep joint learning strategies were revealed. However, the MT learning contributed more to the language recognition system's improvement. The possible reason behind this is that the introduction of extra phonetic information was extremely helpful for language recognition, compared with the joint learning of multi-feature extracted from the same speech.

Furthermore, we fused the best MF and the best MT systems at the score level to investigate the potential improvement and their complementarity, listed in No. 5 in Table 9. The results show that further improvements are achieved in both short-utterance and cross-channel LID.

7. Conclusion

In this paper, we propose deep joint learning methods with multi-feature and multi-task models for language recognition. The deep jointly learning strategy utilizes different kinds of language-related information to extract the language representations simultaneously. We systematically analyze the performance of multi-feature learning models and multi-task learning models



(a) The baseline system. (b) The best MF system. (c) The best MT system

Fig. 15. The two-dimensional distribution of embeddings.

Table 9
The comparison of the best systems and the top systems on the previous OLR challenges ($Cavg$/EER%).

No.	System	Previous OLR challenges' test set	
		OLR18-Short-Utterance	OLR19-Cross-Channel
0	Top 1 on OLR Challenge (Tang et al., 2018, 2019)	0.0462/4.59	0.2008/20.24
1	Top 2 on OLR Challenge (Tang et al., 2018, 2019)	0.0499/5.01	0.2713/27.69
2	Top 3 on OLR Challenge (Tang et al., 2018, 2019)	0.0512/5.19	0.2741/27.44
3	The Best MF System	0.0502/5.31	0.2253/22.60
4	The Best MT System (THCHS30)	0.0411/4.19	0.1785/18.07
5	The fusion of No. 3 and No. 4	**0.0373/3.79**	**0.1749/17.67**

for language recognition, in three kinds of test conditions, including the short-utterance condition, the cross-channel condition, and the open-set condition. The results of the test sets of the latest OLR 2020 Challenge are also compared on the multi-task learning models.

For multi-feature learning, deep joint learning with multi-loss constraints and adaptive weights for loss functions encourages the sub-networks to learn more discriminative language-related information. Additionally, deep joint learning with CCA constraints maximized the correlation of different acoustic feature representations, to learn the intrinsic language identity, and to eliminate the uncorrelated noisy representations. For multi-task learning, we presented different kinds of frameworks to study joint learning strategies for language and phonetic information. Experimental results on OLR datasets indicate that the introduction of deep joint learning improves system performance for LID, especially under the complex test conditions. Multi-feature learning systems achieved slight improvements in the short-utterance task, while they were robust in the cross-channel and open-set test conditions. All multi-task learning systems achieved significant performance enhancements compared with the baseline systems in all test conditions, and the Language-Phoneme embedding extraction structure obtained impressive improvements as well.

In future work, we will focus on improvements to the LID system under more complex test conditions, as well as more feasible optimizations in cross-channel conditions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61876160, Grant No. 62001405).

We would like to thank the organizers of the OLR Challenge for their hard work in organizing the OLR Challenge and providing useful datasets for the community.

References

Andrew, G., Arora, R., Bilmes, J., & Livescu, K. (2013). Deep canonical correlation analysis. In *The 30th Intl. Conference on Machine Learning*.

Brümmer, N. (2007). Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scorestutorial and user manual. Software available at http://sites.google.com/site/nikobrummer/focalmulticlass.

Brümmer, N., & de Villiers, E. (2013). The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF. arXiv e-prints, arXiv:1304.2865.

Cai, W., Chen, J., & Li, M. (2018). Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. http://dx.doi.org/10.21437/Odyssey.2018-11.

Castaldo, F., Colibro, D., Cumani, S., Dalmasso, E., Laface, P., & Vair, C. (2010). Loquendo-politecnico di torino system for the 2009 NIST language recognition evaluation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5002–5005).

Cipolla, R., Gal, Y., & Kendall, A. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7482–7491).

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., & Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*, 788–798.

Dehak, N., Torres-Carrasquillo, P., Reynolds, D., & Dehak, R. (2011). Language recognition via I-Vectors and dimensionality reduction. In *Interspeech* (pp. 857–860).

Hardoon, D., Mourão Miranda, J., Brammer, M., & Shawe-Taylor, J. (2007). Unsupervised analysis of fMRI data using kernel canonical correlation. *NeuroImage*, *37*, 1250–1259.

Heigold, G., Moreno, I., Bengio, S., & Shazeer, N. (2016). End-to-end text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5115–5119).

Huang, J., Li, J., Yu, D., Deng, L., & Gong, Y. (2013). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7304–7308).

Karafiat, M., Burget, L., Matejka, P., Glembek, O., & Cernocky, J. (2012). I-Vector-based discriminative adaptation for automatic speech recognition. In *IEEE Workshop on Automatic Speech Recognition & Understanding*.

Kenny, P., Gupta, V., Stafylakis, T., Ouellet, P., & Alam, J. (2014). Deep neural networks for extracting baum-welch statistics for speaker recognition. In *Odyssey*.

Kozhirbayev, Z., Yessenbayev, Z., & Karabalayeva, M. (2017). Kazakh and Russian languages identification using long short-term memory recurrent neural networks. In *IEEE 11th International Conference on Application of Information and Communication Technologies (AICT)* (pp. 1–5).

Lei, Y., Scheffer, N., Ferrer, L., & McLaren, M. (2014). A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1695–1699).

Li, Z., He, L., Zhang, W., & Liu, J. (2010). Multi-feature combination for speaker recognition. In *7th International Symposium on Chinese Spoken Language Processing* (pp. 318–321).

Li, Z., Lu, H., Zhou, J., Li, L., & Hong, Q. (2019). Speaker embedding extraction with multi-feature integration structure. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 450–454).

Li, Z., Zhao, M., Hong, Q., Li, L., Tang, Z., Wang, D., et al. (2020). AP20-OLR challenge: Three tasks and their baselines. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

Liu, Y., He, L., Liu, J., & Johnson, M. (2018). Speaker embedding extraction with phonetic information. In *Interspeech* (pp. 2247–2251).

Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., & Moreno, P. (2014). Automatic language identification using deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5337–5341).

Martinez, D., Plchot, O., Burget, L., Glembek, O., & Matejka, P. (2011). Language recognition in ivectors space.. In *Interspeech* (pp. 861–864).

Murty, K. S. R., & Yegnanarayana, B. (2006). Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE Signal Processing Letters*, *13*, 52–55.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. (2011). Multimodal deep learning. In *The 28th International Conference on Machine Learning* (pp. 689–696).

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). Automatic differentiation in PyTorch. In *NIPS-W*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

Ravanelli, M., Do, V. H., & Janin, A. (2014). TANDEM-Bottleneck feature combination using hierarchical deep neural networks. In *The 9th International Symposium on Chinese Spoken Language Processing* (pp. 113–117).

Richardson, F., Reynolds, D., & Dehak, N. (2015). Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, *22*(10), 1671–1675.

Rodriguez-Fuentes, L. J., Brümmer, N., Peñagarikano, M., Varona, A., Díez, M., & Bordel, G. (2013). The albayzin 2012 language recognition evaluation plan. In *Interspeech* (pp. 1497–1501).

Sadjadi, S., Kheyrkhah, T., Greenberg, C., Singer, E., Reynolds, D., Mason, L., et al. (2018). Performance analysis of the 2017 NIST language recognition evaluation. In *Interspeech* (pp. 1798–1802).

Sadjadi, S., Kheyrkhah, T., Tong, A., Greenberg, C., Reynolds, D., Singer, E., et al. (2018). The 2017 NIST language recognition evaluation. (pp. 82–89). http://dx.doi.org/10.21437/Odyssey.2018-12.

Saon, G., Soltau, H., Nahamoo, D., & Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *IEEE Workshop on Automatic Speech Recognition and Understanding* (pp. 55–59).

Shen, P., Lu, X., Li, S., & Kawai, H. (2017). Conditional generative adversarial nets classifier for spoken language identification. In *Interspeech* (pp. 2814–2818).

Snyder, D., Garcia-Romero, D., Mccree, A., Sell, G., & Khudanpur, S. (2018). Spoken language recognition using X-vectors. http://dx.doi.org/10.21437/Odyssey.2018-15.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-VEctors: Robust DNN embeddings for speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5329–5333).

Tang, Z., Wang, D., & Chen, Q. (2018). AP18-OLR challenge: Three tasks and their baselines. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

Tang, Z., Wang, D., Chen, Y., & Chen, Q. (2017). AP17-OLR challenge: Data, plan, and baseline. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 749–753).

Tang, Z., Wang, D., Chen, Y., Li, L., & Abel, A. (2018). Phonetic temporal neural model for language identification. *IEEE/ACM Trans. Audio Speech Lang. Process.*, *26*(1), 134–144.

Tang, Z., Wang, D., & Song, L. (2019). AP19-OLR challenge: Three tasks and their baselines. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

Tian, Y., He, L., Liu, Y., & Liu, J. (2016). Investigation of senone-based long-short term memory RNNs for spoken language recognition. In *Odyssey* (pp. 89–93).

Tong, F., Zhao, M., Zhou, J., Lu, H., Li, Z., Li, L., et al. (2021). ASV-Subtools: Open Source toolkit for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. Accepted).

Villalba, J., Chen, N., Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., et al. (2019). State-of-the-art speaker recognition for telephone and video speech: The JHU-mit submission for NIST SRE18. In *Interspeech* (pp. 1488–1492).

Vinokourov, A., Shawe-Taylor, J., & Cristianini, N. (2002). Inferring a semantic representation of text via cross-language correlation analysis. In *The 15th International Conference on Neural Information Processing Systems* (pp. 1497–1504).

Wang, D., Li, L., Tang, D., & Chen, Q. (2016). AP16-OL7: A multilingual database for oriental languages and a language recognition baseline. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 1–5).

Wang, S., Rohdin, J., Burget, L., Plchot, O., Qian, Y., Yu, K., et al. (2019). On the usage of phonetic information for text-independent speaker embedding extraction. In *Interspeech* (pp. 1148–1152).

Wang, D., & Zhang, X. (2015). THCHS-30 : A free chinese speech corpus. ArXiv, abs/1512.01882.

Yang, Y., Wang, S., Gong, X., Qian, Y., & Yu, K. (2020). Text adaptation for speaker verification with speaker-text factorized embeddings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6454–6458).

Zhao, M., Li, R., Yan, S., Li, Z., Lu, H., Xia, S., et al. (2019). Phone-aware multi-task learning and length expanding for short-duration language recognition. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 433–437).

Zhao, F., Li, H., & Zhang, X. (2019). A robust text-independent speaker verification method based on speech separation and deep speaker. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6101–6105).