A Unified Framework for Multilingual Speech Recognition in Air Traffic Control Systems

Yi Lin, Dongyue Guo, Jianwei Zhang, Zhengmao Chen, and Bo Yang

***Abstract*— This work focuses on robust speech recognition in air traffic control (ATC) by designing a novel processing paradigm to integrate multilingual speech recognition into a single framework using three cascaded modules: an acoustic model (AM), a pronunciation model (PM), and a language model (LM). The AM converts ATC speech into phoneme-based text sequences that the PM then translates into a word-based sequence, which is the ultimate goal of this research. The LM corrects both phoneme- and word-based errors in the decoding results. The AM, including the convolutional neural network (CNN) and recurrent neural network (RNN), considers the spatial and temporal dependences of the speech features and is trained by the connectionist temporal classification loss. To cope with radio transmission noise and diversity among speakers, a multiscale CNN architecture is proposed to fit the diverse data distributions and improve the performance. Phoneme-to-word translation is addressed via a proposed machine translation PM with an encoder–decoder architecture. RNN-based LMs are trained to consider the code-switching specificity of the ATC speech by building dependences with common words. We validate the proposed approach using large amounts of real Chinese and English ATC recordings and achieve a 3.95% label error rate on Chinese characters and English words, outperforming other popular approaches. The decoding efficiency is also comparable to that of the end-to-end model, and its generalizability is validated on several open corpora, making it suitable for real-time approaches to further support ATC applications, such as ATC prediction and safety checking.**

***Index Terms*— Acoustic model (AM), air traffic control (ATC), machine translation pronunciation model (PM), multiscale CNN (MCNN), multilingual, robust speech recognition.**

NOMENCLATURE

AM	Acoustic model.
AP	Average pooling.
ASR	Automatic speech recognition.
ATC	Air traffic control.
BLSTM	Bidirectional long short-term memory.
CNN	Convolutional neural network.
CTC	Connectionist temporal classification.
DNN	Deep neural network.
FC	Fully connected.
LER	Label error rate.
LM	Language model.
MCNN	Multiscale CNN.
MP	Max pooling.
MTPM	Machine translation-based pronunciation model.
OOV	Out of vocabulary.
PLM	Phoneme-based LM.
PM	Pronunciation model.
RNN	Recurrent neural network.
WLM	Word-based LM.

Manuscript received May 16, 2019; revised December 22, 2019, March 6, 2020, and July 3, 2020; accepted August 8, 2020. Date of publication August 24, 2020; date of current version August 4, 2021. This work was jointly supported by the National Science Foundation of China (NSFC) and the Civil Aviation Administration of China (CAAC), under Grant U1833115. *(Corresponding author: Bo Yang.)*

The authors are with the College of Computer Science, Sichuan University, Chengdu 610000, China (e-mail: yilin@scu.edu.cn; 2017226049015@stu.scu.edu.cn; zhangjianwei@scu.edu.cn; chengzhengmao@scu.edu.cn; boyang@scu.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3015830

I. INTRODUCTION

ASR is the process of translating spoken language into computer-readable texts [1]. An ASR provides a powerful, automatic, real-time interface for monitoring speech input. In ATC, communications exchanged between a pilot and a controller include a wealth of situational context information (i.e., real-time traffic dynamics), which is a crucial aspect when making efficient decisions regarding air traffic operations. It is believed that understanding the controlling intent embodied by ATC speech is the key to ensuring flight safety [2]. Various works have applied ASR techniques to translate ATC speeches to reduce communication errors [3], improve operational efficiency [4], and relieve controllers' workloads [5].

Our previous works [6], [7] studied flight control safety monitoring by extracting the controlling intent from real-time ATC speeches. In those works, two independent ASR models were separately trained to translate Chinese and English ATC speech. In this work, we propose a unified framework to address the speech recognition challenges specific to ATC systems (ATCSs), which are listed as follows.

1) *Volatile Background Noise and Inferior Intelligibility:* In practice, a controller is responsible for a sector and usually communicates with several pilots in that sector on the same radio frequency or channel. Therefore, the noise model in each frequency changes as the speaker changes due to equipment and radio transmission differences. Fig. 1(a)–(f) shows the spectra of several continuous ATC speech segments in the same communication frequency. Clearly, the feature intensities differ due to the different noise models. Consequently, the intelligibility of ATC speeches can be degraded by severe and volatile background noise and by other interference from the radiotelephony system.

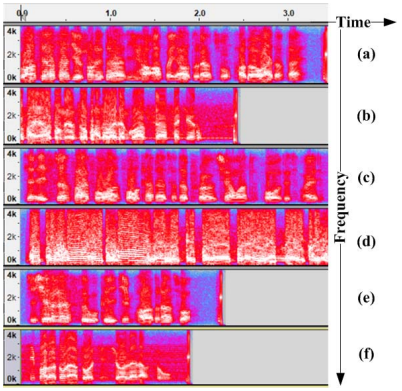


Fig. 1. (a)–(f) Spectra of several continuous ATC speeches in the same frequency.

TABLE I
SPEECH RATE OF DIFFERENT CORPORA

Language	Corpus	Mean of speech rate	Standard deviation of speech rate
Chinese	ATC speeches	5.15	1.10
	THCHS-30	3.48	0.47
English	ATC speeches	3.28	0.75
	Librispeech	2.73	0.47

All measurements are in words per second. In this work, a word indicates a Chinese character or an English word. Only the train-clean-100 dataset from Librispeech was used to calculate the given measurements.

2) *Unstable Speech Rate:* In general, the speech rate of ATC speech is higher than that in daily life due to the time constraints of the traffic situation. As shown in Table I, the speech rates of the Chinese and English speech in our corpus average of 5.15 and 3.28 words/s, respectively. For comparison, the speech rates of the open corpora THCHS-30 [8] and Librispeech (train-clean-100) [9] average only approximately 3.48 and 2.73 words/s, respectively. However, the speech rate is also affected by real-time conditions and can show huge differences depending on the speaker and the flight density. In our corpus, the standard deviations of the speech rates are 1.10 and 0.75 words/s for Chinese and English, respectively, whereas the equivalent measurements for both the open corpora are only approximately 0.47 words, as shown in Table I.
3) *Multilingual ASR:* In general, English is the universal language in ATC communication according to the International Civil Aviation Organization (ICAO) requirements. However, in practice, domestic pilots typically speak with controllers in local languages. Therefore, speeches on the same frequency may include speech in multiple languages. More specifically, based on the controlling rules, multiple languages may occur during the same speech segment, which is the most exceptional characteristic of the ASR task in the ATC domain. For example, a runway number "A2" might be pronounced as "alpha " during ATC communication.
4) *Code Switching:* The ICAO published the criteria for different communication procedures and word pronunciations [10]. Only the standard terminologies are allowed to be used in ATC communication. Code switching was proposed to eliminate speech misunderstandings caused by homonyms or near homonyms, such as "nine-niner." In addition, some words in ATC communication are pronounced differently than in daily life, which makes ATC communications sound like a special dialect for ATC only. Meanwhile, the standard terminology constrains the vocabulary of the ASR task to certain words that form a subset of the common vocabulary.
5) *Vocabulary Imbalance:* Despite the standard terminology restrictions, in practice, OOV words occur in ATC speech because speakers do not strictly comply with the terminology rules. Moreover, the frequencies of different words in our corpus are extremely unbalanced (causing a sample sparsity problem). According to our data set analysis, approximately 40% of the words in the vocabulary appear fewer than ten times (such as modal words), whereas other words may appear up to a million times (such as spoken digits). From a model training perspective, this situation results in an imbalanced data set that severely degrades the classification accuracy between speech frames and text labels.

Due to the mentioned specific conditions, the existing ASR approaches are unsuitable for the ATC domain, which prompted us to study a dedicated approach to this issue. In this article, we propose a deep-learning-based approach and design a special paradigm that unifies multilingual ASR into a single framework. Unlike popular end-to-end ASR models, we propose a cascaded pipeline with an AM, an LM, and a PM. The AM is a typical ASR model that translates ATC speech into a phoneme-based text sequence. The PM converts these phoneme-based text sequences into human-readable text (words). The LM is designed to improve the decoding accuracy; it applies phoneme-based (PLM) and word-based (WLM) methods to the AM and PM, respectively. The LM's decoding process captures the contextual meanings among different words (code-switching words) to improve the ASR performance and cope with the previously mentioned code-switching issue in ATC speech.

Learning from state-of-the-art ASR models, we design an improved RNN- and CTC-based model for the AM, in which a 2-D CNN-based multiscale feature modeling method is proposed to extract high-level acoustic features from both the temporal and frequency dimensions of the speech feature. MCNN kernels are proposed to capture discriminative and robust features from speeches with different noise models (the frequency dimension) and speech rates (the time/frame dimension). In addition, we also propose using AP to filter the intensive noise to reduce its overlapping disturbance on human speech; this can relieve the impact of the instantaneous pulses in the radio frequency. A BLSTM block is applied to build sequential correlations among different speech frames, and an FC layer is designed to map the temporal features of the BLSTM layer to the vocabulary. This layer predicts the probability of a certain word given the framewise speech features. Finally, the differences between the predicted labels and true labels are evaluated by the CTC loss function and

backpropagated to the other neural layers to optimize their trainable parameters.

By analyzing the task of the PM (i.e., converting the phoneme-based text into word-based text), we propose an MTPM for this module. It is believed that both phonemes and words are representations of spoken language that can use different basic units, such as Chinese and English. An encoder–decoder architecture with an attention mechanism is proposed to achieve text conversion. Furthermore, the fault tolerance of the PM is improved by the LM correction, which improves the overall robustness of the proposed framework. Experimental results on large amounts of real Chinese and English ATC recordings show that the proposed framework can achieve the ASR task with high performance and a low 3.95% LER. We confirm the efficiency and effectiveness of the proposed modules through the designed experiments. In summary, our original contributions in this work are as follows.

1) A deep-learning-based integrated framework is proposed to translate ATC speech into text; this framework unifies the multilingual ASR task (Chinese and English in this work) into a single framework. The framework includes the AM, PM, and LM (PLM and WLM) models. By fixing the AM vocabulary, the reusability of the proposed framework for recognizing specified languages is highly improved by applying the incremental learning technique. In addition, the sample sparsity in the word vocabulary is mitigated by designing phoneme-based labels.

2) A multilingual speech corpus is built to support ASR research in ATC by collecting raw speeches from real ATCSs.

3) An improved CTC-based AM is proposed to convert the ATC speech into a phoneme-based text sequence; the AM considers both the spatial and temporal dependences of the speech feature. An MCNN/AP architecture is designed to address two specific problems that occur in ATC speech, i.e., volatile background noise and unstable speech rate.

4) We process the PM as a machine translation task by analyzing its intrinsic characteristics. An attention-based encoder–decoder architecture is proposed to achieve this goal.

5) Two-level LMs are designed to support the decoding of the AM and PM, which also benefit the code-switching problem specific to the ATC domain.

The remainder of this article is organized as follows. In Section II, we introduce works related to our research. The proposed framework is sketched in Section III. The details of the proposed framework are introduced in Section IV. Section V describes the experimental configurations, and the results of the experiments are reported and discussed in Section VI. Finally, conclusions and future works are given in Section VII.

II. Related Works

The ASR technique is applied to translate spoken language into computer-readable input, such as text or binary coding. Developments in computer science and signal processing have resulted in ASR research adopting a variety of technical improvements. In early works, special devices were developed to translate isolated words for a specified person [11]. In 1959, a phoneme recognizer was built to match four specified vowels and nine consonants based on a pattern similarity recognition principle [12]; this was the first study to apply statistical information to an ASR system. Lately, dynamic programming [13] was proposed to address the nonuniform temporal scale of speeches, which greatly improved the ASR performance. Linear predictive coding (LPC) was introduced to simplify the estimation of the vocal tract [14] and has also been applied to pattern recognition [15] in the ASR task. During this period, the n-gram LM [16] was proposed to improve the decoding accuracy, and it rapidly became an indispensable part of ASR systems. With the rapid development of statistical models, the hidden Markov model (HMM) pushed ASR research into a new stage, resulting in HMM-based frameworks [17]. Since then, many improved HMM-based algorithms have been proposed that showed promising performances on the ASR task [18]–[22]. In the HMM framework, a Gaussian mixture model (GMM) was proposed to fit the data distribution between speech frames and text labels. Contemporaneously, the artificial neural network (ANN)-based method was also studied to address the ASR task [23]; now, such methods are fundamental components of the ASR model. With the development of pattern recognition algorithms, support vector machine (SVM)-based methods were promoted that minimized the empirical recognition error in ASR systems [24], [25]. Subsequently, ASR production based on the abovementioned theories, such as HTK [26], expanded greatly.

The training efficiency of neural network models has been improved by advances in computer hardware, which also enhanced their applicability for ASR tasks [27]. A DNN was applied to build the data distribution between frames and labels and formulate the HMM/DNN framework [28]–[30]. Experimental results have shown that HMM/DNN-based approaches achieve better performances than do HMM-based approaches. Subsequently, the CNN was applied to mine spatial correlations from speech features [31]–[33]. Considering the temporal characteristics of speech, the RNN was naturally proposed to solve the temporal modeling issue [34]. The long short-term memory (LSTM) block was further applied to cope with the gradient vanishing problem of the RNN block during model training [35], [36]. Some works applied combined CNN and RNN networks to capitalize on all their advantages [37], [38]; these combined models yielded even higher recognition accuracies than did the HMM and DNN approaches. To map variable-length speech frames to variable-length output labels automatically, Graves and Jaitly [39] proposed the CTC loss function, which is now a standard paradigm in end-to-end ASR models. The end-to-end ASR approaches not only outperformed other existing approaches but also provided an intuitive pipeline from speech features to word labels directly [40], [41]. Recently, an encoder–decoder-based sequence-to-sequence model was also studied to translate spoken language and some preliminary results have been shown [42], [43]. The LAS model combined a sequence-to-sequence model with the attention mechanism and designed a "listen, attention, and spell" processing paradigm to complete the ASR task [44]. The "cocktail party" problem is a popular

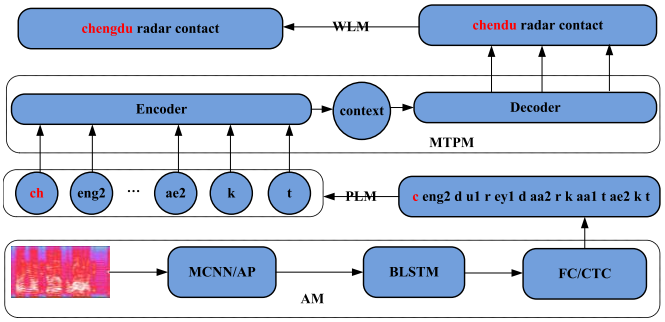


Fig. 2. Architecture of the proposed ASR framework.

topic [45], [46] in ASR research that also involves overlapped speech segmentation and recognition. However, the push-to-talk (PTT) signal of ATC equipment ensures that different speakers' speeches within the same communication channel are mutually exclusive. Therefore, the "cocktail party" problem does not need to be considered in this work.

The specificities and challenges of speech recognition in the ATC domain were reviewed in [47]. Kopald *et al.* [48] applied ASR to a safety monitoring task in ATC. MITER Corporation proposed a system to detect closed runway violations in airports [49]. An ASR tool was developed to support the management of flight arrivals (flight landing) [50]. Recently, the Airbus company held a challenge for speech recognition in ATC, and over 20 teams submitted their results [51]. The ASR interface for air traffic management was studied in [4]. ASR has also been used to reduce the controllers' workloads by checking flight information [5]. A context-dependent speech recognition and understanding system was developed to support decision-making by considering real-time flight information [52]. Semisupervised training of a DNN-based AM was proposed to address the scarcity of training samples in ATC [53], and a knowledge extraction algorithm was also applied to improve the performance of speech recognition [54].

III. Proposed Framework

Unlike the state-of-the-art end-to-end ASR models, we propose a cascaded framework that translates ATC speech based on task specificities. A novel processing paradigm is designed with deep-learning models that proceed from speech frames to phoneme texts and then to word texts, corresponding to the cascaded modules AM and PM, respectively. By designing this pipeline, the multilingual ASR task can be achieved within one framework. The proposed paradigm is shown in Fig. 2. The end result is that the ATC speech is converted into computer-readable text through the AM, PLM, MTPM, and WLM models. The PLM and WLM are designed to support ASR decoding and address the code-switching words, as shown in the red text in Fig. 2.

In our proposal, the AM is processed as a typical ASR task, using the MCNN/BLSTM/CTC architecture designed to translate the ATC speech into a phoneme-based text sequence by outputting basic phoneme units. To cope with the volatile background noise and unstable speech rate, we design an improved architecture called MCNN, which performs MCNN-based feature modeling. An AP layer is applied to filter the background noise and extract the high-level acoustic features. To address the PM task in this work, we propose an encoder–decoder architecture (MTPM) that translates phoneme texts to word texts. An attention mechanism is applied to mine the long-term dependences in the phoneme sequences.

A cascaded framework is proposed by considering the advantages of the CNN/BLSTM/CTC and encoder/decoder architecture for processing the specific tasks of the ASR in ATCSs. Most importantly, the following ASR task concerns in ATC can also be considered and addressed in a proper manner, enabling the framework to be applied in practice.

A. Modeling Unit

As discussed previously, the multilingual ASR is a key issue addressed by this work. Typically, Chinese characters and English words form the core semantics for human understanding; thus, these representations should serve as the modeling unit in an end-to-end ASR model. However, in this case, the vocabulary is a large and unfixed set as the training corpus increases. The OOV words degrade the ASR performance, which further affects the applicability and robustness of the ASR interface in real-time situations. The phoneme-based label design can ameliorate this issue to some extent. The final solution to the OOV issue is tackled by the PM in this work. For a given language, the phoneme units can be considered as a fixed set; this fact allows us to ignore changes to the AM vocabulary and instead apply incremental learning to upgrade its model parameters. When facing a new training corpus, the optimized model parameters can be trained incrementally rather than having to modify the network architecture to train a new AM. This design greatly improves the reusability of the model in real applications.

B. Training and Decoding Efficiency

In general, the architecture of the AM is much deeper than that of the PM (10^7 versus 10^5 trainable parameters), which means that more training time and computational resources are required to optimize its parameters. Therefore, we process the OOV issue in the PM phase to save computational resources. By modifying the architecture of a shallow neural network (PM), the OOV problem can be considered and solved with less resource consumption. Meanwhile, adopting phonemes as the AM output unit further reduces the computational complexity and improves the decoding efficiency during the inference process. There are 1472 Chinese characters and 1299 English words in our corpus; however, these are represented by only 290 phoneme units. This tenfold difference in output labels would impose an unnecessary burden on model training and decoding.

C. Label Granularity and Balance

Semantically, Chinese phrases and English words are the primary units in spoken language; however, they clearly do not have identical scales from a vocalization perspective.

Fortunately, in this work, the phoneme-based label unifies spoken language vocalization, which is especially important to ASR modeling. Moreover, word imbalances in the vocabulary are greatly alleviated by using phoneme labels (which are subconcepts of words). In this work, transferring vocalization from words (2771) into phonemes (290) also reduces the imbalance among output labels, which improves the cohesiveness of the data samples and provides further benefits for convergence during model training.

Considering the advantages of the proposed modules and the specific challenges of the ASR task in ATC, we believe that a cascaded framework is a preferable solution for the ASR task in this work. The proposed cascaded framework has the advantages of both AM reusability and PM extendibility, which ensures that the framework is capable of dealing with the new vocabulary words that will appear in a large training corpus. This approach also addresses the label (word) imbalance and improves the overall performance. In addition, a small and fixed AM output vocabulary improves the decoding efficiency to meet the requirements of real-time applications.

IV. Methodologies

A. AM

The AM is the foundation of the proposed framework because it converts speech signals into computer-readable text. The input and output of the AM are the 39-D mel frequency cepstral coefficients (MFCCs) and a sequence of phoneme labels, respectively. For each frame, the AM predicts the probability of a given label given the frame features, i.e., $p(l_i|f_t)$, where f_t denotes the features of the tth frame. Here, $l_t \in A$ is the phoneme unit, where A represents the phoneme vocabulary (there are approximately 290 phonemes for the Chinese and English speech in this work).

Based on the ASR task, an improved CTC-based deep-learning model is proposed to address the special challenges of volatile background noise and unstable speech rate in ATC. Specifically, we propose a new architecture called MCNN to solve these issues, i.e., the 2-D CNN-based multiscale acoustic feature modeling method. The 2-D CNN builds spatial correlations from speech features using both the temporal and frequency dimensions. It is multiscale in that multiple CNN kernel configurations serve as the feature extractor to deal with different temporal and frequency distributions of the ATC speech. The different convolutional kernels learn acoustic features at different scales by using different receptive fields on the speech features, which improves the robustness of the framework when faced with volatile background noise. Similarly, the temporal characteristics of speeches spoken at unstable speech rates can be captured by treating different frames as the same phoneme label. The multiple convolutional kernels learn different noise distribution patterns from raw speeches, and the truth label finally confirms the configurations for different noise models by optimizing their weights with the backpropagated CTC loss.

In general, universal ASR models apply MP to extract salient features from speech features. In this work, we propose using the AP operation to accomplish this task instead of analyzing the noise distribution in ATC speeches. As shown in Fig. 1, the noise intensity is generally higher than that of human speech and is distributed in a dispersed fashion on the temporal and frequency dimensions. The AP operation discards the high intensities (noise) to alleviate noise overlapping with human speech, which provides more discriminative features for the subsequent sequential modeling. Furthermore, it is also helpful to compress the data dimension. The MCNN/AP scheme is shown in Fig. 3. There are two MCNN layers and an AP layer in the network. The first MCNN focuses on building multiple resolutions on the frequency dimension, whereas the second one is designed to model the correlations on the temporal dimension.

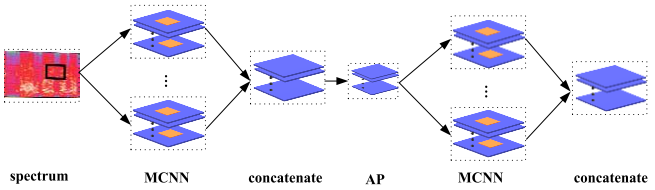


Fig. 3. MCNN/AP scheme in the AM.

Following the MCNN/AP architecture, several BLSTM layers mine the temporal dependences among the speech frames [35] by considering their temporal characteristics. Because this is a typical sequential classification task, a bidirectional LSTM is applied to improve the classification accuracy by considering the forward and backward patterns of different phoneme combinations [39]. Equations (1) and (2) show the BLSTM inference rules

$$\vec{h}_t = \Gamma(f_t, \vec{h}_{t-1}, \vec{b}), \overleftarrow{h}_t = \Gamma(f_t, \overleftarrow{h}_{t+1}, \overleftarrow{b}) \tag{1}$$

$$l_t = W_{\vec{h}l}\vec{h}_t + W_{\overleftarrow{h}l}\overleftarrow{h}_t + b \tag{2}$$

where h and b are the hidden unit and bias, respectively. The notations $\rightarrow$ and $\leftarrow$ denote the inference chains from the forward and backward directions, respectively, Γ denotes the rule of the LSTM cell, and $W_{\cdot}$ denotes the vectorized trainable weights. The CTC is applied to evaluate the differences between the true labels and predicted labels and further support the parameter optimization [39]. Let the input speech be $F = \langle f_1, \ldots, f_T \rangle$ and y_k^t denote the probability that the tth frame corresponds to the output label k. For a certain input speech, the probability of any output sequence π is shown in (3). Therefore, the probability of the final sequence can be obtained by (4), in which Ξ is the set of all possible sequences. For example, by using '_' to denote a blank, both the outputs "a_bb_c" and "_ab_c_" correspond to the final output "abc"

$$p(\pi|F) = \prod_{t=1}^{T} y_{\pi_t}^t, \quad \pi_t \in A \tag{3}$$

$$p(l|F) = \sum_{\pi \in \Xi^{-1}(l)} p(\pi|F). \tag{4}$$

B. PM

The PM is proposed for converting the phoneme-based text sequence generated by the AM into a word-based sequence,

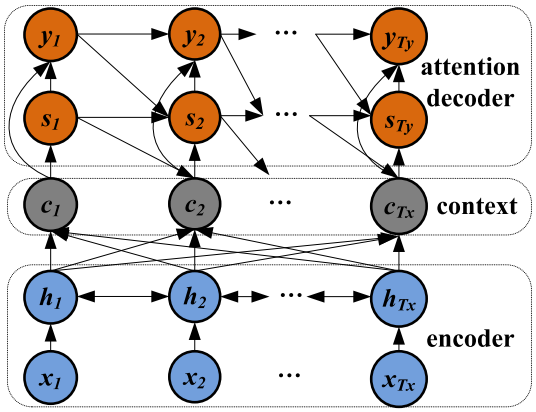


Fig. 4. Architecture of the MTPM.

which is the representation that corresponds to human understanding. In this work, based on the fact that both phonemes and words are representations of spoken speech, we propose a machine translation model to achieve this goal. By learning state-of-the-art translation models, an encoder–decoder network is proposed to formulate an MTPM. The architecture of the PM is shown in Fig. 4; it maps a variable-length phoneme text sequence to a variable-length word text sequence. Finally, the translation from speech to text (Chinese character and English word) is achieved by the designed AM/PM paradigm.

The encoder extracts contextual meanings of the input sequence in the source language (phoneme) and generates a context vector that fully represents the bidirectionally exchanged semantic context of the input information to improve the modeling accuracy. An attention mechanism is proposed to assign different weights to different contextual components that have the ability to mine long-term dependences from the context vector. During the encoding procedure, the context vector is decoded as an output sequence in the target language (words). The hidden state of the encoder is computed by (5), in which $f_e(\cdot)$ denotes the RNN inference. Considering the attention context weights, the decoder predicts the output words based on both the input at the current step and its historical outputs, as shown in (6)–(10), where $W_\cdot$, v_a^T, and U are trained weights [55], and σ and ξ are the softmax and tanh activation functions, respectively

$$h_t = f_e(x_t, h_{t-1}, h_{t+1}) \tag{5}$$

$$p(y_t|y_1, \ldots, y_{t-1}, x_t) = \sigma(W_s \tilde{s}_t) \tag{6}$$

$$\tilde{s}_t = \xi[W_c(c_t, s_t)] \tag{7}$$

$$s_t = f_d(y_{t-1}, s_{t-1}) \tag{8}$$

$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j, \quad \alpha_{ij} = \sigma(e_{ij}) \tag{9}$$

$$e_{tj} = a(s_{t-1}, h_j) = v_a^T \xi(W s_{t-1} + U h_j). \tag{10}$$

C. LM

In the proposed pipeline, the AM/PM modules accomplish the required ASR task. However, framewise greedy decoding with maximal probability may not be an optimal selection for the entire input speech. Thus, the LM is proposed to improve the decoding accuracy based on the contextual meaning and to distinguish among words and phrases with similar sounds. The LM predicts the probability of a given sentence in a certain semantic application, as shown in (11). Based on this contextual meaning, the LM can be used to recover from typical spelling errors by predicting the probability of the next vocabulary given its historical input words, i.e., $p(l_{t+1} = A_j | l_1 \cdots l_t)$

$$p(L) = p(l_1, \ldots, l_T) = p(l_1) p(l_2|l_1) \cdots p(l_T|l_1, \ldots, l_{T-1}). \tag{11}$$

The statistical LM [56] is usually simplified by the Markov property to reduce the large search space, i.e., the n-gram one. Consequently, a neural network was proposed to build contextual dependencies by using its powerful ability to model nonlinear features [57], i.e., an NNLM. Naturally, an RNN [58] is the typical architecture for an NNLM, which considers the sequential correlations among different vocabulary words. Combined with the LM, some prediction errors or ambiguities of the AM or the PM can be solved.

In the inference phase, the final transcription Y is obtained by maximizing an integrated evaluation measurement of the LM and AM/PM. Taking AM decoding as an example, the combination method is as follows:

$$Q(Y) = \log(p_{\text{am}}(Y|X)) + \alpha \log(p_{\text{lm}}(Y)) + \beta \text{len}(Y) \tag{12}$$

where $p_{\text{am}}(Y|X)$ and $p_{\text{lm}}(Y)$ denote the probabilities predicted by the AM and LM, respectively. The hyperparameter α controls the relative contributions of the AM (set to 1) and the LM. The last terms, β and $\text{len}(Y)$, form a penalty factor of the transcription length. The beam search strategy [59] is applied to obtain the optimal transcription.

In this work, the dependences between code-switching words (which are special terminology in ATC) and common words can also be captured by the LM. For instance, an airline name (special terminology) is usually followed by an aircraft identification sequence (digits that include both common words and code-switching words). Almost all the digits in ATC speeches are switched to other words to eliminate misunderstanding due to unclear pronunciation. This fact allows the LM to improve the decoding accuracy with high confidence from the perspective of contextual meaning.

V. Experimental Configurations

A. Data Description

To support ASR research in ATC, we built a training corpus in which real raw speeches were collected from several civil airports in China, including Chengdu, Shanghai, and Kunming. A community version of our training corpus is introduced in detail in [60], and it can be applied to noncommercial research as a preliminary corpus. The real operating data (the raw speeches) were saved as one file per hour continuously: silence and background noise were also recorded. Based on the PTT signal, the raw speech is first split into segments to simulate a real controlling scene and remove silence and

TABLE II
DATA DESCRIPTIONS

Data amount (Hours)			
	Chinese	English	Summary
Train	1022	253	1275
Dev	11	3	14
Test	115	25	140
Vocabulary of the AM			
	Chinese	English	
Standard	Pinyin	CMU Pronouncing Dictionary	
Size	204	79	
Tools	unicode_to_ pinyin*	Cmudict**	
Example	两 (l iang3)	echo (EH1 K OW0)	
Transcription			
Phoneme	EH1 K OW0 EH1 K OW0 b a1 N OW0 V EH1 M B ER0 CH AA1 R L IY0 AW1 L F AH0 l iang3 q ian2 d eng3 g uo2 h ang2 s iy4 s iy4 uu u3 l iang3		
Word	echo echo 八 november charlie alpha 两 前 等 国 航 四 四 五 两		

*https://github.com/sculyi/Deep-Learning/blob/master/pylibs/resource/chn_py/unicode_to_hanyu_pinyin.txt.
** An open source python package, https://pypi.org/project/cmudict/0.2.0/, English letters are also designed for the special ATC-related words.

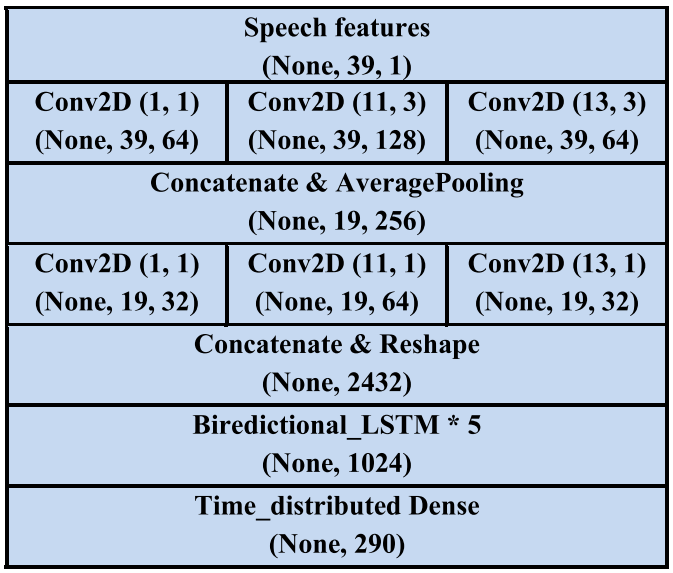


Fig. 5. Architecture of the AM in the proposed framework.

noise. All the speech segments were transcribed manually by humans; an example transcription is shown in Table II. There are 290 labels in the AM vocabulary: 204 for Chinese, 79 for English, 2 for special words ("<UNK>" and "<SPACE>"), and 5 reserved labels for future use. Basically, the manual annotations of the transcriptions are word labels (Chinese character and English word). We used the tools listed in the table to generate the phoneme labels. To avoid repetition, the Chinese and English phonemes are notated in the lower and upper cases, respectively. Currently, a total of 1148 h of Chinese speech and 281 h of English speech have been extracted and transcribed from approximately 3600 h of raw speech. The duration of each training sample ranges from 2 to 10 s based on their different controlling instructions. The data statistics are summarized in Table II. During the model training, approximately 25% of the training samples were randomly selected to improve the data size and diversity by adjusting the speech rate and denoising (i.e., data augmentation).

The same data set divisions were used to train, validate, and test the AM, PM, and LM to ensure the fairness of the experiments, and the augmented samples used for the different ASR methods were also the same. The input and output of the PM are the phoneme text and word text of the transcription, respectively. The LMs were trained with the speech transcriptions on the training data set, and the PLM and WLM were trained with the phoneme and word transcriptions, respectively.

After preparing the training samples, we applied the perceptual evaluation of speech quality (PESQ) [61] to evaluate the speech quality. The PESQ yields a score between 1 (poor) and 4.5 (good) based on a reference speech: 3.8 is the acceptable score for a telephonic voice. We randomly selected 100 speech segments from the open THCHS30 and Librispeech corpus as reference speeches and found that the average PESQs for the Chinese and English training samples used in this work are 3.359 and 3.441, respectively (i.e., all the samples are below the acceptable score). The final accuracy is calculated to validate the generalizability of the proposed approach for translating ATC speeches with different PESQ scores.

The LER, i.e., the word error rate based on the Chinese characters and English words, is applied to evaluate the performance of the proposed ASR framework. Meanwhile, the real-time factor (RTF) is applied to evaluate the decoding efficiency, as shown in (13). Here, T_s is the time required for a speech with a duration of T_d; both two measurements are in the same units, either seconds or minutes. The HMM-based approaches are not measured by the RTF because they are not in the same framework as RNN/CTC or our proposed approach

$$\text{RTF} = \frac{T_d}{T_s}. \tag{13}$$

B. Network Architecture

The network architecture of the AM and its output shape is shown in Fig. 5. In the listed layer shapes, the "None" denotes the variable-length temporal dimension. In the proposed MCNN, the padding mode is set to "same" to maintain the output dimension and further support the concatenation of different CNN outputs. The (1, 1) filter is applied to normalize the acoustic features in the local receptive field and compress the data dimension. The first MCNN layer focuses primarily on mining the frequential dependences through the (11, 3) and (13, 3) kernels, whereas the second layer is used to extract high-level features in the temporal dimension through the (11, 1) and (13, 1) kernel configurations. The BLSTM layers are designed to build temporal correlations among the speech features, and each hidden BLSTM layer has the same number of neurons. Finally, the prediction layer is designed to map the extracted representations to the output unit, i.e., phoneme vocabulary. Following each CNN and BLSTM layer, the batch normalization and dropout layers are designed to speed up the training process and prevent overfitting, respectively. The rectified linear unit (ReLU) is the activation function for the CNN

layers, and the softmax layer normalizes the probabilities with which a frame corresponds to the vocabulary words in the prediction layer.

The proposed PM includes two LSTM layers in the encoder with 128 nodes per layer. The embedding size of the encoder is 80. The decoder consists of two BLSTM layers with 256 nodes. In this work, the PLM and WLM are implemented by referring to [62]. The input length of the PLM is 12 phonemes, and two BLSTM layers with 80 nodes are designed to build the context dependences. The network configuration for the WLM is as follows; the input length is five words, and there are two BLSTM layers with 128 nodes each.

C. Model Training

The deep-learning models in this work were constructed using the open framework Keras 2.0.4 and executed on a TensorFlow 1.0 backend. The training server was equipped with 2 Intel Core i7-6800K processors, 2 NVIDIA GeForce GTX 1080Ti GPUs, 64-GB memory, and an Ubuntu 16.04 operating system.

During AM training, the Adam optimizer was used to optimize the model weight parameters. The initial learning rate was set to 10^{-4} and halved when the validation accuracy did not decrease after ten consecutive tests (500 iterations per test). The batch size was set to 160 when training in parallel on the two GPUs. To reduce the loss to a certain level as soon as possible, all the training samples were sorted by their durations during the first training epoch. In ATC speeches, similar training sample durations may indicate that their texts belong to the same controlling instruction and share a higher similarity. All the training samples were shuffled after the first training epoch to improve model robustness. An early stopping strategy based on the validation loss was applied to check the training progress.

During PM training, Adam was applied with a 10^{-5} initial learning rate to optimize the weight parameters. Gradient clipping with a threshold set to 5.0 was applied to prevent the gradient explosion problem. The batch size was 64. The PLM and WLM training operations used the same configuration as the PM.

VI. Results and Discussion

A. AM Tests

Basically, the AM draws on the advantages of the RNN/CTC-based end-to-end ASR model, but the CNN architecture is improved to address certain ATC-specific conditions. In this experiment, we first tested the AM performance with different layer configurations, including the configurations of both the CNN and BLSTM. By referring to various popular end-to-end ASR models, we applied the following CNN configurations to validate the proposed MCNN architecture by evaluating the final LER of the AM.

1) *1-D CNN:* One CNN layer with 11@1000 kernels.

2) *2-D CNN:* By referring to [37], this model uses two CNN layers with filter configurations of 9∗9 at 256 and 4∗3 at 256.

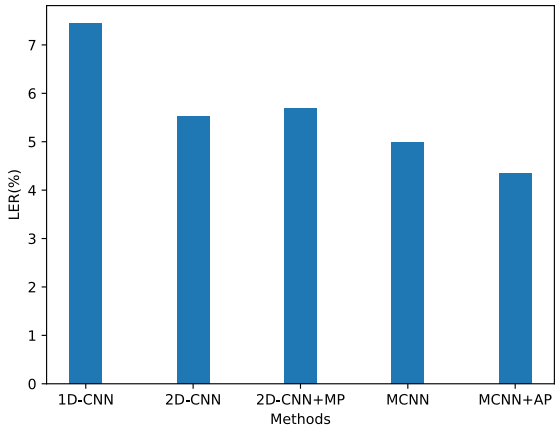


Fig. 6. LER of the AM with different CNN configurations.

3) *2-D CNN+MP (MP):* Based on 2), an MP layer with a 2∗2 receptive field follows the first CNN layer.

4) *MCNN and MCNN+AP:* The proposed MCNN with the configurations discussed earlier.

In this group experiment, we focus on demonstrating the validity of the proposed MCNN and AP architecture. Therefore, we used the same BLSTM configuration for all listed methods, in which 5 BLSTM layers with 512 nodes were applied to build the sequential information of the speech frames. Combining the CNN and BLSTM layers with the CTC loss, the models in 1)–3) were end-to-end ASR models, which also serve as the baseline models in this section. All the training samples (with augmentation) were used to optimize the model parameters, and the test samples were decoded without the assistance of the LM.

The results of these different configurations are shown in Fig. 6. As the experimental results show, the proposed MCNN/AP architecture achieves a better performance with the same BLSTM layer configuration compared with the other CNN architectures. In general, a 2-D CNN is a better option for the AM task in this work because the 1-D CNN suffers from the largest LER (7.44%). We can attribute these experimental results to the volatile background noise in ATC speech. The MCNN can extract the high-level acoustic features at different scales using different CNN kernel designs. It is precise because of the different scales on temporal (speech rate) and frequency (background noise) dimensions that the proposed architecture can be adapted to more complex environments. Due to its extensive applicability, the proposed architecture is expected to improve the robustness and generalizability of the ASR model for dealing with outlier data features.

In addition, comparing 2) with 3) and the two experiments in 4), the MP degrades the ASR performance and results in additional LER (by 0.17%), whereas the AP improves the ASR performance and reduces the LER (by 0.65%). Therefore, the experimental results show the validity of the proposed MCNN and AP architecture. In this research, due to the high intensity and wide distribution of the background noise, the AP operation is preferable to the MP for filtering the noise that overlaps human speech. In contrast, the MP operation takes

TABLE III
AM PERFORMANCE WITH DIFFERENT BLSTM CONFIGURATIONS

No.	#LSTM layer	#Neuron	AM LER (%)
1	3 BLSTM	512	6.28
2	5 BLSTM	256	5.51
3		512	4.34
4	7 BLSTM	256	4.73
5		512	**4.20***
6	5 LSTM	1024	4.97

TABLE IV
PERFORMANCE OF THE PM AND LM

No.	AM	PLM	PM	WLM
1	-	-	0.13	0.12
2	4.34	-	4.82	4.65
3	4.34	3.88	4.09	**3.95***

All measurements in the table is LER (%).

the highest candidate from the local field—which is probably the noise in the target of this work (ATC speeches).

Following the CNN experiments, to determine an optimal BLSTM architecture, we further tested the AM LER on different BLSTM layer configurations with the proposed MCNN+AP architecture. The experimental results are listed in Table III. Similar to the CNN experiment, all the training samples were used to optimize the model parameters, and the test samples were decoded without assistance from the LM. The results show that using more BLSTM layers with more neurons benefits the AM task in this work. The performance improves substantially, from 6.28% to 4.20% with respect to the different numbers of BLSTM layers and neurons. In addition, increasing the BLSTM layer improves the performance more than does increasing the neurons in each hidden layer, which is also called "going deeper." Meanwhile, we also designed experiment 6 to confirm the validity of the BLSTM architecture for dealing with the bidirectional context of speech frames. Compared with experiment 3, the BLSTM layer obtains a better LER even when the number of neurons is the same. As a sequential classification task, the BLSTM builds the temporal dependences of speech frames from both historical and future perspectives to improve the AM's performance.

In summary, several RNN/CTC-based end-to-end ASR models with different CNN kernels and pooling operations were designed to validate both the proposed MCNN/AP and the entire AM network. The BLSTM was also considered in the designed experiments. The experimental results confirm the effectiveness of the proposed architecture on the ASR task in this work.

B. PM and LM Tests

In this study, the PM converts the phoneme-based text sequence into a humanly readable word-based text sequence. Because an encoder–decoder is the standard architecture used for the machine translation task, we did not design a baseline model for the PM experiments. Instead, we focused on testing the PM accuracy under different inputs regarding the manual transcription, AM prediction, and AM prediction with LM correction. In addition, the LM correction for PM decoding is also considered in this section. The experimental results are reported in Table IV. Experiment 1 shows the results on all the test samples in the data set, i.e., all the PM input sequences are correct. It is believed that the proposed PM is able to perform the text conversion from phoneme to word with high accuracy; it achieves an LER of only 0.13% (which can be reduced to 0.12% with WLM correction). The results of experiments 2 and 3 were obtained by inputting the AM prediction results into the other modules (PM and LM), which means that spelling errors may exist in their input sequences. The results show that both the PLM and WLM improve the decoding accuracy. In addition, the PLM obtains greater improvement than does the WLM, approximately 0.55% versus 0.15%, respectively. Compared with the word vocabulary, the phoneme unit is more basic and higher dependences can be found among their labels. When given an input sequence containing errors, the PM causes additional accuracy degradation; however, the errors can be corrected by the proposed LM to some extent. Finally, the proposed cascaded processing (AM/PLM/PM/PLM) improves the LER from 4.34% to 3.95%, which is the final accuracy of our proposed ASR framework.

C. ASR Tests

In this section, we report the performance of different ASR methods to further demonstrate the effectiveness of the proposed framework. The HMM/GMM-, HMM/DNN-, CNN/CTC-, and RNN/CTC-based methods served as the baseline models. The HMM/GMM and HMM/DNN methods were implemented based on the Kaldi framework, specifically, THCHS30. We applied only the implementation details in the default scripts. The RNN/CTC-based method was implemented based on DeepSpeech2 [59], which outputs Chinese characters and English words directly, for a total of 2771 words. The network architecture of the RNN/CTC-based method is summarized as follows: one CNN layer with 11 at 1000 kernels, seven BLSTM layers with 512 nodes in each direction, and one FC layer. The training hyperparameters were the same as those of DeepSpeech2. The CNN/CTC-based baseline was implemented based on the Jasper framework [63], whose output vocabulary is the same as that of DeepSpeech2. The architecture of the CNN/CTC baseline was designed following Jasper ($10*3$). The experimental results are listed in Table V. The RNN-based LMs used in this experiment were the same as those in the WLM. The experimental results show that the proposed ASR method yields the best performance among all the tested methods in this work with an LER of only 3.95%. In general, the performances of the neural network-based approaches (experiments 2–4) are better than those of HMM-based approaches due to their powerful ability to model nonlinear acoustic features. In addition, benefiting from the automatic alignment mechanism, the LER obtained by the CTC-based end-to-end ASR

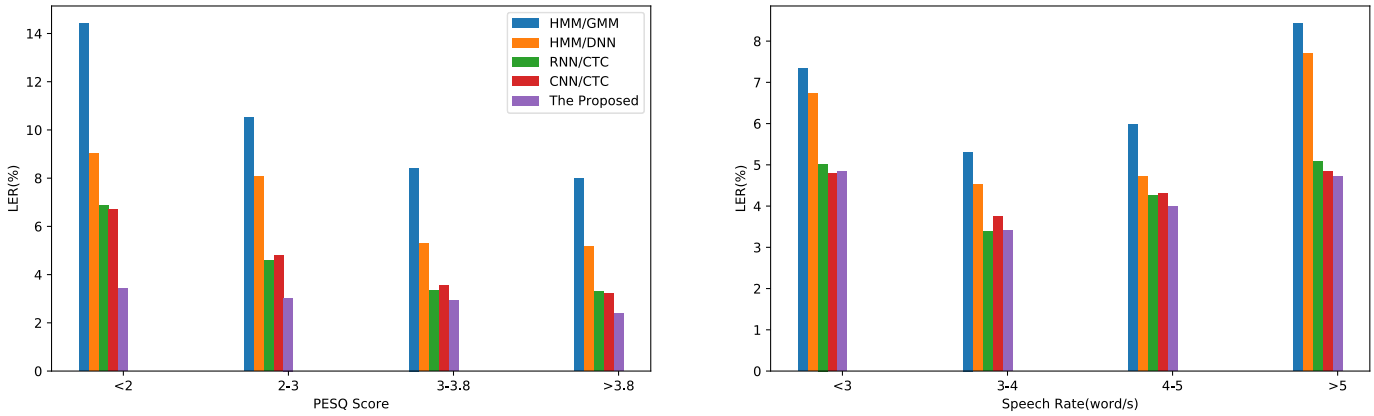


Fig. 7. Performance of different PESQ and speech rates obtained by different ASR approaches.

TABLE V
PERFORMANCE OF DIFFERENT ASR METHODS ON OUR CORPUS

No.	Methods	LM	LER (%)
1	HMM/GMM	3-gram	5.94
2	HMM/DNN	RNN based	5.17
3	RNN/CTC		4.23
4	CNN/CTC		4.71
5	The proposed		3.95*

approach constitutes a 0.94% improvement compared with the HMM/DNN approach. By combining the ASR specificities in ATC and the advantages drawn from the abovementioned three methods, we proposed our framework and obtain the desired performance for the ASR task in the ATC domain.

To further the efficiency and effectiveness of the proposed framework when dealing with the special challenges of the ASR task in ATC, we compare the LERs of different speech rates and PESQ scores (speech quality) in Fig. 7. The HMM-based approaches extract acoustic features using handcrafted feature engineering; thus, they are unable to adapt sufficiently when applied to speech data drawn from different distributions, and they cause turbulence in model accuracy when faced with different background noise levels and speech rates, as shown in the experimental results. Due to their powerful ability to model nonlinear features, the HMM/DNN methods obtain better performances compared with the HMM/GMM methods.

As shown in the experimental results (see Fig. 7), due to the automatic CTC alignment mechanism and powerful modeling abilities of neural networks, the RNN/CTC and CNN/CTC-based end-to-end approaches yield accuracies comparable to those of our proposed ASR framework, i.e., LERs of 4.23%, 4.71%, and 3.95%. However, when faced with extreme input conditions (low PESQ scores or high speech rates), the results of the proposed approach are more accurate and more stable. As shown in the figure, the LER of the RNN/CTC approach fluctuates in a wider range than that of the proposed approach. The CNN/CTC baseline has better applicability to different temporal resolutions due to the multiple kernels in the 1-D CNN layer. As shown in the figure, the CNN/CTC-based baseline achieves high accuracy for different speech rates compared with the other baselines. In summary, the multiple 2-D CNN kernels in our proposed framework can address the ASR specificities in ATC by learning the data distribution at different scales, and it shows promising performances on the divergent data in this work.

The training losses of the RNN/CTC, CNN/CTC, and the proposed AM are shown in Fig. 8. Note that the losses of these three approaches are normalized on the "training iteration" dimension. In Fig. 8, the training losses generated by the baseline approaches are generally larger than that of our proposed approach throughout the entire training process. Furthermore, the training loss of the CNN/CTC-based approach is larger than that of the RNN/CTC-based approach and our proposed model, which proves the effectiveness of the RNN architecture in this work. Fig. 8 shows the training losses at the end of the first epoch and the convergence phase. Because the training samples were shuffled at the beginning of the second training epoch, the training loss rebounds to a higher level due to the padding of the input frames. In this work, the ATC speech vocabulary is only a subset of the words in daily life because speakers must comply with the related rules. Therefore, the tight cohesion among ATC speeches is able to promote the ASR performance.

Finally, we used the RTF to evaluate the decoding efficiency of the CNN/CTC, RNN/CTC, and the proposed approach (LM decoding with a GPU). The average RTFs for the test samples are 0.089, 0.140, and 0.147, respectively, which means that the time consumption of the proposed approach for decoding a 10-s speech is approximately 1.47 s. The CNN/CTC-based approach achieves the highest decoding speed because its architecture is totally CNN-based. The RTF of the RNN/CTC approach is better than that of our proposed approach because our proposal involves a cascaded processing paradigm. However, the RTF of our proposal is promising for real-time tasks.

D. Generalization Test

To further demonstrate the effectiveness of the proposed framework, we trained it with other ASR systems built for both the common applications in [8] and [9] and the ATC domain [64]. The corpora are summarized in Table VI; note

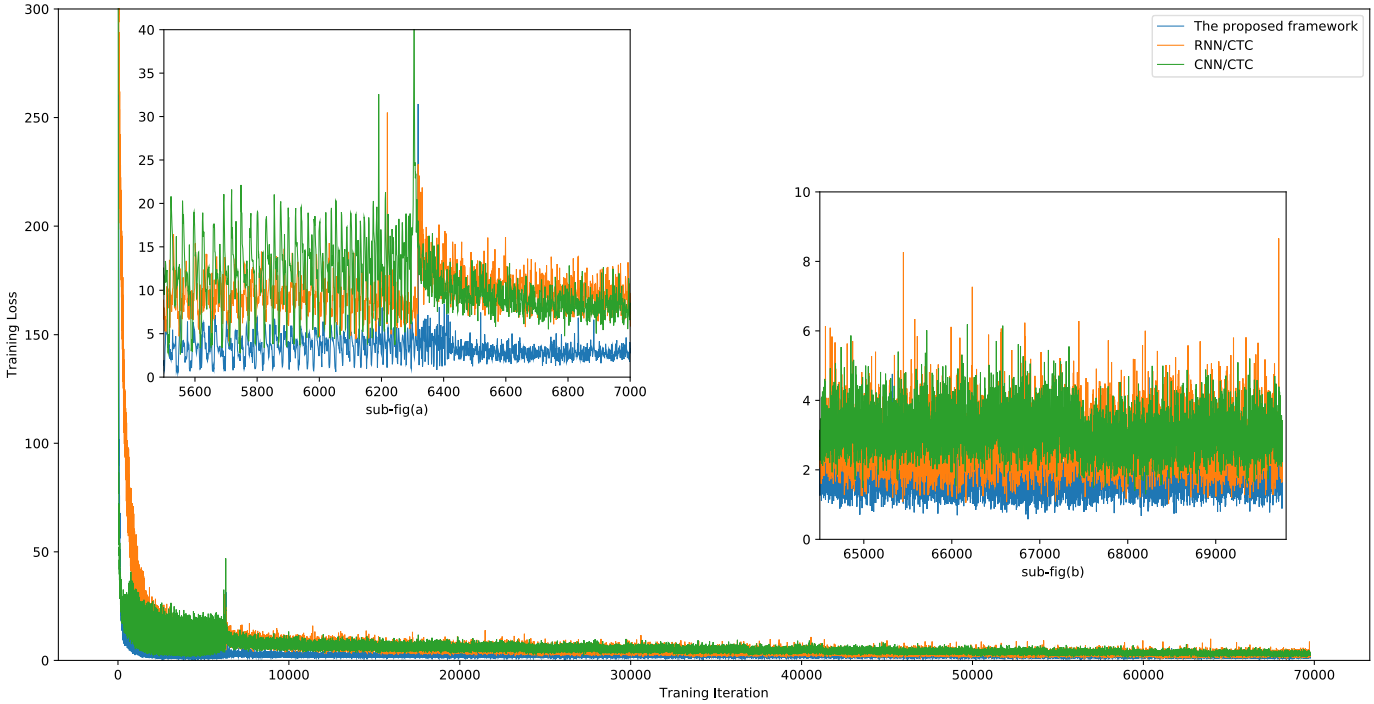


Fig. 8. Comparison of training losses.

TABLE VI
CORPORA FOR GENERALIZATION TEST

Index	Corpus	#Hours	#Words	Language	Domain
1	THCHS30	30	2896	Chinese	Common
2	Librispeech	960	90513	English	Common
3	ATCOSIM	10	859	English	ATC

TABLE VII
RESULTS OF GENERALIZATION TEST

Corpus	Dataset	Baseline	The proposed
Librispeech	dev-clean	8.52	8.49
	dev-other	20.53	18.41
	test-clean	8.71	8.99
	test-other	23.44	20.97
THCHS30	test	8.52	9.14
ATCOSIM	test*	7.61	7.12

* The test set are randomly selected from the ATCOSIM, about 700 utterances.

that no Chinese ATC corpus is publicly available for this experiment. In this section, an RNN/CTC-based model serves as the baseline and has the same architecture as was used in the ASR tests in Section VI-C. All the models were trained without data augmentation and decoded with an n-gram LM. The results are reported in Table VII.

The performance of the proposed approach is comparable to that of the baseline, which is a recent top-ranked ASR model. On some data sets, the performance of the proposed framework is slightly better than that of the baseline for translating noisy speeches. On the large-scale corpus (Librispeech), the baseline obtains better performance than does the proposed framework on clean data sets but inferior performance on noisy data sets, that is, the proposed framework improves the ability to model complex acoustic features. As mentioned earlier, additional errors can be caused by the cascaded paradigm in our proposed approach, which may explain its poorer performance on clean data sets.

For the smaller corpora (THSCHS30 and ATCOSIM), we obtained varying results, i.e., a better performance on ATCOSIM and an inferior performance on THCHS30. These results can be attributed to the large vocabulary, which imposes training burdens on a smaller data set. Actually, in the proposed framework, the AM achieves high performance (6.5% on THCHS30); however, the lower repeatability of phoneme-to-word translation severely degrades the accuracy of the PM. Conversely, due to the intrinsic domain specificity of ATCOSIM, the higher lexical repetition enables the ASR model to obtain higher performance.

In summary, even though the proposed framework originally targeted the ATC domain, it can be migrated to other common applications or other ATC speech corpora. The cascaded paradigm is clearly helpful for integrating the multilingual ASR task. The MCNN can address complex noise models. In addition, a smaller vocabulary would be beneficial to the training and convergence of the proposed approach.

VII. CONCLUSION AND FUTURE WORK

In this work, we proposed a unified framework to solve the multilingual ASR issue in ATC applications. Based on real operating data, we clarified the special challenges for the ASR task in the ATC domain (i.e., volatile background noise and inferior intelligibility, unstable speech rate, multilingual ASR, code switching, and vocabulary imbalance).

Then, we proposed the AM/PM-based novel paradigm to unify the multilingual ASR into a single framework to reduce system complexity. In the AM, multiple CNN kernels with AP operations are proposed to address two ASR issues (i.e., background noise and unstable speech rate). A machine translation-based PM is proposed based on the task that applies an encoder–decoder architecture. Phoneme- and word-based LMs are applied to build correlations between the code-switching words and common words. LMs are also used to correct spelling errors guided by the AM or PM predictions. The experimental results on large amounts of real operating data show that the proposed approach performs the ASR task better than other approaches, achieving an LER of only 3.95%. In addition, the proposed framework was applied to a common ASR corpus and other ATC speech corpora. The proposed ASR framework can serve as a strong support tool in air traffic systems, such as by checking the correctness of a pilot's repetition and understanding the controlling intent for safety monitoring.

In the future, we first plan to increase the data set size to cover more words and improve the data diversity. More effective architectures will also be considered to extract features to support the speech recognition task, such as residual blocks and ConvLSTM blocks. Meanwhile, we also plan to explore applications in air traffic research as that could utilize the proposed ASR approach.

ACKNOWLEDGMENT

The authors would like to thank the Southwest Air Traffic Management Bureau, CAAC, and Wisesoft, China, for building the corpus. They would also like to thank the other members of the research team, Zhongping Yang, Jun Yang, Jing Yu, and Yufeng Zhang.

REFERENCES

[1] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognit.*, vol. 44, no. 3, pp. 572–587, Mar. 2011.

[2] Y. Lin, X. Tan, B. Yang, K. Yang, J. Zhang, and J. Yu, "Real-time controlling dynamics sensing in air traffic system," *Sensors*, vol. 19, no. 3, p. 679, Feb. 2019.

[3] C. M. Geacăr, "Reducing pilot/ATC communication errors using voice recognition," in *Proc. 27th Int. Congr. Aeronaut. Sci.*, 2010, pp. 1–7.

[4] J. Ferreiros *et al.*, "A speech interface for air traffic control terminals," *Aerosp. Sci. Technol.*, vol. 21, no. 1, pp. 7–15, Sep. 2012.

[5] H. Helmke, O. Ohneiser, T. Muhlhausen, and M. Wies, "Reducing controller workload with automatic speech recognition," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2016, pp. 1–10.

[6] Y. Lin *et al.*, "A real-time ATC safety monitoring approach based on deep learning," in *Proc. 8th Conf. CCATM, Chin. Soc. Aeronaut. Astronaut.*, 2017, pp. 1–6.

[7] Y. Lin, L. Deng, Z. Chen, X. Wu, J. Zhang, and B. Yang, "A real-time ATC safety monitoring framework using a deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 23, 2019, doi: 10.1109/TITS.2019.2940992.

[8] D. Wang and X. Zhang, "THCHS-30: A free Chinese speech corpus," Dec. 2015, *arXiv:1512.01882*. [Online]. Available: https://arxiv.org/abs/1512.01882

[9] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[10] *Manual on the Implementation of ICAO Language Proficiency Requirements*, 2nd ed. International Civil Aviation Organization, Montréal, QC, Canada, 2010.

[11] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *J. Acoust. Soc. Amer.*, vol. 24, no. 6, pp. 637–642, Nov. 1952.

[12] P. Denes, "The design and operation of the mechanical speech recognizer at university college london," *J. Brit. Inst. Radio Eng.*, vol. 19, no. 4, pp. 219–229, Apr. 1959.

[13] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics*, vol. 4, no. 1, pp. 52–57, 1972.

[14] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Amer.*, vol. 50, no. 2B, pp. 637–655, 1971.

[15] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, no. 1, pp. 67–72, Feb. 1975.

[16] F. Jelinek, L. Bahl, and R. Mercer, "Design of a linguistic statistical decoder for the recognition of continuous speech," *IEEE Trans. Inf. Theory*, vol. 21, no. 3, pp. 250–256, May 1975.

[17] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, no. 4, pp. 532–556 Apr. 1976.

[18] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 11, pp. 1870–1878, Nov. 1990.

[19] M. Aymen, A. Abdelaziz, S. Halim, and H. Maaref, "Hidden Markov models for automatic speech recognition," in *Proc. Int. Conf. Commun., Comput. Control Appl. (CCCA)*, Mar. 2011, pp. 1–6.

[20] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1035–1074, Apr. 1983.

[21] D. X. Sun and F. Jelinek, "Statistical methods for speech recognition," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, p. 650, Jun. 1999.

[22] B. H. Juang and L. R. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, Aug. 1991.

[23] R. P. Lippmann, "Review of neural networks for speech recognition," *Neural Comput.*, vol. 1, no. 1, pp. 1–38, Mar. 1989.

[24] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[25] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 11, Apr. 2005, pp. 49–52.

[26] Y. Steve *et al.*, *The HTK Book*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[27] M. Stuart and M. Manic, "Survey of progress in deep neural networks for resource-constrained applications," in *Proc. 43rd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2017, pp. 7259–7266.

[28] A. Abe, K. Yamamoto, and S. Nakagawa, "Robust speech recognition using DNN-HMM acoustic model combining noise-aware training with spectral subtraction," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Jan. 2015, pp. 2849–2853.

[29] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[30] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[31] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8614–8618.

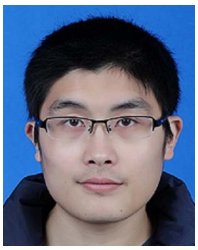
[32] L. Deng, O. Abdel-Hamid, and D. Yu, "A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6669–6673.

[33] Y. Zhang *et al.*, "Towards end-to-end speech recognition with deep convolutional neural networks," in *Proc. Interspeech*, Sep. 2016, pp. 410–414.

[34] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2015, pp. 167–174.

[35] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 273–278.

[36] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," in *Proc. Interspeech*, Aug. 2017, pp. 1591–1595.
[37] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4580–4584.
[38] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. Interspeech*, Aug. 2017, pp. 3707–3711.
[39] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, vol. 32, no. 1, pp. 1764–1772.
[40] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1240–1253, Dec. 2017.
[41] T. Zenkel *et al.*, "Comparison of decoding strategies for CTC acoustic models," in *Proc. Interspeech*, Aug. 2017, pp. 513–517.
[42] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech*, Aug. 2017, pp. 939–943.
[43] C.-C. Chiu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.
[44] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4960–4964.
[45] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 31–35.
[46] D. Yu, M. Kolbaek, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 241–245.
[47] V. N. Nguyen and H. Holone, "Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control," *International J. Comput. Inf. Eng.*, vol. 9, no. 8, pp. 1916–1925, 2015.
[48] H. D. Kopald, A. Chanen, S. Chen, E. C. Smith, and R. M. Tarakan, "Applying automatic speech recognition technology to air traffic management," in *Proc. IEEE/AIAA 32nd Digit. Avionics Syst. Conf. (DASC)*, Oct. 2013, p. 6C3-1-6C3-15.
[49] H. Kopald and S. Chen, "Design and evaluation of the closed runway operation prevention device," *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 58, no. 1, pp. 82–86, Sep. 2014.
[50] H. Gurluk *et al.*, "Assistant based speech recognition–another pair of eyes for the arrival manager," in *Proc. IEEE/AIAA 34th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2015, p. 3B6-1.
[51] T. Pellegrini, J. Farinas, E. Delpech, and F. Lancelot, "The airbus air traffic control speech recognition 2018 challenge: Towards ATC automatic transcription and call sign detection," 2018, *arXiv:1810.12614*. [Online]. Available: http://arxiv.org/abs/1810.12614
[52] Y. Oualil, D. Klakow, G. Szaszak, A. Srinivasamurthy, H. Helmke, and P. Motlicek, "A context-aware speech recognition and understanding system for air traffic control domain," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 404–408.
[53] L. Šmídl, J. Švec, A. Pražák, and J. Trmal, "Semi-supervised training of DNN-based acoustic model for ATC speech recognition," in *Proc. SPECOM*. Cham, Switzerland: Springer, 2018, pp. 646–655.
[54] A. Srinivasamurthy, P. Motlicek, I. Himawan, G. Szaszák, Y. Oualil, and H. Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *Proc. Interspeech*, 2017, pp. 2406–2410, doi: 10.21437/Interspeech.2017-1446.
[55] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
[56] J. B. Mariño *et al.*, "N-gram-based machine translation," *Comput. Linguistics*, vol. 32, no. 4, pp. 527–549, Dec. 2006.
[57] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 932–938.
[58] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2010, pp. 1045–1048.
[59] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, Dec. 2015.
[60] B. Yang *et al.*, "ATCSpeech: A multilingual pilot-controller speech corpus from real air traffic control environment," Nov. 2019, *arXiv:1911.11365*. [Online]. Available: https://arxiv.org/abs/1911.11365
[61] J. G. Beerends, A. P. Hekstra, A. W. Rix, and M. P. Hollier, "Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment Part—II: Psychoacoustic model," *J. Audio Eng. Soc.*, vol. 50, no. 10, pp. 765–778, 2002.
[62] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*. Berlin, Germany: Springer-Verlag, 2006, pp. 137–186.
[63] J. Li *et al.*, "Jasper: An end-to-end convolutional neural acoustic model," in *Proc. Interspeech*, Sep. 2019, pp. 71–75.
[64] K. Hofbauer, S. Petrik, and H. Hering, "The ATCOSIM corpus of non-prompted clean air traffic control speech," in *Proc. 6th Int. Conf. Lang. Resour. Eval. (LREC)*, 2008, pp. 1–6.



Yi Lin received the Ph.D. degree from Sichuan University, Chengdu, China, in 2019.

He currently works as an Assistant Professor with the College of Computer Science, Sichuan University. He is also a Visiting Scholar with the Department of Civil and Environmental Engineering, University of Wisconsin–Madison, Madison, WI, USA. His research interests include air traffic flow management and planning, machine learning, and deep-learning-based air traffic management applications.

Dongyue Guo received the M.E. degree from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree with the College of Computer Science.

His research interest is in signal preprocessing.

Jianwei Zhang received the Ph.D. degree from Sichuan University, Chengdu, China, in 2008.

He has taught and conducted research at Sichuan University since 1993. He has published more than 30 articles. His research interests include air traffic management, and intelligent image analysis and processing.

Dr. Zhang received the National Science and Technology Progress Award in China.

Zhengmao Chen received the M.S. degree from Sichuan University, Chengdu, China, in 2003.

He has been a Teacher with Sichuan University since 2003. His research focuses on signal processing, data fusion, and developing the voice record system for air traffic control systems.

Bo Yang received the Ph.D. degree from Sichuan University, Chengdu, China, in 2012.

He has taught and conducted research work at Sichuan University since 1996. His research interest includes deep-learning application for air traffic management.

Dr. Yang received the National Science and Technology Progress Award twice in China.