aerospace

MDPI

Article

Speech Recognition for Air Traffic Control Utilizing a Multi-Head State-Space Model and Transfer Learning

Haijun Liang †, Hanwen Chang † and Jianguo Kong *

College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China; navyliang@cafuc.edu.cn (H.L.); hanwen1902@cafuc.edu.cn (H.C.)
* Correspondence: kongjianguo@cafuc.edu.cn
† These authors contributed equally to this work.

Abstract: In the present study, a novel end-to-end automatic speech recognition (ASR) framework, namely, ResNeXt-Mssm-CTC, has been developed for air traffic control (ATC) systems. This framework is built upon the Multi-Head State-Space Model (Mssm) and incorporates transfer learning techniques. Residual Networks with Cardinality (ResNeXt) employ multi-layered convolutions with residual connections to augment the extraction of intricate feature representations from speech signals. The Mssm is endowed with specialized gating mechanisms, which incorporate parallel heads that acquire knowledge of both local and global temporal dynamics in sequence data. Connectionist temporal classification (CTC) is utilized in the context of sequence labeling, eliminating the requirement for forced alignment and accommodating labels of varying lengths. Moreover, the utilization of transfer learning has been shown to improve performance on the target task by leveraging knowledge acquired from a source task. The experimental results indicate that the model proposed in this study exhibits superior performance compared to other baseline models. Specifically, when pretrained on the Aishell corpus, the model achieves a minimum character error rate (CER) of 7.2% and 8.3%. Furthermore, when applied to the ATC corpus, the CER is reduced to 5.5% and 6.7%.

Keywords: end-to-end ASR; ResNeXt-Mssm-CTC; air traffic control; transfer learning

Citation: Liang, H.; Chang, H.; Kong, J. Speech Recognition for Air Traffic Control Utilizing a Multi-Head State-Space Model and Transfer Learning. *Aerospace* **2024**, *11*, 390. https://doi.org/10.3390/aerospace11050390

Academic Editor: Jules Simo

Received: 12 April 2024
Revised: 7 May 2024
Accepted: 13 May 2024
Published: 14 May 2024

1. Introduction

Aviation operational safety serves as the fundamental basis for the advancement of civil aviation. Air traffic control (ATC) plays a crucial role in ensuring the safety and efficiency of aircraft operations by managing airspace usage and providing guidance for multiple aircraft during takeoffs, landings, and flights. ATC is responsible for maintaining flight order and optimizing airspace utilization. Radio voice communication serves as the predominant mode of communication between ATC controllers (ATCOs) and pilots [1]. ATCOs transmit control instructions to pilots using radio voice communication. Upon receiving these instructions, pilots reiterate them through radio communication to validate their accuracy. In recent years, the integration of ASR technology into the ATC domain has emerged as a means to develop intelligent ATC systems. This integration aims to mitigate the potential for human error and improve flight safety [1–3]. In current procedures, ASR technology plays a pivotal role in the new ATC workflow, and the generation of accurate ASR results is of utmost importance in facilitating subsequent applications.

In recent years, there has been a gradual shift from traditional systems to end-to-end deep learning models. The cost function for connectionist temporal classification (CTC), as proposed by Graves [4], effectively tackles the significant discrepancy in the lengths of input and output sequences. This approach eliminates the need for individually constructing the acoustic model (AM), pronunciation model (PM), and language model (LM), thereby reducing the complexity of the process. Recurrent neural networks (RNNs), such as the widely recognized Long Short-Term Memory (LSTM) [5] and Gated Recurrent Units

(GRUs) [6], address the challenges of gradient explosion and vanishing that occur during training. These models effectively capture long-distance dependency relationships in audio sequences. Combining them with the CTC approach has yielded remarkable outcomes in the domain of speech recognition tasks [7,8]. The Transformer model, as proposed by Vaswani et al. [9], offers several advantages in the field of speech recognition. These include easy parallelizable training and the ability to efficiently capture global features in sequences, which drives the adoption of attention-based models. The global attention mechanism of the Transformer, however, diminishes its capacity to effectively capture local dependency relationships. To tackle this issue, scholars have integrated convolutional neural networks (CNNs) with attention mechanisms, a technique that has demonstrated remarkable efficacy in the domain of speech recognition [10–12]. Simultaneously, scholars have directed their attention towards alternative approaches to sequence modeling, apart from RNNs, as evidenced in the works of researchers [13–16]. Among these techniques, the state-space model (SSM) [17] is a well-established method in the fields of signal processing and control theory. It is extensively utilized in various applications involving continuous or discrete-time series and control problems [18,19]. However, the cyclic time-varying characteristics of the system result in high computational expenses, posing a challenge in its universal application in large-scale modeling tasks. Recently, researchers have proposed a reformulation and equivalent modeling of the state-space model using variable-length kernels in a convolutional manner. This approach simplifies the extended model and enables parallel training [13,14]. Further investigation suggests that an enhanced iteration of the time-invariant state-space model has the capability to effectively represent long-range dependency relationships. This model can serve as a substitute for self-attention mechanisms in sequence-modeling tasks, such as language modeling [20–22].

In the present study, we acknowledge the significance of both global and local dependency relationships in the context of speech recognition modeling. We present a unique amalgamation of ResNeXt [23] and Mssm [24]. The subsequent sections of this paper are structured as follows. In Section 2, the principles and structures of the ResNeXt and Mssm modules are elucidated, providing a comprehensive framework of the model. Furthermore, we present the implementation process of transfer learning techniques and decoding algorithms. In Section 3, detailed information is presented regarding experimental data, experimental platforms, experimental analysis, and specific model parameters. In Section 4, the primary experimental results of the model and pertinent ablation experiments are presented. In Section 5, a comprehensive overview of the entire study is presented, along with an outline of future research directions.

2. Methodology

This section aims to introduce the fundamental components of the design model employed in this study. Section 2.1 introduces the Mssm module. Section 2.2 delineates the comprehensive framework of the model, incorporating the ResNeXt module. Section 2.3 presents the CTC decoding algorithm. Moreover, within the realm of air traffic control, there is a scarcity of speech data, especially in cases where professionally annotated data are essential. The efficacy of end-to-end speech recognition systems is significantly impacted by the quantity, range, and diversity of the training data accessible. This limitation carries substantial implications for the practicality of deploying end-to-end speech recognition systems in civil aviation. Hence, this research employs transfer learning methods from related domains and data augmentation strategies to increase data variability, with the goal of enhancing the model's ability to generalize within a specific task. Section 2.4 presents an overview of transfer learning techniques.

2.1. Mssm Module

The Time-Invariant State-Space Model can be described as a fully linear recurrent network, as demonstrated in Equation (1).

$$\left.\begin{array}{l} x_k = Ax_{k-1} + Bu_k \\ y_k = Cx_k + Du_k \end{array}\right\} y = SSM(u) \tag{1}$$

The process involves the transformation of an input signal, denoted as u, into an output signal, denoted as y, through a concealed process represented by x. Because of the linear nature of this model, it can also be expressed as a convolution [13,25], enabling parallelized training without the need for recurrences. More significantly, the efficiency and effectiveness of this model are enhanced by imposing constraints on the parameters A, B, C, and D, such that they are block-diagonal matrices, and by ensuring the stability of the transition matrix A. This stability condition guarantees that the state-space model (SSM) produces bounded outputs for bounded inputs [21,26]. Additionally, the efficacy of this model is heavily contingent upon its initialization. Studies have demonstrated that by employing a suitable initialization scheme, the system can proficiently encode the historical information of the input signal [20,27]. The amalgamation of these concepts ultimately gives rise to a model known as $S4$ [21]. The $S4$ model can be considered as predominantly unidirectional. For non-causal applications, such as an audio encoder for offline speech recognition, a bidirectional $S4$ with nonlinear activations and pointwise linear layers [28] can be employed, as demonstrated in Equation (2). "Rev" denotes the reversal of the input sequence u.

$$\begin{array}{c} y \leftarrow Cat([S4(u), Rev(S4(Rev(u)))]) \\ y \leftarrow Linear(Activation(y)) \end{array} \tag{2}$$

Inspired by the concept of multi-head self-attention, the $S4$ layer can be expanded in a more flexible manner by projecting the input signal of dimension D_i onto $H \in \{2, 4, 8, \ldots\}$ independent signals of dimension $\overline{D}_i = D_i/H$. Each of these signals is then processed independently using an SSM with independent initialization. While it is feasible to utilize a basic nonlinear activation function like ReLU or GELU, we have chosen to implement a novel gating mechanism, outlined below. Subsequently, the process is iterated to construct a stacked module, as illustrated in Figure 1a. This module functions as an alternative model for $S4$ in Equation (2) by establishing a bidirectional model. This multi-head design provides flexibility in understanding significant temporal intervals and diverse forms of temporal patterns in sequential data.

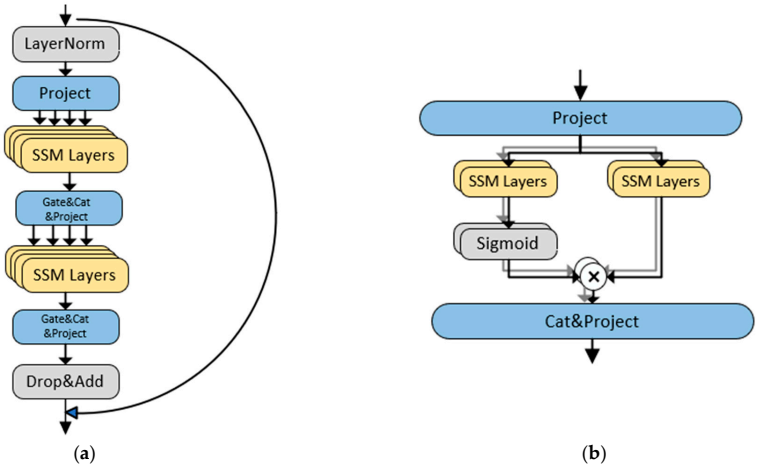

Figure 1. Multi-head generalization and novel inter-head gating idea. Instead of gating the two-dimensional output of a single SSM, this approach is based on elementwise gating of a head using the corresponding output of a different head. (**a**) Stacked and multi-head block. (**b**) Inter-head gating in a multi-head block.

By default, the GELU activation function was employed in prior research, as demonstrated in Equation (2). In certain experimental studies [21,28], it was observed that the activation of GLU also yielded positive outcomes. However, the utilization of a multi-head state-space architecture with H heads offers an increased number of gating possibilities. This design utilizes an inter-head gating (IHG) methodology, in which half of the heads regulate the remaining heads, facilitating communication between different heads. This approach has been demonstrated to produce superior outcomes [24], as illustrated in Figure 1b. The computation of the IHG output involves blending the heads according to the following equation (where σ represents the sigmoid function), as depicted in Equation (3).

$$a^{(h)} = y^{(h)} \cdot \sigma\left(y^{(h+H/2)}\right), h = \{1, \ldots, H/2\} \quad (3)$$

2.2. Overall Architecture of the Model

The Mel spectrogram encompasses information in both the temporal and spectral domains. As ResNeXt serves as the input frontend for our model, we utilize the Mel spectrogram as the input data. The architectural framework of our model is depicted in Figure 2.

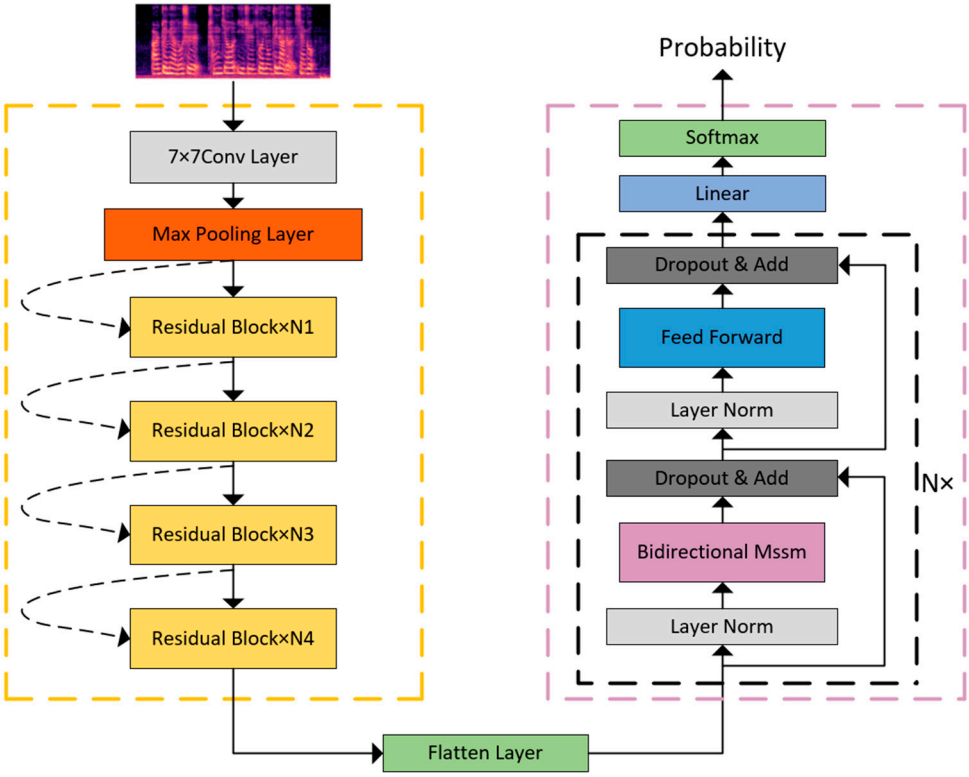

Figure 2. Overall architecture of the model.

The initial component of the model comprises ResNeXt, as shown in Figure 3. The utilization of local receptive fields in convolutional layers enables the network to concentrate on specific features within the data, thereby playing a critical role in the recognition of the local structure present in speech signals. Typically, speech signals exhibit local features in both the frequency and time domains, and the convolutional kernels' local receptive field aids in capturing these crucial features. Convolutional operations exhibit translation invariance, indicating that the model is resilient to shifts in the input. In the field of speech recognition, it is common for speakers to exhibit slight variations in speech rate. The utilization of convolutional layers with translation invariance in the model assists in effectively addressing these variations. The utilization of pooling layers in the model

serves to reduce the dimensionality of the output generated by the convolutional layers. This not only enhances the computational efficiency of the model, but also enhances its resilience to minor variations in the input, such as superfluous information and noise present in speech signals. ResNeXt employs a cascading arrangement of convolutional layers to progressively capture higher-level characteristics, transitioning from local acoustic features to more conceptual speech segments. This approach facilitates the model's comprehension of the hierarchical organization inherent in speech signals. Meanwhile, the incorporation of residual connections mitigates the issue of gradient vanishing that arises from the increase in the depth of CNNs. Finally, the output, which establishes local time–frequency correlations, is subjected to dimension reduction through a flattening layer in order to be utilized as input for the subsequent phase.

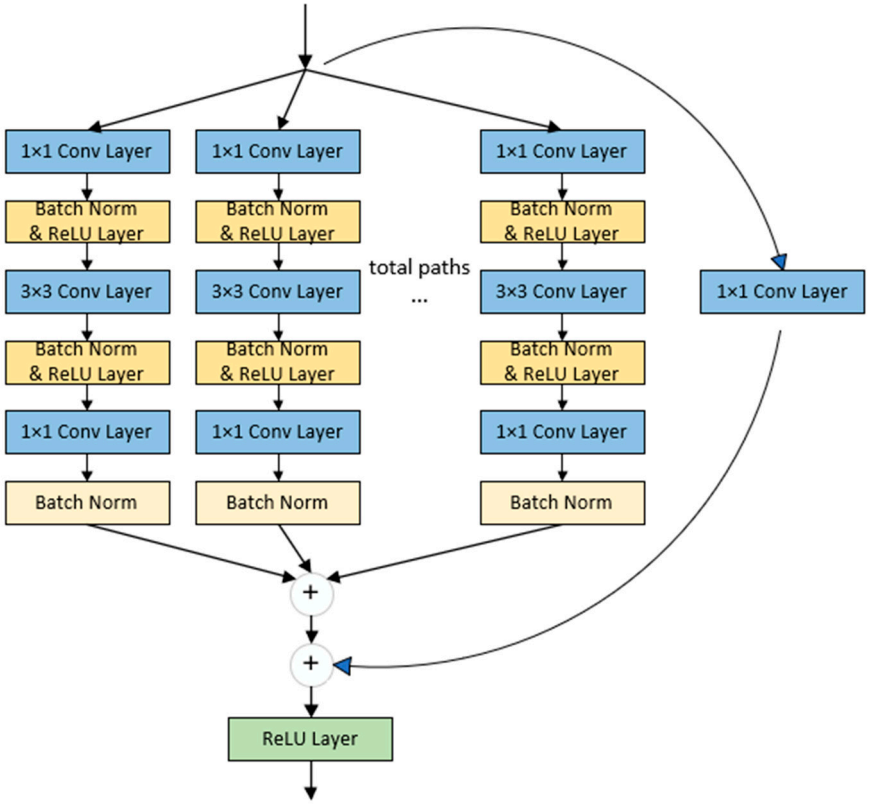

Figure 3. Residual block for ResNeXt50.

The second section predominantly consists of a series of interconnected Mssm modules. Firstly, the utilization of the multi-head mechanism enables the model to acquire diverse representations simultaneously, wherein each head is dedicated to capturing distinct facets of the input data. This phenomenon contributes to the augmentation of the model's ability to accurately represent input data, particularly in scenarios where speech signals contain multiple distinct features. Secondly, the incorporation of a multi-head structure enhances the model's capacity for generalization, thereby enabling it to effectively handle variations such as diverse speakers, fluctuations in speech rate, and environmental noise. Each individual head has the ability to specialize in capturing distinct patterns in speech variations. Finally, given that speech signals are inherently sequential in nature, the utilization of a multi-head structure enables the model to effectively process and attend to information across various temporal scales simultaneously. This contributes to the enhancement of the modeling capacity for temporal relationships in speech signals, thereby facilitating a deeper comprehension of the dynamic characteristics of speech. After these two components, the output is subjected to a linear transformation using a linear layer. This transformation maps the feature dimensions to the number of classes at each time step.

Subsequently, the final classification prediction probabilities are calculated using a SoftMax layer.

2.3. Training and Decoding

The CTC algorithm does not necessitate precise alignment between the input and output. In the proposed end-to-end ASR model, the goal is to predict the text sequence $S = \{s_1, ..., s_m\}$ from the input speech signal $X = \{x_1, ..., x_o\}$, in which s_i is from a special vocabulary based on Chinese characters and English letters. In general, multiple frames in X correspond to a token of S . The length of speech frames is usually much longer than the label length. To address this issue, the CTC loss function was designed to automatically achieve alignment between the speech and label sequence. The tth frame corresponds to the output label k, and its probability is denoted $z^t_{\pi_t}$. Given the speech input X, the probability of the output sequence π is shown in Equation (4). Therefore, the probability of the final sequence can be obtained via Equation (5), in which v is the set of all possible sequences and A denote the length of T sequences over the vocabulary.

$$p(\pi|X) = \prod_{t=1}^{T} z^t_{\pi_t}, \pi \in A \tag{4}$$

$$p(S|X) = \sum_{\pi \in v^{-1}(s)} p(\pi|X) \tag{5}$$

The decoding process employs Beam Search as opposed to the conventional greedy algorithm. Beam Search represents a middle-ground approach that balances the pursuit of the global optimum with the need to manage the trade-off between search time and model accuracy.

2.4. Transfer Learning

In the realm of traditional machine learning, it is a fundamental requirement that the training data and testing data have independent and identically distributed characteristics. Transfer learning techniques from related domains, however, overcome this limitation. Transfer learning has been shown to improve learning performance in the target domain by leveraging knowledge from other domains and reducing the reliance on a substantial volume of data during the learning process. The utilization of transfer learning techniques enables the exploitation of invariant fundamental features and shared structures among interconnected domains, thereby facilitating the transfer and reuse of supervised information.

Specifically, the domain under consideration comprises two distinct components: the feature space X and its corresponding marginal probability distribution $P(X)$, where $\{x_1, \ldots, x_n\} \in X$. Given the domain $\mathcal{D} = \{X, P(X)\}$, the task $\mathcal{T}$ encompasses the label space y and the target prediction function $f(x)$. Furthermore, in the context of transfer learning, when considering a specific source domain D_s and source task $\mathcal{T}_s$, as well as a target domain $\mathcal{D}_t$ and target task $\mathcal{T}_t$ (where $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$), the objective is to leverage the knowledge acquired from $\mathcal{D}_s$ and $\mathcal{T}_s$ in order to improve and optimize the learning efficiency of the prediction function $f_t(x)$ in $\mathcal{D}_t$.

This technique integrates pretraining and knowledge transfer, effectively mitigating the problem of limited data availability in the civil aviation field and tackling specific pronunciation difficulties. Initially, our model undergoes pretraining on a large-scale Aishell dataset in the source domain in order to attain proficient transcription capabilities. Subsequently, the model undergoes fine-tuning and retraining on the relatively smaller target domain ATC corpus, thereby facilitating knowledge transfer and enhancing its ultimate performance through the sharing of weight parameters. Ultimately, the model demonstrates satisfactory performance on the source task and maintains its ability to generalize to the target task.

3. Experimental Evaluation

3.1. Experimental Data

The dataset utilized for pretraining in transfer learning consists of the Aishell-1 corpus and the enhanced Aishell-1 corpus, which incorporates data augmentation techniques. These two datasets are collectively referred to as the extended Aishell corpus. The ATC corpus is sourced from real-time control recordings obtained from the North China Air Traffic Control Bureau of Civil Aviation of China, as well as control simulation training recordings from the Civil Aviation Flight University of China. Please refer to Table 1 for further information. Additionally, the modeling units comprise a total of 4243 Chinese characters, along with one special character denoted as "blank" and one unidentified character referred to as "unk".

Table 1. Basic information of the corpus.

Corpus	Language	Access	Size	Utterances		
				Train	Dev	Test
expanded Aishell-1	Chinese	Public	329 h	240,288	14,326	7176
ATC speech (Mandarin)	Chinese	Simulation	67 h	28,785	1240	1371

3.2. Experimental Platform

The experiments were performed using a Windows operating system. The computer configuration was as follows: Intel Xeon Silver 4110 CPU (Intel, Santa Clara, CA, USA), two NVIDIA RTX2080Ti 11 G discrete graphics cards, 128 GB 2 666 MHz ECC memory, 480 GB SSD, and a 4 TB SATA hard disk. The Pytorch (1.10.2) framework was used to build the neural network model.

3.3. Experimental Analysis

The dataset utilized in this research consists of Mel spectrograms with a shape of (None, 3, 64, 512), where the dimension "None" represents a variable value. These four dimensions correspond to batch size, channels, height, and width, respectively. During the training phase, the forward–backward algorithm is utilized to calculate the loss of the CTC objective function. The Adam optimizer is employed to update the model weights, utilizing an initial learning rate of 0.01, and hyperparameters $\beta\_1 = 0.9$, $\beta\_2 = 0.98$, $\epsilon = 10^{-9}$. Additionally, a learning rate scheduler is employed, which incorporates a warm-up learning rate schedule for the initial 1000 steps, gradually augmenting the learning rate to its initial value. Subsequently, the learning rate experienced a decay of 20% in each training epoch. During the process of inference, a Beam Search technique is employed with a width of 5 in order to acquire the ultimate predicted text.

Additionally, a series of ablation experiments were conducted. Firstly, in the context of the extended Aishell corpus, we conducted a comparative analysis of various representative networks, namely, LSTM, GRU, Transformer (MHSA + FFN), and Mssm, using the same convolutional structure. Then, following pretraining on the ATC corpus, the experimental results were analyzed by manipulating the number of layers in the Mssm module. Finally, following pretraining on the ATC corpus, the Mssm module remained unchanged as we employed various convolutional structures to examine the experimental outcomes. In the conducted experiments, the assessment of the speech recognition task's overall performance was carried out by considering various metrics, including the character error rate (CER), real-time factor (RTF), total parameters, and training time. CER is determined using the formula presented as Equation (6).

$$CER = \frac{I + D + S}{N} \times 100\% \tag{6}$$

where the denominator N represents the total length of the true label and the notations I, D, and S denote the number of the insertion, deletion, and substitution operations, respectively. The RTF was applied to evaluate the decoding efficiency.

$$RTF = \frac{T_d}{T_s} \tag{7}$$

Here, T_d is the time decoded for a speech with a duration of T_s.

3.4. Model Parameters

During the experimental procedure, our model's fundamental structure is ResNeXt 50_Mssm@X, with X denoting the quantity of layers in Mssm. The precise parameter configurations of the model are detailed in Table 2.

Table 2. Details of architecture (input: Mel spectrogram with shape equal to $(None,\ 3,\ 64,\ 512)$.

Structural Order	Output Size	Parameter Setup
Conv layer	$(None, 64,\ 32,\ 256)$	$k = (7,\ 7),\ s = (2,\ 2), f = 64$
Max pooling layer	$(None, 64,\ 16,\ 128)$	$k = (3,\ 3),\ s = (2,\ 2)$
Residual block × 3	$(None, 64,\ 16,\ 128)$	$\{k_1, k_2 = (3,\ 3)\ and\ f_1, f_2 = 64\} \times 3$
Residual block × 4	$(None, 128,\ 8,\ 128)$	$\{k_1, k_2 = (3,\ 3)\ and\ f_1, f_2 = 128\} \times 4$
Residual block × 6	$(None, 256,\ 4,\ 128)$	$\{k_1, k_2 = (3,\ 3)\ and\ f_1, f_2 = 256\} \times 6$
Residual block × 3	$(None, 512,\ 1,\ 128)$	$\{k_1, k_2 = (3,\ 3)\ and\ f_1, f_2 = 512\} \times 3$
Permute	$(None, 1,\ 128,\ 512)$	$(0,\ 2,\ 3,\ 1)$
Flatten layer	$(None, 128,\ 512)$	$start\_dim = 1, end\_dim = 2$
Mssm module × X	$(None, 128,\ 512)$	$numheads = 8,\ ffn\_dim = 2048,$ $mssm\ intermediate\_dim = 512$
Linear layer	$(None, 128,\ 4245)$	$Linear(512, len(dict))$

4. Results and Discussion

4.1. Main Results

Table 3 displays a comparison of the word error rate (WER) performance of our model ResNeXt50_Mssm@12 with contemporary models on the ATC corpus. The models compared include Transformer [29], Conformer in WeNet [30], Branchformer [11], and Zipformer-M [31]. No external language model was employed. Among models with similar parameters, our ResNeXt50_Mssm@12 model demonstrates the lowest character error rate (CER) of 5.5% and 6.7% on the ATC corpus. In comparison to other models, our model exhibits greater competitiveness in recognition outcomes.

Table 3. Evaluation metrics for different models on the ATC corpus.

Model	Type	Params	CER (%)	
			Dev	Test
Transformer [29]	Transducer	64.5 M	6.9	8.2
Conformer in WeNet [30]	CTC/AED	46.3 M	5.8	7.1
Branchformer [11]	CTC/AED	45.4 M	5.6	7.0
Zipformer m [31]	Transducer	73.4 M	5.6	6.9
ResNeXt50_Mssm@12 (ours)	CTC	54.03 M	5.5	6.7

4.2. Ablation Studies

4.2.1. Pretraining Results for Different Backend Models on the Extended Aishell Corpus

Comparison experiments for the models were conducted using the ResNeXt50_Mssm @8 architecture during the pretraining process. In the comparison models, the number of layers in the MHSA + FFN model is identical to that in our framework, with both models consisting of eight layers. Additionally, our proposed model consists of four stacked layers of Bidirectional LSTM (BiLSTM) and Bidirectional GRU (BiGRU), each containing

512 hidden units. The number of trainable parameters in this model is nearly identical to that in the comparison model.

The loss curve for the specific training process is depicted in Figure 4. The iteration comprises of 20 epochs, totaling 100,120 steps. Each step consists of 48 samples, and the loss value is recorded every two iterations. When comparing the convergence of the Transformer (MHSA + FFN) and Mssm models to BiLSTM and BiGRU, it is evident that the former two models demonstrate superior convergence. In the initial phases of training, the convergence rates of BiLSTM and BiGRU exhibit similarity, followed by Transformer (MHSA + FFN), while the Mssm model demonstrates the most optimal convergence speed. In the advanced phases of the training process, the training loss of each model tends to stabilize; however, the training loss of BiLSTM and BiGRU models exhibits notable fluctuations. In contrast, the loss curves of the Transformer (MHSA + FFN) and Mssm models exhibit smaller amplitudes.

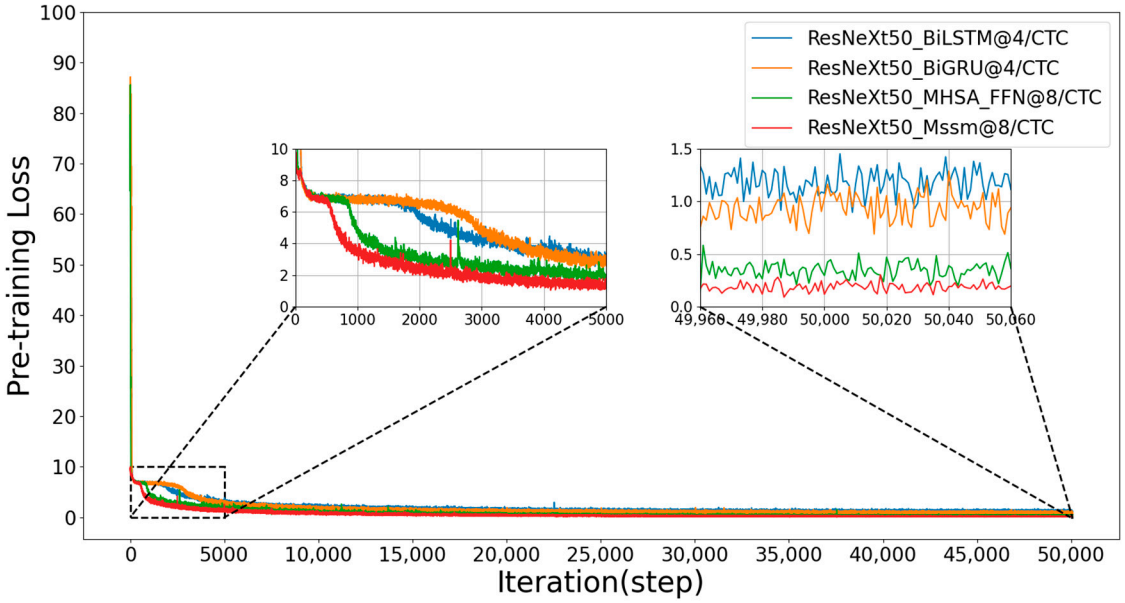

Figure 4. Pretraining loss in different models.

Table 4 presents the comparative outcomes of four models with respect to the CER, RTF, total parameters, and training time. Our model demonstrates superior competitiveness in recognition results compared to other models, achieving a final result of 8.3% on the test set. Additionally, our model demonstrates comparatively reduced time consumption per step and overall training duration. The total number of parameters in our model is 38.5 million, which is 0.82 times, 0.93 times, and 1.18 times the number of parameters in the BiLSTM, BiGRU, and Transformer (MHSA + FFN) models, respectively. Additionally, the mean RTF for the test samples are 0.24, 0.23, 0.20, and 0.21 for BiLSTM, BiGRU, Transformer (MHSA + FFN), and our model, respectively.

Table 4. Evaluation metrics for different backend models.

Model	CER (%)		RTF	Params	Training Time
	Dev	Test			
ResNeXt50_BiLSTM@4	13.0	14.4	0.24	46.9 M	1.51 s/step
ResNeXt50_BiGRU@4	12.9	14.1	0.23	41.2 M	1.36 s/step
ResNeXt50_MHSA_FFN@8	8.1	8.9	0.20	32.6 M	1.33 s/step
ResNeXt50_Mssm@8(ours)	7.2	8.3	0.21	38.5 M	1. 35 s/step

4.2.2. Experimental Results for Mssm Modules with Different Numbers of Layers in the ATC Corpus

To determine the most effective configuration of the proposed framework to enhance task performance, each set of experiments was initially trained using the extended Aishell corpus dataset. The weight set that yielded the most favorable experimental outcomes during the testing phase was preserved as the initial parameters of the model for the subsequent retraining procedure on the ATC corpus dataset. The training process is depicted in Figure 5. A total of ten epochs were conducted, consisting of 10,510 steps, with 48 samples in each step. Mssm modules with 4, 8, and 12 layers all demonstrated prompt convergence. From step 2000 onwards, the convergence speed exhibited greater stability as the number of layers increased.

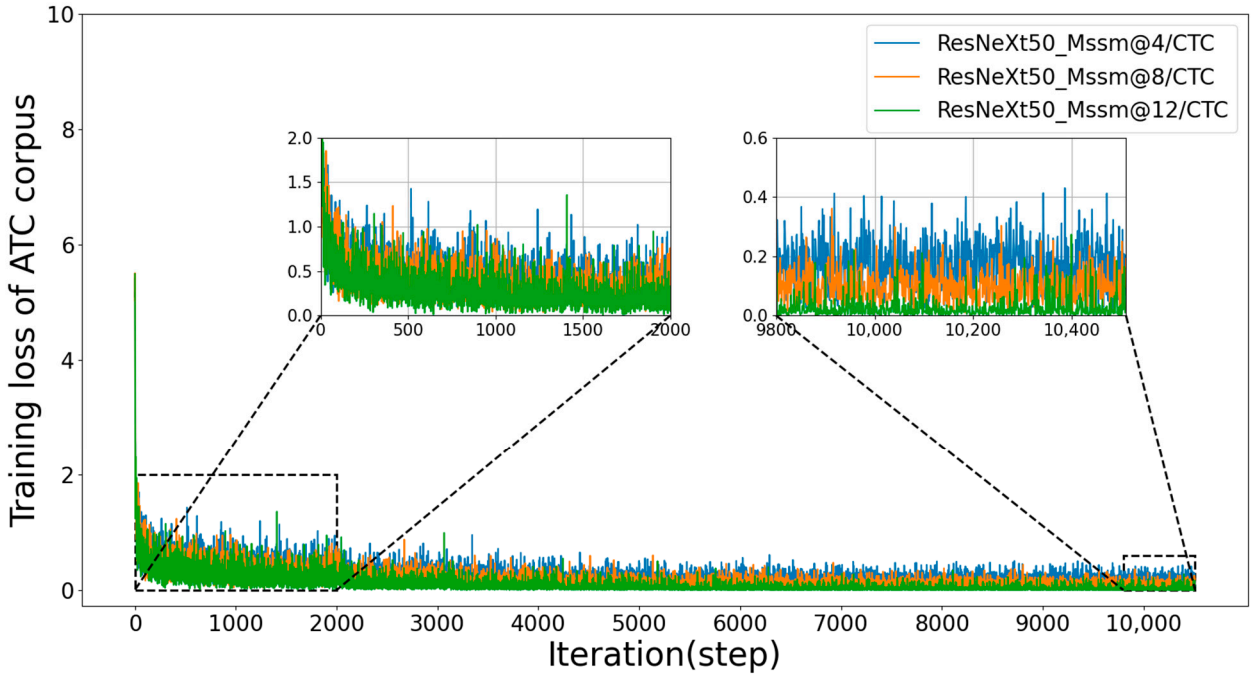

Figure 5. Training loss in Mssm modules with different numbers of layers in the ATC corpus.

The comparison results of the character error rate (CER) with varying numbers of layers in the Mssm module on the dev and test sets of the ATC corpus are presented in Table 5. The CER for the Mssm modules with 4, 8, and 12 layers on the test set are 7.4%, 6.9%, and 6.7%, respectively. The CER for the 12-layer Mssm module exhibits a reduction of 9.5% and 2.9% when compared to the 4-layer and 8-layer Mssm modules, respectively. With the proliferation of layers, there is a direct correlation between the number of layers and the training time, as well as the total number of model parameters, resulting in linear growth. Increasing the number of layers in the Mssm module has a substantial impact on improving recognition accuracy.

Table 5. Evaluation metrics of Mssm module with different layers in the ATC corpus.

Model	CER (%)		RTF	Params	Training Time
	Dev	Test			
ResNeXt50_Mssm@4	6.2	7.4	0.17	22.4 M	1.02 s/step
ResNeXt50_Mssm@8	5.8	6.9	0.21	38.5 M	1. 35 s/step
ResNeXt50_Mssm@12	5.5	6.7	0.24	54.03 M	1.83 s/step

4.2.3. Experiment Results with Different Convolutional Structures in the ATC Corpus

The influence of various convolutional structures on the outcomes of the target task is presented in Table 6. In order to maintain a roughly equivalent total number of parameters, we fixed the number of layers in the Mssm module at 12 and selected VGG16, VGG19, ResNet34, ResNet50, and ResNeXt50 as the experimental controls. There was no statistically significant disparity observed in the duration of training for each network architecture. The duration per step for each of the five models was 2.11 s, 2.23 s, 1.76 s, 1.96 s, and 1.83 s, respectively. The total training time for all models was kept within 6 h. On the dev and test sets, the CERs of the five models were 6.0%, 5.8%, 5.7%, 5.7%, and 5.5%, as well as 7.1%, 7.0%, 7.0%, 6.8%, and 6.7%, respectively. The performance of ResNeXt50 exceeded that of VGG16, VGG19, ResNet34, and ResNet50.

Table 6. Evaluation metrics for different convolutional structures in the ATC corpus.

Model	CER (%)		RTF	Params	Training Time
	Dev	Test			
VGG16_Mssm@12	6.0	7.1	0.23	54.3 M	2.11 s/step
VGG19_Mssm@12	5.8	7.0	0.25	62.4 M	2.23 s/step
ResNet34_Mssm@12	5.7	7.0	0.23	53.5 M	1.76 s/step
ResNet50_Mssm@12	5.7	6.8	0.24	61.6 M	1.96 s/step
ResNeXt50_Mssm@12	5.5	6.7	0.24	54.03 M	1.83 s/step

5. Conclusions

This study introduced a novel end-to-end speech recognition model named ResNeXt-Mssm-CTC. ResNeXt focuses on extracting spectrotemporal features from speech signals. The Mssm module is designed to learn both local and global temporal dynamics in sequential data, while CTC is utilized to automatically manage the inconsistency in the lengths of input and output sequences. Moreover, transfer learning methods aim to mitigate the limited availability of annotated data and the particular phrase pronunciations within the air traffic control (ATC) domain. The model attained a character error rate (CER) of 8.3% in the primary task. Furthermore, the ResNeXt50_Mssm@12 model demonstrated superior recognition performance in the specified task, resulting in a reduction in the character error rate (CER) to 6.7%. This framework is anticipated to function as a foundational element in intelligent air traffic control systems, making a substantial contribution to smooth operation and the overall enhancement of efficiency. Its applications include verifying consistency in repeated instructions, conducting post-event investigations, serving as a backup controller, facilitating air traffic control simulations, and supporting training activities.

Moreover, in forthcoming research, we will collect and expand the ATC corpus to optimize our model. We aim to explore speech recognition tasks in English and mixed Mandarin–English scenarios, with a focus on providing technical support for automatic speech recognition (ASR) applications in the air traffic control (ATC) domain.

Author Contributions: Conceptualization, H.L. and H.C.; methodology, H.C.; software, H.C.; validation, J.K., H.L., and H.C.; formal analysis, J.K.; investigation, H.L.; resources, H.L.; data curation, H.C.; writing—original draft preparation, H.C.; writing—review and editing, H.L.; visualization, J.K.; supervision, H.L.; project administration, H.L.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the National Key R&D Program of China (No. 2021YFF0603904), the Fundamental Research Funds for the Central Universities (No. PHD2023-035) and the Intelligent Civil Aviation Project of the Civil Aviation Flight University of China in 2022 (No. ZHMH2022-009).

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lin, Y. Spoken instruction understanding in air traffic control: Challenge, technique, and application. *Aerospace* **2021**, *8*, 65. [CrossRef]
2. Lin, Y.; Guo, D.; Zhang, J.; Chen, Z.; Yang, B. A unified framework for multilingual speech recognition in air traffic control systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3608–3620. [CrossRef] [PubMed]
3. Lin, Y.; Deng, L.; Chen, Z.; Wu, X.; Zhang, J.; Yang, B. A real-time ATC safety monitoring framework using a deep learning approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *21*, 4572–4581. [CrossRef]
4. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
5. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
6. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
7. Zhang, Y.; Lu, X. A speech recognition acoustic model based on LSTM-CTC. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 1052–1055.
8. Shi, Y.; Hwang, M.Y.; Lei, X. End-to-end speech recognition using a high rank lstm-ctc based model. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 7080–7084.
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
10. Gulati, A.; Qin, J.; Chiu, C.C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv* **2020**, arXiv:2005.08100.
11. Peng, Y.; Dalmia, S.; Lane, I.; Watanabe, S. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In Proceedings of the International Conference on Machine Learning, London, UK, 2–5 July 2022; PMLR: London, UK, 2022; pp. 17627–17643.
12. Kim, K.; Wu, F.; Peng, Y.; Pan, J.; Sridhar, P.; Han, K.J.; Watanabe, S. E-branchformer: Branchformer with enhanced merging for speech recognition. In Proceedings of the 2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar, 9–12 January 2023; pp. 84–91.
13. Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; Ré, C. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 572–585.
14. Voelker, A.; Kajić, I.; Eliasmith, C. Legendre memory units: Continuous-time representation in recurrent neural networks. In Proceedings of the Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
15. Lei, T.; Zhang, Y.; Wang, S.I.; Dai, H.; Artzi, Y. Simple recurrent units for highly parallelizable recurrence. *arXiv* **2017**, arXiv:1709.02755.
16. Lei, T. When attention meets fast recurrence: Training language models with reduced compute. *arXiv* **2021**, arXiv:2102.12459.
17. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
18. Hyndman, R.; Koehler, A.B.; Ord, J.K.; Snyder, R.D. *Forecasting with Exponential Smoothing: The State Space Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
19. Durbin, J.; Koopman, S.J. *Time Series Analysis by State Space Methods*; OUP Oxford: Oxford, UK, 2012.
20. Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1474–1487.
21. Gu, A.; Goel, K.; Ré, C. Efficiently modeling long sequences with structured state spaces. *arXiv* **2021**, arXiv:2111.00396.
22. Mehta, H.; Gupta, A.; Cutkosky, A.; Neyshabur, B. Long range language modeling via gated state spaces. *arXiv* **2022**, arXiv:2206.13947.
23. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
24. Fathullah, Y.; Wu, C.; Shangguan, Y.; Jia, J.; Xiong, W.; Mahadeokar, J.; Liu, C.; Shi, Y.; Kalinli, O.; Seltzer, M.; et al. Multi-Head State Space Model for Speech Recognition. *arXiv* **2023**, arXiv:2305.12498.
25. Chilkuri, N.R.; Eliasmith, C. Parallelizing legendre memory unit training. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: London, UK, 2021; pp. 1898–1907.
26. Smith, J.T.H.; Warrington, A.; Linderman, S.W. Simplified state space layers for sequence modeling. *arXiv* **2022**, arXiv:2208.04933.
27. Gu, A.; Johnson, I.; Timalsina, A.; Rudra, A.; Ré, C. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv* **2022**, arXiv:2206.12037.
28. Goel, K.; Gu, A.; Donahue, C.; Ré, C. It's raw! audio generation with state-space models. In Proceedings of the International Conference on Machine Learning, London, UK, 2–5 July 2022; PMLR: London, UK, 2022; pp. 7616–7633.
29. Zhang, Q.; Lu, H.; Sak, H.; Tripathi, A.; McDermott, E.; Koo, S.; Kumar, S. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7829–7833.

30. Yao, Z.; Wu, D.; Wang, X.; Zhang, B.; Yu, F.; Yang, C.; Peng, Z.; Chen, X.; Xie, L.; Lei, X. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. *arXiv* **2021**, arXiv:2102.01547.
31. Yao, Z.; Guo, L.; Yang, X.; Kang, W.; Kuang, F.; Yang, Y.; Jin, Z.; Lin, L.; Povey, D. Zipformer: A faster and better encoder for automatic speech recognition. *arXiv* **2023**, arXiv:2310.11230.