applied sciences

MDPI

Article

Construction of an Online Cloud Platform for Zhuang Speech Recognition and Translation with Edge-Computing-Based Deep Learning Algorithm

Zeping Fan [1,2], Min Huang [1,2], Xuejun Zhang [1,2,3,*], Rongqi Liu [1,2], Xinyi Lyu [1], Taisen Duan [1,2], Zhaohui Bu [4] and Jianghua Liang [5]

[1] School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China
[2] Guangxi Key Laboratory of Multimedia Communications and Network Technology, Guangxi University, Nanning 530004, China
[3] Guangxi Big White & Little Black Robots Co., Ltd., Nanning 530007, China
[4] School of Foreign Language, Guangxi University, Nanning 530004, China
[5] School of Journalism and Communication, Guangxi University, Nanning 530004, China
* Correspondence: xjzhang@gxu.edu.cn

Abstract: The Zhuang ethnic minority in China possesses its own ethnic language and no ethnic script. Cultural exchange and transmission encounter hurdles as the Zhuang rely exclusively on oral communication. An online cloud-based platform was required to enhance linguistic communication. First, a database of 200 h of annotated Zhuang speech was created by collecting standard Zhuang speeches and improving database quality by removing transcription inconsistencies and text normalization. Second, SAformerNet, a more efficient and accurate transformer-based automatic speech recognition (ASR) network, is achieved by inserting additional downsampling modules. Subsequently, a Neural Machine Translation (NMT) model for translating Zhuang into other languages is constructed by fine-tuning the BART model and corpus filtering strategy. Finally, for the network's responsiveness to real-world needs, edge-computing techniques are applied to relieve network bandwidth pressure. An edge-computing private cloud system based on FPGA acceleration is proposed to improve model operation efficiency. Experiments show that the most critical metric of the system, model accuracy, is above 93%, and inference time is reduced by 29%. The computational delay for multi-head self-attention (MHSA) and feed-forward network (FFN) modules has been reduced by 7.1 and 1.9 times, respectively, and terminal response time is accelerated by 20% on average. Generally, the scheme provides a prototype tool for small-scale Zhuang remote natural language tasks in mountainous areas.

Keywords: automatic speech recognition; natural language processing; neural machine translation; transformer; cloud edge computing; network programming

Citation: Fan, Z.; Huang, M.; Zhang, X.; Liu, R.; Lyu, X.; Duan, T.; Bu, Z.; Liang, J. Construction of an Online Cloud Platform for Zhuang Speech Recognition and Translation with Edge-Computing-Based Deep Learning Algorithm. *Appl. Sci.* **2023**, *13*, 12184. https://doi.org/10.3390/app132212184

Academic Editor: Keun Ho Ryu

Received: 28 September 2023
Revised: 31 October 2023
Accepted: 5 November 2023
Published: 9 November 2023

1. Introduction

The Zhuang people are the largest ethnic minority in China, with a population of approximately 15 M. They predominantly inhabit the southern region of China. Furthermore, there is a significant Zhuang community in the regions bordering China in northern Vietnam, constituting one of the largest ethnic minority groups in Vietnam [1].

The Zhuang language is primarily used in areas of Guangxi inhabited by the Zhuang people [2]. Its underlying vocabulary and pronunciation system are closely related to those of the Dong-Tai language group, which is part of the Zhuang-Dong language family [3]. Internationally, Zhuang is considered an independent language group. However, in China it is classified as part of the "Sino-Tibetan languages" [4]. Zhuang is a minority language with limited coverage and audience as it is highly localized and regionalized. There are no traditional characters, and the language is only transmitted orally, resulting in a lack

of standardized and unified official characters to inherit. Additionally, it is impossible to form a sufficient amount of paper or electronic corpus resources. The lack of an extensive Chinese–Zhuang parallel corpus results in limited research on ASR and MT of Zhuang. Consequently, the current research is still relatively scarce, unable to meet the application requirements of industrial production and community life. An established resolution would be to explore the edge of the cloud platform system with remote speech recognition functions and multilingual translation. Additionally, Guangxi is the window of "Association of Southeast Asian Nations (ASEAN)" cooperation and exchanges in the "Silk Road Economic Belt and the 21st-Century Maritime Silk Road". The development of a machine translation (MT) system for the Zhuang language can not only enrich minority languages and cultures but also enhance communication between Zhuang and other languages in Guangxi and ASEAN nations.

For ASR tasks, the focus is to transcribe the information of the input speech sequence into the corresponding linguistic text. The traditional approach to implementing speech recognition systems involves combining an acoustic model, a pronunciation dictionary, and a language model. However, in recent years, Convolutional Neural Network (CNN) has achieved considerable success in computer vision (CV), as demonstrated by models such as VGG [5], Res-Net [6], and Google-LeNet [7]. Meanwhile, the Transformer [8] has also had breakthroughs in Natural Language Processing (NLP) through the utilization of an attention mechanism. While both CNN and Transformer architectures possess their own advantages and limitations, the newly introduced hybrid CNN-Transformer [9–11] architecture overcomes the lack of CNN's ability to capture global context. This enables end-to-end neural network model to achieve remarkable progress in ASR tasks while also resolving the challenges of forced alignment and multi-module training faced by conventional speech recognition systems. Hybrid architecture becomes a trend.

The rapid development of neural networks also provides novel research directions for a subfield of computational linguistics: Machine Translation. End-to-end Neural Machine Translation (NMT) has emerged as the prominent approach for MT research. Nevertheless, it cannot yet be trusted to work independently in real-world applications. On the one hand, it is challenging for NMT to sufficiently take into account the contextual factors present in the source text, where social, cultural, economic, religious, and political contexts constrain the production of the original text [12]. On the other hand, researchers require an assessment of the Machine Translation Output (MTO) and a considerable quantity of high-quality corpora to improve translation models [13]; for instance, BLEU [14,15]. However, these metrics focus solely on exact match identification, which leads to minimal correlation with assessments made by humans that account for rich morphology [16]. For low-resource languages, the corpus needs to be further collected and expanded, which is an issue that needs to be addressed.

The convolutionally augmented Transformer has complex architecture. For instance, the design integrates several normalization and activation functions, multi-head attention (MHA), and macaron structures. As the model's dimensions increase, so does the computational overhead. Additionally, the attention mechanism involves several large matrix multiplications and data interactions that necessitate high computational and storage overheads [17,18]. To improve the condition, the convolutionally augmented Transformer is structurally optimized to reduce its overall complexity. The model's inference is accelerated by utilizing a Field Programmable Gate Array (FPGA) to make it practical and efficiently deploy the model on a dedicated hardware platform [19–22].

The proposed work has the following contributions:

1. A method for automatic construction method of a large-scale Zhuang speech annotation database is proposed, which allows for acquisition of deep learning data in a short timeframe.
2. SAformerNet, a more efficient and accurate transformer-based automatic speech recognition (ASR) network is achieved by inserting additional downsampling modules.

This model has speedy training and decoding processes, and equivalent recognition performance to the existing ASR model under certain computational resources.
3. A neural machine translation model for Zhuang and other languages using a deep learning architecture is presented.
4. A natural language processing system based on edge computing is proposed for the efficient and accurate multilingual translation of Zhuang's output, thus completing the cycle from theory to practice and efficiently serving the cause of village revitalization.

The rest of the paper is organized as follows. In Section 2, we discuss the current state of research on Zhuang language recognition models. In Section 3, we describe in detail the essential structure of the system being constructed. Section 4 describes the construction of the Zhuang language cloud platform system and conducts experiments. Finally, we conclude in Section 5.

2. Previous Work

Recent end-to-end ASR models typically consist of an encoder and decoder, wherein a strong encoder architecture plays a vital role in achieving optimal ASR system performance. The Conformer architecture efficiently models both global and local dependencies by employing a convolutionally augmented transformer, and it becomes a practical model for ASR tasks as well as various speech processing tasks. Additionally, ref. [23] proposes a progressive downsampling scheme to reduce the training and inference costs of the model. Ref. [24] introduces a similar progressive downsampling scheme, but also includes an upsampling mechanism [25,26] to improve the representational capability of the model. Meanwhile, ref. [27] utilizes an additional add-on downsampling module that fine-tunes the time-frequency speech features. However, the quadratic complexity of the attention layer remains excessively high at longer sequence lengths. Furthermore, downsampling approach reduces the training and inference costs of model, but temporal downsampling leads to unstable and divergent training, weakening the representational capabilities of model.

The Transformer abandons the RNN and CNN commonly used in research and instead adopts the self-attention mechanism to encode the sequenced information, resulting in an improved translation performance and training efficiency of the NMT model. Several pre-training techniques [28,29] have been proposed to train large models using large existing corpora. The pre-training and fine-tuning approach further improves the translation results of NMT. Replicating target data for constructing a pseudo-parallel corpus can yield positive results [30,31] to improve the translation quality of low-resource machine translation. Nevertheless, the lack of corpus resources remains the most common problem of NMT. For Zhuang language, data augmentation and expansion as well as noise reduction of parallel corpus are required for low-resource scenarios.

Transformer-based models are becoming larger and more computation-intensive, and existing FPGA acceleration attempts to overcome this problem in two ways, including data flow optimization and model optimization. Ref. [19] describes splitting large matrices in transformer and designing pulsation arrays based on computational flow, while also optimizing computation of nonlinear functions. Ref. [32] mentions reducing computation including approximation algorithms, along with leveraging parallelism and specializing data paths to optimize hardware utilization. In Ref. [33], the weights were converted to block circulant matrix form and FFT's multiplication method was applied instead of matrix-vector multiplication. FPGAs are high-speed, use significantly less power and cost less to realize system design requirements.

Although ASR and NMT systems have gained significant attention in the field of artificial intelligence (AI), research on the neural network models of Zhuang is at a preliminary stage and existing literature is lacking. Currently, no practical cloud platform or natural language processing system has been developed for Zhuang language. Communicating with volunteers and marketing specialized products in remote mountainous regions can be challenging, particularly with villagers who only speak Zhuang language. This difficulty also extends to cultural exchange and transmission. An established resolution would be to

explore the edge of the cloud platform system with remote speech recognition functions and multilingual translation.

3. System Design

In this section, the system architecture and workflow of the Zhuang speech recognition and translation cloud platform are introduced. The system architecture is consisted of four components, including the Zhuang language database automatic construction system, the Zhuang language speech recognition model, the Zhuang–Other languages neural machine translation model, and the edge cloud platform acceleration, as shown in Figure 1.

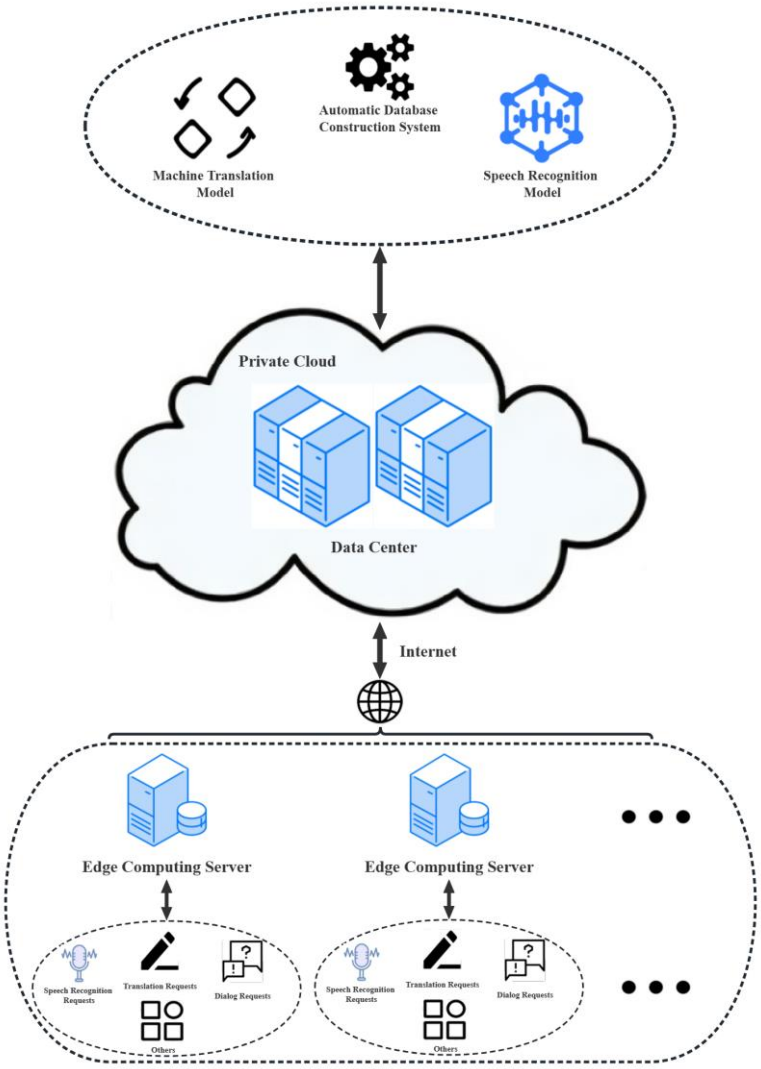


Figure 1. The system architecture of the Zhuang speech recognition and translation cloud platform.

Applying deep learning to speech recognition requires collecting and annotating a large amount of audio data in the Zhuang language. Consequently, an automated system was created to create a database of the Zhuang language. This system is capable of extracting audio clips and video subtitles from the archives of Guangxi TV over the past ten years. This system not only obtains a minimum of 200 h of audio and labelled data but also drastically reduces the amount of manual effort needed. The next step was to develop a speech recognition network. SAformerNet, a more efficient and accurate transformer-based ASR network is achieved by inserting additional downsampling modules. SAformerNet has a faster training speed and a shorter decoding time than Conformer, but with similar recognition performance. The advantage of this development is that the network can be

rapidly deployed. After that, NMT model for translating Zhuang into other languages is constructed by fine-tuning the BART model and corpus filtering strategy. The model is utilized to develop an encoder adaptor comprised of two FFNs and a decoder adaptor including a FFN, and an encoder-decoder attention layer integrated with BART. This is fine-tuned on a specific dataset to preserve computational resources and significantly diminish the duration of model training. Finally, the FPGA-accelerated edge computing approach is used to achieve fast response of network programs, and the optimization of the matrix product of GPU/CUDA and FPGA hardware devices is used to achieve edge-side acceleration. A small terminal platform is designed with the esp32-s3 development board as the main controller. The configuration of the edge server is NVIDIA GeForce GTX 1650, with 4 T storage. The model was trained on a single Nvidia RTX 3090 GPU. The architecture of the model was accelerated on a Xilinx XCZU9EG FPGA.

3.1. Establishment of Zhuang Language Speech Database

Currently, there is no publicly available Zhuang speech database. Online resources solely consist of basic Zhuang dialogues and phonetic tutorials, which are unable to fulfill the model training requirements and are difficult to obtain. Consequently, it is necessary to establish a Zhuang speech database to train the model. The Zhuang language in different regions of Guangxi is different, but the standard is Wuming Zhuang. Since Guangxi TV's Zhuang newscast speaks the standard Zhuang language, the Zhuang language data was obtained from the newscast, the subtitles were extracted, and the audio files were exported from the corresponding video, as shown in Figure 2. The following is the process of establishing the database:

Figure 2. Text timeline correction and audio waveform extraction.

At first, it was necessary to sub-frame the video of the Guangxi TV Zhuang news program and capture all the conversations in it as much as possible. Then, the time when the conversations appear and end was determined. In addition, the video subtitle color parameter was relatively important to help generate the correct timeline information and prevent missing text. To improve the timeline accuracy, the audio that was loaded was Fourier transformed and windowed to eliminate the disturbance. Additionally, the subtitles were synchronized with the audio after being corrected. Following that, the static conversational text was extracted from the images through image processing, whereby the algorithm utilizes grayscale eroding and dilating and contour chroma-lightness similarity methods. Finally, according to the timeline information, Pydub was used to trim the video and audio components of the Zhuang news program from Guangxi TV, resulting in the extraction of the corresponding audio. The raw text was filtered to remove inappropriate content. Symbols such as (<, >], [≪, ≫ were removed. Abbreviations such as GXUTV and

CCTV were capitalized. Sentences that were longer than 25 characters were deleted. All text files were encoded in Utf-8.

Guangxi TV launched a Zhuang language news program in 2013 and has now accumulated ten years of Zhuang language videos. After thorough screening and processing, a database containing 200 h of annotated Zhuang language speech can be established.

3.2. Multi Language Neural Machine Translation Model Based on BART

Machine translation generally means the utilization of computers to achieve automatic translation from one language to another [34]. For the input source language S, the machine translation system can convert it to the target language T. The entire language conversion process can be modeled as the probability of obtaining the target language under the condition of inputting the source language $P(Y)$ as follows:

$$P(Y) = P(T|S) \tag{1}$$

Due to the contextual semantic links between the words in the target sentence, the above probability Equation (1) can be further decomposed into the individual conditional probabilistic factors of the words in the target utterance [35,36]:

$$P(Y) = P(T|S) = \prod_{i=1}^{N} p(T_i|S;\theta) \tag{2}$$

In Equation (2), N is the length of the target sentence and θ is a parameter of the machine translation model.

BART [29] is a pre-trained language model based on the overall structure of the Transformer, which has demonstrated significant improvements in natural language generation tasks. In this study, an NMT model is built by pre-training and fine-tuning parameters using a BART combination adaptor [37–39]. Two BART models, pre-trained as the encoder and decoder, are employed in tandem. The encoder adaptor is installed immediately following the encoder BART layer, and similarly, the decoder adaptor is located right after the decoder BART layer. Only the two adaptors are fine-tuned during training, which not only saves computational resources but also drastically reduces the model training time. The model structure for NMT is presented in Figure 3, where M and N represent the same structure for M and N stacks, respectively.

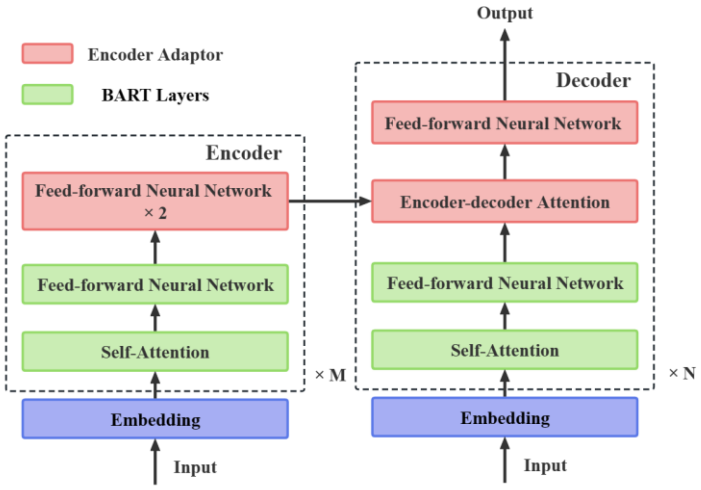


Figure 3. Multilingual Neural Machine Translation Model Architecture.

Specifically, two pre-trained BART models, X_{BART} and Y_{BART}, are used as the source-side language encoder and the target-side language decoder, respectively. The two adaptors are fine-tuned to target the sequence generation framework on a parallel Chinese–other lan-

guage corpus. The loss function of the whole process is modeled as the following equation:

$$L(Y|X;\theta_{AE},\theta_{AD}) = -\sum_{i=1}^{N} logP(y_i|y_{i-1},x;\theta_{AE},\theta_{AD}) \quad (3)$$

L in Equation (3) is the loss function for the entire model, N is the length of the target sequence, x is the input language of the source, θ_{AE} and θ_{AD} are the parameters of the encoding adaptor and decoding adaptor, respectively, and y is the output language of the target. During the training period, the pre-trained models X_{BART} and Y_{BART} in Equation (3) remain unchanged.

The procedure for fine-tuning the encoder is as follows. Initially, the input undergoes layer normalization and enters the first layer of the FFN. The *Tanh* activation function between the two layers of the FFN assigns a non-linear activation to the current hidden state. Subsequently, the input enters the second FFN, where the initial input is combined as the output hidden state of the whole adaptor. The final encoder output is the final representation of the source language after passing through multiple encoder stacks:

$$H_{l+1}^{E} = AE(X_{BART}(H + W_2 \cdot (Tanh(W_1 \cdot (LN(H))))))) \quad (4)$$

In Equation (4), H denote the input hidden state of the encoder adaptor. LN represents normalization, while W_1 and W_2 stand for parameters of two FFN. *Tanh* denotes the nonlinear activation function. H_l^E represents the final outcome of the entire representation of the source-side language encoder. The attention layer of the decoder adaptor will model the interdependence of languages:

$$H_{l+1}^{D} = AD\left(Y_{BART}\left(H_l^D\right), H^E, H^E\right) \quad (5)$$

In Equation (5), H_1^D represents the output hidden state of the $1th$ decoder.

The implementation of BART for machine translation tasks involves fine-tuning the introduced decoding adaptor and encoding adaptor. This not only circumvents extensive tuning of the BART, but also sustains BART's high-precision performance in natural language generation tasks. Since parallel corpora of low-resource languages are data-poor, a pivot language transformation approach was employed to augment and expand the corpus with data [40]. The process for enhancing data with the pivot language transformation method [41–43] is shown in Figure 4.

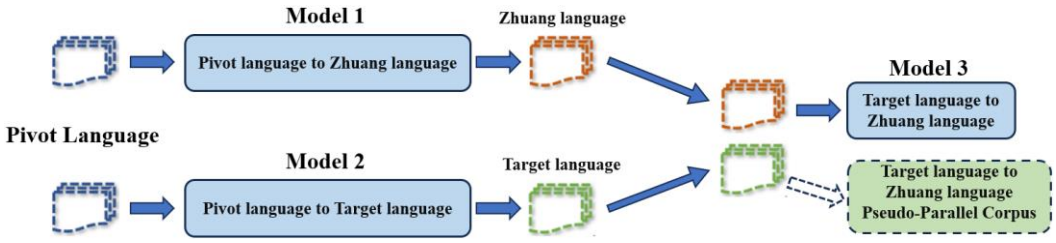


Figure 4. Flowchart for parallel corpus data enhancement based on pivot language transformation.

However, in low-resource situations, the initial base translation model is obtained by training from a small-scale parallel corpus. As a result, pseudo-parallel sentence pairs that contain more noise may be generated, possibly leading to insignificant or limited improvements in the final translation outcomes. To address the issue of low-quality pseudo-parallel corpus, a filtering mechanism is proposed by first using the real parallel corpus to train the reverse neural machine translation models, and then feeding the monolingual corpus of the respective models into the reverse machine translation model to generate translations. Finally, the filtering mechanism is used to filter the resulting translations. The real monolingual data is combined with the filtered translations to create a pseudo-parallel corpus, which is then used alongside the real parallel corpus for further forward machine

translation model training. The trained translation system is then utilized repeatedly to generate new pseudo-parallel data to replace the previous data, and the process is shown in Figure 5.

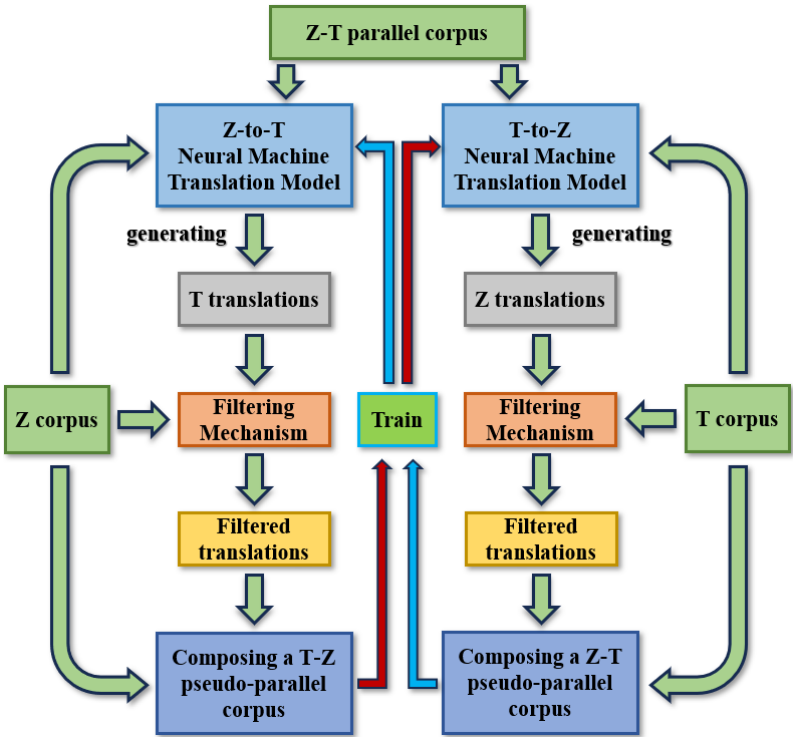


Figure 5. Flowchart of corpus filtering strategy under low resources.

Nevertheless, in the case where there is only one reference translation, it is challenging to use a similarity measure evaluation method that fulfils the requirements of the application. In order to better evaluate the machine translation method, a text similarity measure algorithm based on the ordering of common words was proposed. This algorithm has less computational overhead and can characterize more cases than the similarity measure based on the editing distance. The process of similarity computation can be described as follows:

Firstly, do the partitioning of $Sentence_1$ and $Sentence_1$, assuming that they are divided into k words and l words, respectively, and store the common words of the two clauses in the order of the original sentence in the arrays A and B, respectively. Sorting in terms of A to B. Assuming the length of the array is n and letting $m = 0$, the sorting can be described as:

step 1: *for* $i = 0$ *to* $n - 1$

step 2: *for* $j = 0$ *to* $n - i - 1$

step 3: If the order of $A[j]$ and $A[j + 1]$ is different from the order of the corresponding contents within B, then perform a bubble sort.

step 4: $m = m + 1$

It can be seen that m is the number of executions of the bubbling sort. The similarity of arrays A and B can be described as:

$$Array_{similarity(A,\,B)} = 1 - \frac{2m}{n \cdot (n-1)} \tag{6}$$

The similarity calculated from Equation (6) can be extended to the Position Based Sentence Similarity (PBSS) approach. PBSS based on the positional ordering of shared words can be quantitatively represented as:

$$PBSS(Sentence_1,\ Sentence_2) = max\left(Array_{similarity}(A_x, B_y)\cdot\frac{n}{max(k,1)}\right) \tag{7}$$

If there are words that appear differently in $Sentence_1$ and $Sentence_2$, the computation process in Equation (7) may have multiple groups (A, B), and the similarity of the two sentences is taken as the maximum value obtained from the calculation of the different groups (A_x, B_y).

3.3. An End-to-End Automatic Speech Recognition Algorithm Based on Conformer

In this paper, a more efficient network architecture for solving the ASR problem is designed to achieve a reduction in the complexity of the Conformer while achieving a lower Character Error Rate (CER) for a given computational budget. A novel convolution-attention hybrid downsampling layer is designed, as shown in Figure 6. The novel sampling layer diverges from the Macaron structural design of the Conformer block and resembles the standard Transformer structure more closely. The convolution module is placed between the two feed-forward networks, and the MHSA module is positioned in front of the feed-forward networks. The original convolutional module is replaced by a convolutional downsampling module, shown in Figure 7, to reduce the computational cost of CNNs, and enable faster training and inference. Downsampling is performed to reduce the sampling rate of the input sequences to 40 ms, followed by sequential downsampling to reduce the sampling rate of each input sequence to 80 ms. During this process, the feature sequences are projected to wider feature dimensions to maintain the complexity of each encoder. This effectively reduces the redundancy in the feature embedding vectors learned by the Conformer block, thus further reducing the computational overhead without loss of accuracy.

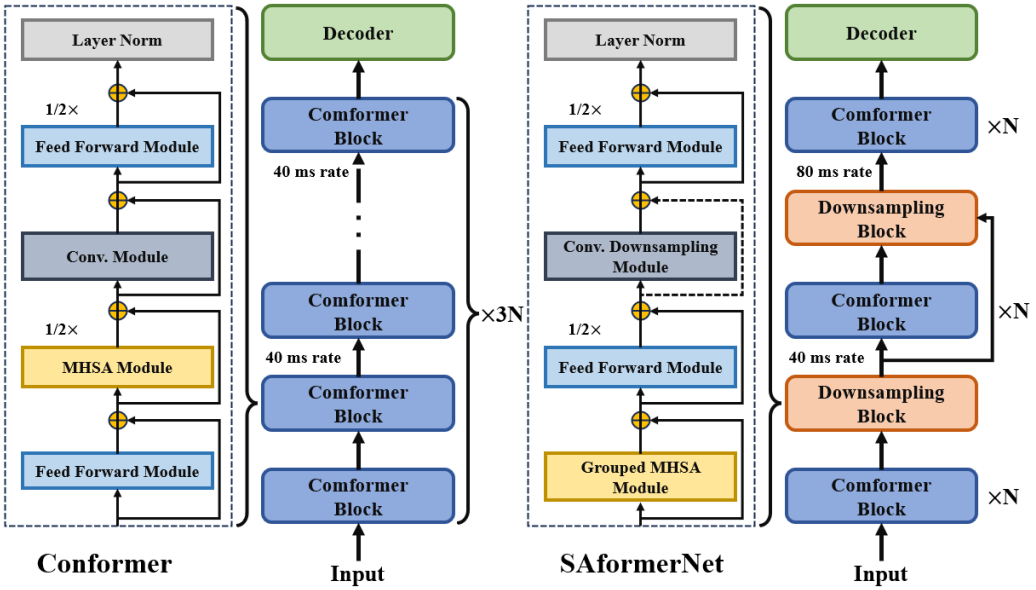


Figure 6. The Conformer architecture (**Left**). The Conformer consists of two macaron-like feed-forward layers, with a MHSA and convolutional module in the middle layer. The feature dimensions are kept constant throughout the network; The SAformerNet architecture (**Right**). SAformerNet consist of multiple Conformer blocks with a hybrid convolution-attention structure. The coded sequences are subjected to multiple downsampling operations before being projected to wider feature dimensions.

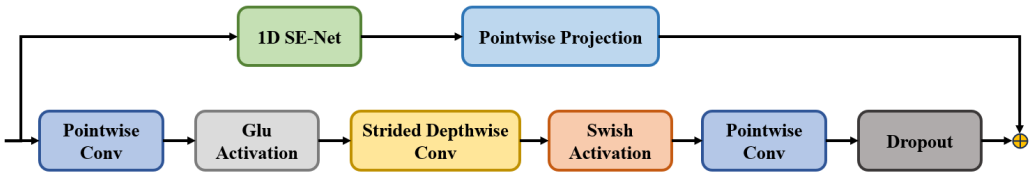


Figure 7. The Convolution Downsampling Module. The residual module contains a pointwise projection layer and a 1D SE-Net. The strided depth-wise convolution is responsible for the downsampling function; The 1D SE-Net utilizes a convolutional layer to extract features, which are pooled to compress the convolutional result into a 1D vector and applied to the convolutional output using pointwise multiplications.

However, temporal downsampling results in unstable and divergent training, and downsampling the feature vectors reduces the amount of contextual information available to the decoder, thus hindering the successful decoding of the entire sequence. Inspired by Res-Net, a new network structure is introduced as a squeeze-and-excite (SE) block [44,45] in the convolution downsampling module by residually connecting some of the conformer blocks, as shown in Figure 7. The SE block can recalibrate features to selectively emphasize informative features, thereby mitigating information deficits attributable to downsampling. For input x, the output y of the SE block is denoted by:

$$\begin{gathered} \overline{x}_t = \frac{1}{T}\sum_t x_t, \;\; \theta(x) = Sigmoid(W_2(Act(W_1\overline{x}_t + b_1)) + b_2) \\ SE(x) = \theta(x) \circ x_T \end{gathered} \tag{8}$$

In Equation (8), $Act(\cdot)$ denotes the activation function, $\theta(x)$ is a global channel-wise weight, and $Sigmoid(\cdot)$ refers to the sigmoid function. Where $\circ$ represents element-wise multiplication, W_1, W_2 are weight matrices, and b_1, b_2 are bias vectors.

The residual module contains a pointwise projection layer and a 1D SE-Net, which uses a convolutional layer for feature extraction. The resulting features are then pooled and compressed into a 1D vector before being applied to the convolutional output through pointwise multiplications. The Convolution Downsampling module incorporates a strided depth-wise convolution responsible for the downsampling function, with batch normalization and activation applied after each convolution. The advantages of SE block feature recalibration can accumulate across the network and can be adapted to the needs of the network, especially in deep Conformer Encode. This allows for further reduced redundancy without compromising accuracy.

To further reduce the computational overhead in SAformerNet, the MHSA Module is replaced by the Grouped MHSA Module [23]. The quadratic complexity of the attentional mechanism increases its computational overhead dramatically over long sequences, causing networks of different depths to process completely different amounts of data, especially those that have been downsampled, impairing overall efficiency. The Grouped MHSA Module reduces the attentional complexity from $O(n^2 \cdot d)$ to $O(n^2 \cdot d/g)$ by first grouping long sequences and then performing the attentional operation, and applies it to networks of different depths so that the processed data is approximately the same in different networks, which improves the overall efficiency of SAformerNet. In the MHSA Module, a number of heads H for a hidden sequence $X \in \mathbb{R}^{n \times d}$ is computed as:

$$H_h = softmax\left(\frac{Q_h K_h^T + S_h^{rel}}{\sqrt{d_h}}\right) V_h \tag{9}$$

And in the Grouped MHSA Module, the header H in Equation (9) transforms into:

$$H_h^{grp} = softmax\left(\frac{Q_h^{grp} K_h^{grpT} + S_h^{rel}}{\sqrt{d_h'}}\right) V_h^{grp} \tag{10}$$

Equation (10) is the encoder output after the Grouped MHSA Module. Where Attention queries, keys, values are reshaped from Q, K, $V \in \mathbb{R}^{n \times d}$ to Q^{grp}, K^{grp}, $V^{grp} \in \mathbb{R}^{n' \times d'}$. $S^{rel} \in \mathbb{R}^{n \times n}$ is a relative position score matrix. $n' = n/g$ and $d' = d \times g$. And concatenated Grouped attention output $H^{grp} \in \mathbb{R}^{n' \times d'}$ is reshaped to $H \in \mathbb{R}^{n \times d}$ before the output projection layer. The Grouped attention is applied at the network layer, where the coding sequences are the longest, and the study shows that this approach reduces the computational overhead and improves the efficiency of the model.

3.4. Acceleration Method of Edge Computing for Multimodal Cooperative Operation Based on Cloud Platform

There is currently no online translation system available for Zhuang language speech translation, and it is also impossible to run multimodal networks cooperatively. To handle complex tasks on the network, such as image recognition algorithms and natural language processing, etc., the use of edge computing algorithms is necessary to accelerate the network operations and ensure that the network's response speed satisfies the real demand. In this paper, the focus is on how to operate multimodal networks with different programming languages and call different hardware resources on the same network edge platform, to maximize resource sharing and achieve the purpose of low-carbon.

This system uses the cross-platform programming language XOJO to quickly build a cloud platform. XOJO is a cross-platform integrated development environment (IDE), consisting of an integrated debugger, multi-platform compiler, and other essential components. Thanks to cross-compilation, XOJO facilitates developing applications on Windows that can effortlessly run on Linux and Mac OS X systems. XOJO-developed programs can be compiled directly into CPU executable instructions, resulting in better performance by using the LLVM compiler tool. This system employs the developmental capabilities of its web program. Web applications support the latest versions of popular web browsers, including Internet Explorer, Firefox, Safari, and Chrome. Web applications can be accessed as long as the application on the server remains running. The security of web applications is of utmost importance, as they can be accessed by any online user. XOJO web applications are compiled into binary code, and the source code is not stored on the server. To modify the application, an individual needs to possess extensive knowledge of x86 assembly code and commit a significant amount of time to code tracing. This task is considerably more challenging compared to hacking PHP, CSS, JavaScript, AJAX, and HTML.

For some algorithms with low computational requirements, such as audio data transmission, and some database operation management, the XOJO platform can be utilized. The deep learning algorithms and other computationally intensive programs are accelerated by edge computing for network acceleration [46,47]. Unlike XOJO's web programs, the programs at the edge reside on the server as a cluster of executable programs that parse the relevant commands, give the appropriate results, and finally return them to the client through the Internet. The Algorithms Kernel module can be easily exchanged with information between programs due to the unified packaging and standard interface parameters, which allow the client application to operate without restrictions based on the user's system platform. To ensure that the algorithmic processing time of SAformerNet and BART models is within the user's tolerable range, FPGAs are used for algorithmic acceleration in the local computing at the edge, and the cloud platform framework is shown in Figure 8.

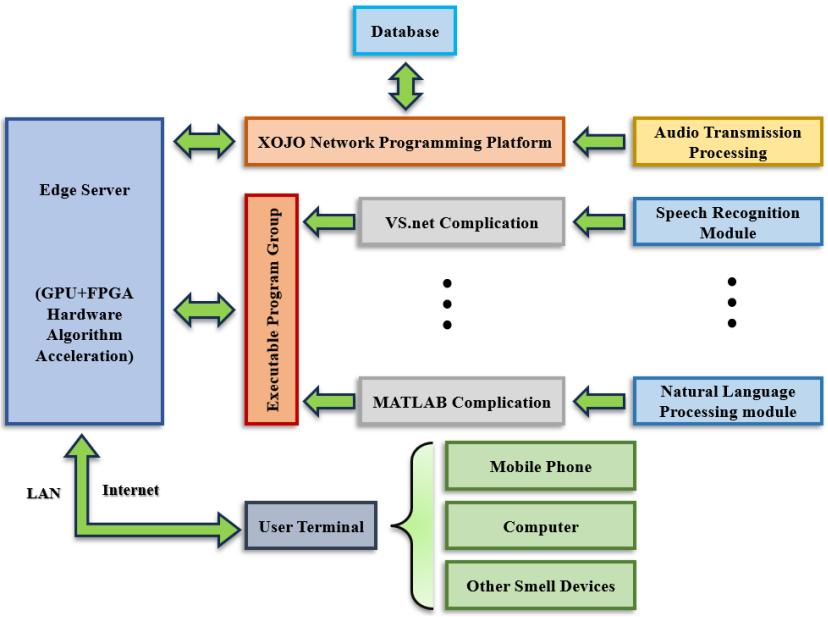


Figure 8. Cloud platform system framework.

The FPGA acceleration for edge computing utilizes the PYNQ chip that adds Python support in contrast to the original ZYNQ architecture. PYNQ is a heterogeneous SOC that incorporates an ARM processor and an FPGA programmable logic device, resulting in many-fold computation acceleration compared to conventional algorithms, as well as noteworthy reductions in power consumption and cost. After completing the training of the models employed in the system, the designed deep learning models were accelerated using Xilinx's PYNQ Ultra-Scale + MPSoC series of boards, and the FPGA-accelerated development process is shown in Figure 9.

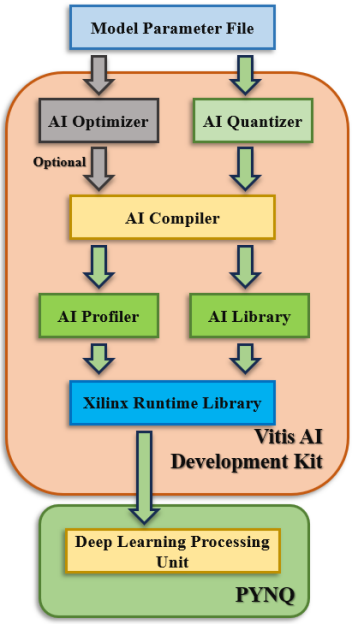


Figure 9. FPGA development process.

If essential, Vitis AI Optimizer can be used after obtaining the model parameter files to prune redundant connections in the neural network and reduce the overall amount of computation required. The Vitis AI Quantizer tool can be used to quantize the model parameters, turning floating-point numbers into fixed-point numbers, which helps in

reducing the amount of memory bandwidth required by the network model. Additionally, the Vitis AI compiler compiles the quantized model into an efficient command set and flow of data. Finally, the Vitis AI Profiler keeps track of function calls and runtimes to activate the deep learning processing unit (DPU) for efficient inference deployment on AI edge using the Vitis AI high-level library and the Xilinx Runtime library.

4. Results

This section describes the relevant experiments and evaluates and analyzes the results.

4.1. Mini-Terminal Platform

A mini terminal platform was developed to evaluate system performance and simulate user interactions. The platform is an integrated system comprising both hardware and software. It captures, processes and analyzes information, including sound, through an array of sensors and control modules. It also transmits the data to the server for business logic processing, as shown in Figure 10. The highlighted areas on the diagram represent the system's respective modules. The system is communicated via a Wi-Fi wireless communication module. The hardware components include the ESP32-s3 primary controller development board, a microphone, a voice wake-up module, an LCD display, an audio amplifier module, and a speaker. The master control ESP32-s3 development board is used as the core control unit and is responsible for operating and transmitting data throughout the system. The microphone is used to capture the sound signal of the user's speech and detect and recognize it through the voice wake-up module. The server is mainly responsible for processing the business logic, parsing user requests and generating the required audio files, and then returning the processed files to the mini-terminal. A deep neural network model is deployed by the server for recognizing audio files and converting them into text. After being processed by the translation or conversational system, the server can respond to user requests and return the response text converted into an audio file to the speakers, thus realizing the function of multilingual interaction.

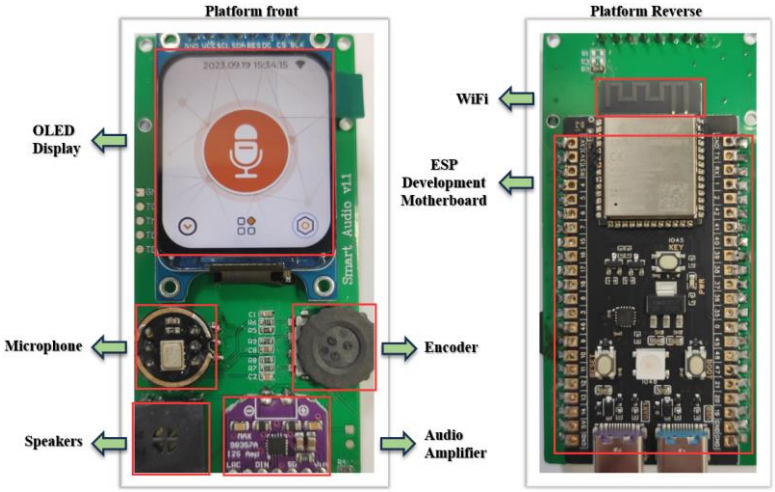


Figure 10. Physical drawing of mini terminal platform structure.

4.2. Experiment Setup

Using the automatic construction system of the Zhuang database, a 200 h database of Zhuang speech with annotations was built for the training of speech recognition models by filtering the videos of Guangxi TV over the past ten years. The major dialect of this database is Wuming Zhuang, and the conversation topics include society, finance, science and technology, entertainment, and health, all of which are used for the training of ASR. SpecAugment [48,49] was employed to augment data and prevent overfitting during training. Specific parameters were set, including a frequency mask size parameter (F = 10) and the time and frequency mask ($mT = mF = 2$). The input features are 80-dimensional

mel-scale log filter bank spectrograms, which were calculated with a 25 ms window and a 10 ms shift. The publicly available dataset AISHELL-1 [50] was also used as a pre-feasibility validation of ASR. The 178 h open-source AISHELL-1 speech database used high-fidelity microphones to record 400 speakers from diverse accent regions in China, involved in intelligent households, drones, industrial production, and other fields. The parallel corpus for MT was selected from the Asian Scientific Papers Excerpt Corpus (ASPEC) version 1.0 [51], with approximately 680,000 parallel utterances. The corpus was divided into 610,000 pairs of training data, 2000 pairs of development data, and 2000 pairs of test data. The development and test data were randomly selected. In addition, 5000 language pairs were randomly selected from the publicly available bilingual dataset OPUS JA-ZH for manual alignment, which was used to validate the generalization ability of the model. The model was trained on a single Nvidia RTX 3090 GPU using the open-source deep learning framework PyTorch, version 2.0.1. To optimize training efficiency and maximize GPU memory usage, the data loader was set up to sort and pack statements based on frame length, and then randomly select these statements to feed to the model.

For the speech recognition model, the corresponding SAformerNet-S and M were designed using Conformer-S and M under the reference of the architectural parameters, keeping the model size constant, and the hyperparameters of the model were set as shown in Table 1. The utilization of four attention heads and setting the hyperparameter g to 3 provides a good balance between computational effort and accuracy. CER evaluates the error rate between the predicted text and the original text, a lower CER indicates better ASR performance. The machine translation model's encoder network comprises six modules that are identical in structure. Each module consists of two sublayer structures, including Multi-head Self-attention and a Fully connected Feed-forward Network (FFN), which perform residual concatenation and normalization on the output of each sublayer. The decoder also has six modules that are identical in structure, with one more Multi-head Attention Network layer than the encoder module. The vector size of the word is 512 and the output dimension of the first linear layer is 2048 using the 8-head attention mechanism. The Byte Pair Encoding (BPE) segmentation algorithm was utilized. Word list size was set to 32,000, and the batch size was set to 32 due to hardware constraints. The effectiveness of machine translation models is conducted with the BLEU. The degree of similarity between two sentences can be determined by the BLEU, which calculates a composite score. A higher score indicates improved machine translation quality. The training data, validation data, and verification data used in the experiment were randomly selected and do not overlap with each other.

Table 1. Detailed architectural configuration of SAformerNet and Comformer.

Model	Encoder Blocks	Encoder Dims	Attention Heads	Group Size	Params (M)
Conformer-S	16	176	4	-	13.0
SAformerNet-S	15	120,168,240	4	3	13.4
Conformer-M	18	256	4	-	30.6
SAformerNet-M	16	180,256,360	4	3	33.4

4.3. Parallel Corpus Selection and Retranslation Results

Data cleaning methods, such as long sentence splitting, de-duplication, and line length ratio control, were applied to the dataset. The original corpus and the slice-filtered parallel corpus were used for training, respectively, and the results are shown in Table 2. The experiments show that the use of the processed parallel corpus, validated on ASPEC-JC, the BLEU score of the Chinese-to-Japanese neural machine translation model improves by 0.92, and the BLEU of the Japanese-to-Chinese neural machine translation model improves by 0.83.

Table 2. Translation effects when different data are used for training and tests.

Translation Model	Data Types	ASPEC-JC BLEU Scores	OPUS JA-ZH BLEU Scores
Chinese-to-Japanese	Original data	34.24	34.20
Japanese-to-Chinese	Original data	32.19	33.41
Chinese-to-Japanese	Processed data	35.16	35.45
Japanese-to-Chinese	Processed data	33.02	33.10

From the model's BLEU score on OPUS JA-ZH, the model has good robustness. However, the Japanese-to-Chinese model, which was trained using the processed data, did not obtain better translation results, indicating that enhancing the quality of the parallel corpus may not necessarily improve the model's generalization ability. The generated pseudo-parallel sentence pairs contain more noise and may provide little or no improvement to the final translation. To enhance the generalization ability of the model, the initial Transformer model is first trained using all the parallel corpus in ASPEC-JC, resulting in two NMT models. The subsequent training employs only the screened real parallel corpus except for the pseudo-parallel corpus. As can be seen from Table 3, the generalization ability of the model is enhanced, improving the effectiveness of NMT. The neural machine translation model from Chinese to Japanese, which incorporates a filtering mechanism, enhances the BLEU score on ASPEC-JC by 1.54. Likewise, the model that translates from Japanese to Chinese enhances the BLEU score by 2.8 compared to the cleaned data.

Table 3. Model performance under different retranslation strategies.

Translation Model	Retranslation Strategies	ASPEC-JC BLEU Scores	OPUS JA-ZH BLEU Scores
Chinese-to-Japanese	No filtering mechanism	35.97	36.12
Japanese-to-Chinese	No filtering mechanism	34.31	34.36
Chinese-to-Japanese	With filtering mechanism	37.51	37.74
Japanese-to-Chinese	With filtering mechanism	35.82	35.92

4.4. Speech Recognition Results

Table 4 provides a comparison of CER between SAformerNet, Conformer, and other ASR models. The results show that SAformerNet has improved compared to previously published systems. Moreover, the SAformerNet-S model achieved competitive results of 5.50%/5.76% on the AISHELL-1 dataset with only 13.4 M parameters. However, this is still a departure from the original work, as it benefited from training with bigger volumes and a more comprehensive set of resources. Furthermore, the experimental results show that the feasibility of the SAformerNet speech recognition model is verified, and a recognition rate of 5.92% is achieved by training and testing on the Zhuang language dataset.

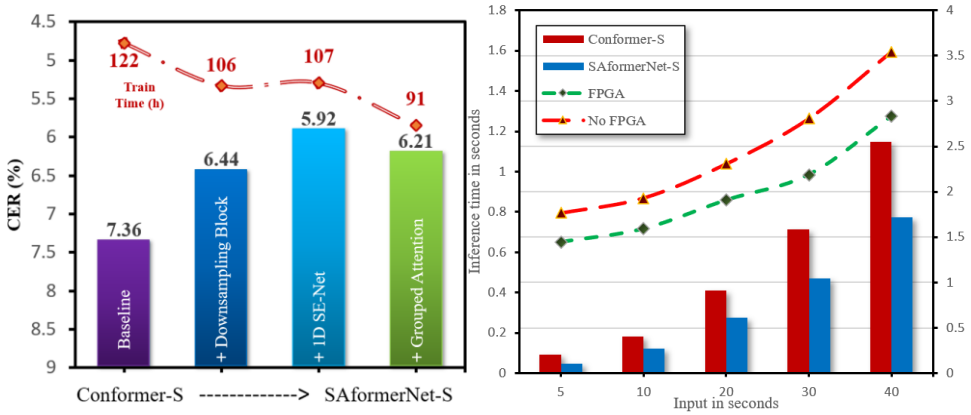
Additionally, as shown in Figure 11, complementary ablation studies are performed using Conformer-S as a baseline model to better understand the improvements brought about by the various improvement methods. The study initially examines the impact of downsampling on accuracy, multiply add operations, and training time reduction. It demonstrates that downsampling effectively decreases temporal redundancy in the feature representation of speech frames. The utilization of SE-Net into the convolution module improves accuracy, shows that feature recalibration can be adapted to the requirements of the network, compensates for missing information during downsampling, and better utilizes MHSA for global downsampling operations. The training time and inference time of the network is further reduced by introducing grouped attention in the network, showing that the use of grouped attention can reduce the complexity of the model. The architecture of the model was evaluated using Vivado 2018.2 on a Xilinx XCZU9EG FPGA, with a batch size set to 1. The utilization report is shown in Table 5.

Table 4. Comparison of SAformerNet with recently published models. At 13.4 M parameters, SAformerNet-S outperforms the baseline model Conformer-S by 2.1%/3.3% on the dev/test dataset of the AISHELL-1, and it is also significantly better than the other models in the Zhuang Language dataset. At 33.4 M model parameters, the model still has nice results. All model performance metrics are derived from our replicated best results.

Model Architecture	Model Type	CER (%)			Params (M)
		AISHELL-1		Zhuang Language Dataset	
		Dev	Test	Test	
LAS + SpecAugm	Seq2Seq	8.63	11.32	13.2	-
Conformer-S	CTC	5.62	5.95	7.36	13.0
Eff. Conformer-S	RNN-T	5.68	6.03	7.49	10.3
SAformerNet-S (ours)	CTC	5.69	6.13	6.21	13.4
w/o grouped Att	CTC	5.50	5.76	5.92	13.4
Conformer-M	CTC	5.40	5.67	6.99	30.5
Eff. Conformer-M	RNN-T	5.43	5.81	7.11	30.7
SAformerNet m (ours)	CTC	5.52	6.01	6.11	33.4
w/o grouped Att	CTC	5.36	5.58	5.67	33.4

Table 5. Xilinx's XCZU9EG FPGA is selected for verification. Utilization report for the hardware accelerator and its primary modules, and comparison between model's latency on FPGA and GPU.

Modules	Available Resources	Hardware Accelerator Modules				Average Resource Usage	GPU Latency	FPGA Latency
		BRAM 912	CLBs 548,160	LUT 274,080	DSP 2520			
MHSA	Matrix Operation	0	192,110	215,464	0	28.50%	1935.8 us	269.7 us
	Softmax	0	37,847	32,560	0	4.75%		
FFN	Layer-Norm	55	8475	14,230	209	5.25%	864.3 us	438.3 us
	Weight Memory	720	240	4696	0	20.19%		



(**a**) (**b**)

Figure 11. (**a**) Ablation study of design choices made for SAformerNet; (**b**) Inference time for model processing of long sequences and inference time for real deployment using FPGA acceleration.

SAformerNet improves the training speed by 26% and the average inference time by 29%, compared to the baseline model. In the real-world deployment use of the model with the FPGA acceleration, the overall average improvement in the result-to-terminal time for speech recognition is 20%.

5. Conclusions

As the window of ASEAN cooperation and communication in "the Belt and Road Initiative", the research and development of speech recognition for the Zhuang language can not only enrich the research of minority languages and cultures, but also contribute to the strengthening of communication between the Zhuang language and other languages in Guangxi and ASEAN countries. An online cloud-based platform was required to enhance linguistic communication. First, standard Zhuang news speeches were collected to build a database of 200 h of Zhuang speeches with annotations. Zhuang speech recognition was realized by an end-to-end automatic speech recognition neural network. Second, a machine translation model from Chinese to each country's language was established. Considering that edge-computing technology can reduce the transmission of a large amount of data, which can alleviate the pressure on network bandwidth. Finally, for the network's responsiveness to real-world needs, edge-computing techniques are applied to relieve network bandwidth pressure. An edge-computing private cloud system based on FPGA acceleration is proposed to improve model operation efficiency. The experiments show that the applicability of the model is verified, with satisfactory results in terms of accuracy and speed. SAformerNet-S outperforms the baseline model Conformer-S by 2.1%/3.3% on the dev/test dataset of the AISHELL-1 and has a recognition rate of 5.92% on the Zhuang language dataset. Compared to the baseline model, SAformerNet demonstrated a 26% improvement in training speed and a 29% average acceleration in inference time. In the practical implementation of the model, the computational delay for the MHSA and FFN modules has been reduced by 7.1 and 1.9 times, respectively, resulting in an overall average improvement in speech recognition result-to-terminal time of 20%. The scheme is suitable for small-scale remote natural language tasks in real-time applications. It serves as a prototype for a tool that supports remote Zhuang speech and multilingual conversational translation in mountainous areas.

In future studies, more emphasis will be placed on the practical application of the system, and improvements will continue to be made in response to new problems.

Author Contributions: Conceptualization, Z.F. and X.Z.; methodology, Z.F.; software, Z.F. and R.L.; validation, X.Z. and Z.B.; investigation, M.H. and X.L.; data curation, Z.F.; writing—original draft preparation, Z.F. and T.D.; writing—review and editing, Z.F. and X.Z.; visualization, X.L.; supervision, X.Z. and J.L.; project administration, X.Z. and Z.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Key Projects of Guangxi Province, grant number 2020AA21077007; the Innovation Project of Guangxi Graduate Education, grant number YCSW2022042; and the Guangxi New Engineering Research and Practice Project, grant number XGK2022003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: ASPEC, Asian Scientific Paper Excerpt Corpus, is constructed by the Japan Science and Technology Agency (JST) in collaboration with the National Institute of Information and Communications Technology (NICT); AISHELL-1, Part of the Hill Shell Chinese Mandarin open-source speech database. The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to express appreciation to the editors and reviewers for their valuable comments and suggestions. Relevant teachers from the School of Foreign Languages of Guangxi University are greatly appreciated for their valuable comments and suggestions. This work is supported by Guangxi Key Laboratory of Multimedia Communications and Network Technology, Guangxi, China, School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, China.

Conflicts of Interest: Author Xuejun Zhang was employed by the company Guangxi Big White & Little Black Robots Co., Ltd., Nanning 530007, China. The remaining authors declare that the research

was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Grey, A. *Language Rights in a Changing China: A National Overview and Zhuang Case Study*; Walter de Gruyter GmbH & Co KG: Berlin, Germany, 2021.
2. A Review of the Relationship and Comparative Research between Zhuang and Chinese Language—Part 7 of the Zhuang Language Research Series. *Inheritance* **2014**, *3*, 124–125.
3. Min, L. *Brief Records of Dong Language*; Ethnic Publishing House: Beijing, China, 1980.
4. Analysis of the Current Situation of Translation Studies in Minority language in China. *Foreign Lang. Teach. Res.* **2015**, *1*, 130–140.
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
7. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, 30.
9. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
10. Zhai, X.; Kolesnikov, A.; Houlsby, N.; Beyer, L. Scaling vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12104–12113.
11. 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.
12. Irshad, I.; Yasmin, M. Feminism and literary translation: A systematic review. *Heliyon* **2022**, *8*, e09082. [CrossRef] [PubMed]
13. Comelles, E.; Atserias, J. VERTa: A linguistic approach to automatic machine translation evaluation. *Lang. Resour. Eval.* **2019**, *53*, 57–86. [CrossRef]
14. Chauhan, S.; Daniel, P. A comprehensive survey on various fully automatic machine translation evaluation metrics. *Neural Process. Lett.* **2022**, 1–55. [CrossRef]
15. Reiter, E. A structured review of the validity of BLEU. *Comput. Linguist.* **2018**, *44*, 393–401. [CrossRef]
16. Guzmán, F.; Joty, S.; Màrquez, L.; Nakov, P. Machine translation evaluation with neural networks. *Comput. Speech Lang.* **2017**, *45*, 180–200. [CrossRef]
17. Kim, S.; Gholami, A.; Yao, Z.; Mahoney, M.; Keutzer, K. I-bert: Integer-only bert quantization. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 5506–5518.
18. Yu, J.; Park, J.; Park, S.; Kim, M.; Lee, S.; Lee, D.; Choi, J. Nn-lut: Neural approximation of non-linear operations for efficient transformer inference. In Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 10–14 July 2022; pp. 577–582.
19. Lu, S.; Wang, M.; Liang, S.; Lin, J.; Wang, Z. Hardware accelerator for multi-head attention and position-wise feed-forward in the transformer. In Proceedings of the 2020 IEEE 33rd International System-on-Chip Conference (SOCC), Virtual, 8–11 September 2020; pp. 84–89.
20. Ye, W.; Zhou, X.; Zhou, J.T.; Chen, C.; Li, K. Accelerating attention mechanism on fpgas based on efficient reconfigurable systolic array. In *ACM Transactions on Embedded Computing Systems (TECS)*; Association for Computing Machinery: New York, NY, USA, 2022.
21. Wang, H.; Zhang, Z.; Han, S. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In Proceedings of the 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Seoul, Republic of Korea, 27 February–3 March 2021; pp. 97–110.
22. Zhang, X.; Wu, Y.; Zhou, P.; Tang, X.; Hu, J. *Algorithm-Hardware Co-Design of Attention Mechanism on Fpga Devices. ACM Transactions on Embedded Computing Systems (TECS)*; Association for Computing Machinery: New York, NY, USA, 2021; Volume 20, pp. 1–24.
23. Burchi, M.; Vielzeuf, V. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Cartagena, Colombia, 13–17 December 2021; pp. 8–15.
24. Kim, S.; Gholami, A.; Shaw, A.; Lee, N.; Mangalam, K.; Malik, J.; Mahoney, M.; Keutzer, K. Squeezeformer: An efficient transformer for automatic speech recognition. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9361–9373.
25. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
26. Perslev, M.; Jensen, M.; Darkner, S.; Jennum, P.; Igel, C. U-time: A fully convolutional network for time series segmentation applied to sleep staging. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; p. 32.

27. Jiang, Y.; Yu, J.; Yang, W.; Zhang, B.; Wang, Y. Nextformer: A convnext augmented conformer for end-to-end speech recognition. *arXiv* **2022**, arXiv:2206.14747.
28. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T. Mass: Masked sequence to sequence pre-training for language generation. *arXiv* **2019**, arXiv:1905.02450.
29. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
30. Sennrich, R.; Birch, A.; Currey, A.; Germann, U.; Haddow, B.; Heafield, K.; Barone, A.; Williams, P. The University of Edinburgh's neural MT systems for WMT17. *arXiv* **2017**, arXiv:1708.00726.
31. Currey, A.; Miceli-Barone, A.V.; Heafield, K. Copied monolingual data improves low-resource neural machine translation. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; pp. 148–156.
32. Ham, T.J.; Jung, S.J.; Kim, S.; Oh, Y.; Park, Y.; Song, Y.; Park, J.; Lee, S.; Park, K.; Lee, J.; et al. A^3: Accelerating attention mechanisms in neural networks with approximation. In Proceedings of the 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), San Diego, CA, USA, 22–26 February 2020; pp. 328–341.
33. Li, B.; Pandey, S.; Fang, H.; Lyu, Y.; Li, J.; Chen, J.; Xie, M.; Wan, L.; Liu, H.; Ding, C. Ftrans: Energy-efficient acceleration of transformers using fpga. In Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design, Boston, MA, USA, 10–12 August 2020; pp. 175–180.
34. Brown, P.F.; Cocke, J.; Della Pietra, S.A.; Della Pietra, V.J.; Jelinek, F.; Lafferty, J.D.; Mercer, R.L.; Roosin, P.S. A statistical approach to machine translation. *Comput. Linguist.* **1990**, *16*, 79–85.
35. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
36. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, ICIR 2015, San Diego, CA, USA, 7–9 May 2015.
37. Shao, C.; Feng, Y.; Zhang, J.; Meng, F.; Chen, X.; Zhou, J. Retrieving sequential information for non-autoregressive neural machine translation. *arXiv* **2019**, arXiv:1906.09444.
38. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://api.semanticscholar.org/ (accessed on 20 September 2023).
39. Libovický, J.; Helcl, J. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. *arXiv* **2018**, arXiv:1811.04719.
40. Fadaee, M.; Bisazza, A.; Monz, C. Data augmentation for low-resource neural machine translation. *arXiv* **2017**, arXiv:1705.00440.
41. Wang, X.; Pham, H.; Dai, Z.; Neubig, G. SwitchOut: An efficient data augmentation algorithm for neural machine translation. *arXiv* **2018**, arXiv:1808.07512.
42. Zhou, J.; Keung, P. Improving non-autoregressive neural machine translation with monolingual data. *arXiv* **2020**, arXiv:2005.00932.
43. Xia, M.; Kong, X.; Anastasopoulos, A.; Neubig, G. Generalized data augmentation for low-resource translation. *arXiv* **2019**, arXiv:1906.03785.
44. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
45. Roy, A.G.; Navab, N.; Wachinger, C. Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, 16–20 September 2018; pp. 421–429.
46. Wang, X.; Ning, Z.; Guo, L.; Guo, S.; Gao, X.; Wang, G. Online learning for distributed computation offloading in wireless powered mobile edge computing networks. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 1841–1855. [CrossRef]
47. Premsankar, G.; Di Francesco, M.; Taleb, T. Edge computing for the Internet of Things: A case study. *IEEE Internet Things J.* **2018**, *5*, 1275–1284. [CrossRef]
48. Park, D.S.; Zhang, Y.; Chiu, C.C.; Chen, Y.; Li, B.; Chan, W.; Le, Q.V.; Wu, Y. Specaugment on large scale datasets. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6879–6883.
49. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv* **2019**, arXiv:1904.08779.
50. Bu, H.; Du, J.; Na, X.; Wu, B.; Zheng, H. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In Proceedings of the 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA), Seoul, Republic of Korea, 1–3 November 2017; pp. 1–5.
51. Nakazawa, T.; Yaguchi, M.; Uchimoto, K.; Utiyama, M.; Sumita, E.; Kurohashi, S.; Isahara, H. ASPEC: Asian scientific paper excerpt corpus. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 2204–2208.