applied sciences

MDPI

Article

Multi-Scale Feature Learning for Language Identification of Overlapped Speech

Zuhragvl Aysa, Mijit Ablimit * and Askar Hamdulla

College of Information Science and Engineering, Xinjiang University, Urumqi 830017, China
* Correspondence: mijit@xju.edu.cn; Tel.: +86-133-0991-2366

Abstract: Language identification is the front end of multilingual speech-processing tasks. The study aims to enhance the accuracy of language identification in complex acoustic environments by proposing a multi-scale feature extraction method. This method replaces the baseline feature extraction network with a multi-scale feature extraction network (SE-Res2Net-CBAM-BILSTM) to extract multi-scale features. A multilingual cocktail party dataset was simulated, and comparative experiments were conducted with various models. The experimental results show that the proposed model achieved language identification accuracies of 97.6% for an Oriental language dataset and 75% for a multilingual cocktail party dataset Furthermore, comparative experiments show that our model outperformed three other models in the accuracy, recall, and F1 values. Finally, a comparison of different loss functions shows that the model performance was better when using focal loss.

Keywords: language identification; spectrogram; overlapped speech; CNN; CBAM; SE-Res2Net

Citation: Aysa, Z.; Ablimit, M.; Hamdulla, A. Multi-Scale Feature Learning for Language Identification of Overlapped Speech. *Appl. Sci.* **2023**, *13*, 4235. https://doi.org/10.3390/app13074235

Academic Editor: Douglas O'Shaughnessy

Received: 15 February 2023
Revised: 21 March 2023
Accepted: 23 March 2023
Published: 27 March 2023

1. Introduction

Language identification (LID) is the process of identifying the language type of a given speech segment [1], and it is a classification task [2]. The overall architecture includes a feature extraction task and a classification task. So, the system can be feature-intensive, as [3] represented by training large-scale data, or it can be model-intensive, as obtained by better classifier models. Both the feature extraction and classification tasks are equally important, but a balance between them is optimal [4], as described in this paper.

Language identification technology has developed considerably in the last decade. New deep learning frameworks provide new opportunities for the development of language identification research [5]. Recently, people have become increasingly concerned about language identification in real and complex scenes. As the core front-end processing module for multilingual intelligent speech processing tasks, language identification can be used in multiple fields, such as automatic speech recognition, speech translation, and speech generation. In noisy multilingual overlapping speech, even the human ear may not be able to accurately identify contents, the clean single speech-based model is not efficient for overlapped voices. Therefore, it is necessary to separate overlapped speeches before identifying or understanding the contents.

This study proposes language identification feature extraction technology based on a squeeze–excitation [6] and multi-scale residual [7] network (SE-Res2Net), which improves the feature extraction method of the baseline model and greatly improves the recognition performance of the original language identification algorithm. Experiments were conducted on the AP17-OLR [8] dataset and a multilingual cocktail party dataset. The accuracy, recall, and F1 values were improved over the baseline model, and the robustness of the model was also improved compared to other models.

2. Related Work

As a front-end technology for speech signal processing, language identification plays a vital role in speech recognition and other related fields, mainly speech translation, pub-

lic safety, and multilingual dialogue systems [9]. Language identification is a typical classification problem, and different features can have different influences. As a result, the corresponding model for each language is trained and saved based on the appropriate algorithm [10]. In the recognition process, the features are first extracted and fed into the classification model, and the language type of the speech signal is determined based on the similarities [11]. Traditional acoustic models, such as the Gaussian mixture model–universal background model (GMM-UBM) [12], the hidden Markov model (HMM), etc. [13], were used for language identification, but these often require an extensive number of training parameters to capture the feature space's complexity. To address this issue, Campbell et al. [14] employed the SVM algorithm to classify the GMM mean supervector of speech (GMM-SVM). Language scholars also developed a language identification method that utilizes i-vectors [15]. This approach involves obtaining i-vectors from speech and employing a back-end discrimination algorithm for language identification. By utilizing this method, the complexity of multilingual modeling can be reduced while still achieving exceptional performance.

In recent years, researchers used deep learning to extract the deep bottleneck features (DBF) [16] of speech signals. This method used i-vectors instead of acoustic features and GMM-UBM to capture language information more effectively. However, it also increases the complexity of the model. Researchers have subsequently proposed end-to-end language identification systems based on different neural network architectures. Firstly, Lopez-Moreno et al. [5] applied deep neural networks (DNNs) to short-time language identification. Gonzalez-Dominguez et al. [17] proposed a long and short-term memory recurrent neural network (LSTM-RNN) for automatic language identification, which effectively solved the problem of gradient disappearance in RNNs. Still, the model was complex and time-consuming. Fernando et al. [18] built an end-to-end language identification system based on bidirectional LSTM (BiLSTM), which effectively takes into account the past and future information of speech. Padi et al. [19] used a bidirectional gated recurrent unit (GRU) network for multi-categorical language identification, which has a more straightforward structure and improved recognition rate compared to the Bi-LSTM network. Another popular approach for language identification is a convolutional neural network (CNN) [20], which extracts local features from speech signals and enhances language identification. CNNs are trained on the spectral map of the raw audio signal. This method involves end-to-end learning with minimal pre-processing, as the neural network can directly map the original data to the final output without relying on the traditional machine learning pipeline.

In 2016, Wang et al. [21] applied an attention mechanism model to language identification systems. This mechanism selects the most relevant speech features for language identification and improves the recognition performance of the network. In 2017, Bartz et al. [22] combined a convolutional network with a recurrent neural network (CRNN) for language identification and proposed a CNN-BiLSTM network [23], which achieved higher accuracy. Although BiLSTM does improve the recognition accuracy, it has some problems, such as an inability to parallelize operations and poor modeling of the effects of hierarchical information. To address these issues, Romero et al. in 2021 [24] proposed an encoder approach based on the "transformer architecture" applied to the language identification task using speech-directed information. In the same year, H Yu et al. applied the unsupervised learning speech pre-training method [25] to the language identification system. In 2022 [26], Nie Y et al. proposed a BERT-based language identification system (BERT-LID) to improve language identification performance, especially on short speech segments. Recently, target language extraction has been introduced as a new task [27], which treats the cocktail party problem as a multilingual scenario that separates all of the voices of people speaking in the target language from the rest of the voices at once. A recent study [28] extended this task to multiple target languages and extracted all the speech signals as one specific language. Based on previous research, another study [29] proposed the blind language separation

task, which separates overlapping speech by language. These methods can be clearly seen in Table 1 below.

Table 1. Characteristics of different language identification technologies.

Proposed Algorithm	Features
GMM-UBM [12]	The method of combining SDC features with a Gaussian mixture model–general background model is proposed, but it requires a large number of parameters, and the training data are usually insufficient.
GMM-SVM [14]	SVM algorithm is proposed to classify the GMM mean super-vector of speech.
i-Vector [15]	Language identification using i-vector obtained from speech combined with a back-end discrimination algorithm
DBF [16]	Replaces the traditional GMM-UBM combined with acoustic features, which can effectively characterize linguistic information and make language more easily distinguishable.
DNN [5]	First proposed application of deep neural networks to language identification tasks.
LSTM-RNN [17]	This method effectively solved the problem of gradient disappearance in RNNs.
BiLSTM [18]	This method effectively takes into account the past and future information of speech.
GRU [19]	GRU has a more straightforward structure and improved recognition rate compared to the Bi-LSTM network.
CNN [20]	CNN extracts local features of speech signals and effectively improves language identification.
Attention mechanism [21]	Through the attention mechanism, the more valuable information for language identification in speech features is obtained, and the effect of language identification is improved.
CRNN [22]	The combined network extracts richer speech feature information, thus improving the accuracy of language identification.
Transformer [24]	Transformer architecture can perform parallel computation and extract deeper and richer feature information.
Pre-trained [25]	The pre-trained model can obtain better discriminative representation and make full use of unsupervised data.
BERT-based [26]	The study extended the original BERT model by taking the phonetic posterior grams (PPGs) derived from the front-end phone recognizer as input.

In this study, we took the CNN-CBAM-BiLSTM [30] network based on the dual attention mechanism convolutional block attention module (CBAM) as the baseline model and improved it with the SE-Res2Net network by proposing a multi-scale language identification method. This study constructed a multilingual cocktail party dataset to simulate a multilingual cocktail party scene considering the complex acoustic scenarios in real life, and comparative experiments were also conducted on this dataset with different models.

3. Algorithm Model Structure

3.1. Algorithm Module

3.1.1. SE-Res2Net-CBAM-BiLSTM Network

This study proposes a SE-Res2Net-CBAM-BiLSTM language identification method that combines the compressed excitation multi-scale residual (SE-Res2Net) module and CBAM. The SE-Res2Net module can extract features in parallel through multiple branches

and adaptively weigh different channels so as to improve the performance and generalization ability of the network. These advantages can help the SE-Res2Net-CBAM-BiLSTM network to better understand and extract image features so as to achieve better performance in language identification tasks. The network block diagram is shown in Figure 1.

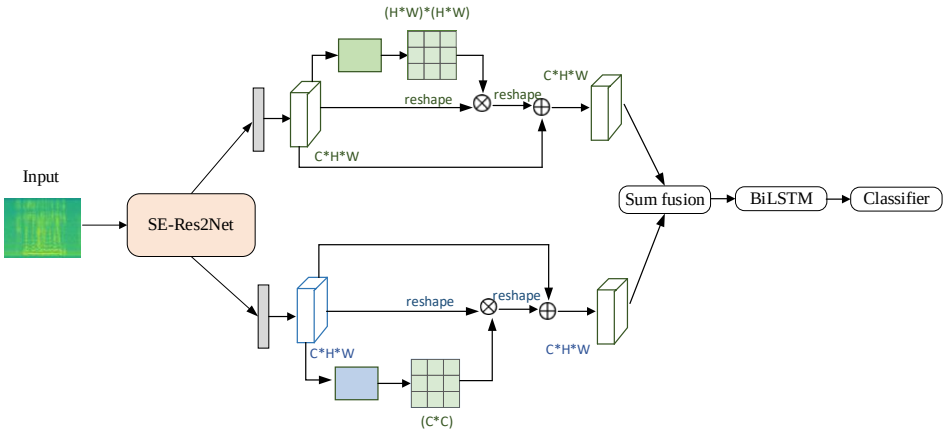


Figure 1. Block diagram of the SE-Res2Net-CBAM-BiLSTM language identification model structure.

Specifically, on the basis of the CBAM-BiLSTM network, the SE-Res2Net multi-scale feature extraction module was added, and a language identification network with a multi-scale mechanism was built to improve the network's feature expression ability. First, feature extraction is performed to convert speech into corresponding spectral features, and then the spectral features are used as input to extract multi-scale features through the SE module and Res2Net module. In the multi-scale extraction module, the network's description ability is further enhanced, and the feature representation is clearer. Then, it is sent to the CBAM module to assign different weights to the channel and spatial dimensions of the feature map so as to derive features useful for language identification. Finally, the output of the CBAM module is sent to the tile layer through the full connection layer, and finally, the language classification is realized through the softmax classifier.

3.1.2. Feature Extraction

Sound waves are produced by the vibration of the vocal cords. A speech signal is a time-domain signal that changes in time and amplitude. It can be seen as the sum of periodic signals with different frequencies. It is usually transformed from the time domain to the frequency domain by Fourier transform for analysis [31]. In order to better represent the characteristics of the sound, the acoustic signal needs to be converted into a computer-recognizable form of acoustic feature vectors. Commonly used acoustic feature extraction methods include the mel frequency cepstral coefficient (MFCC), frequency domain features (FBANK), the linear predictive cepstral coefficient (LPCC), etc. [32]. This study adopted the method of extracting the features of the speech spectrogram through experiments. The spectrogram can retain the feature information of the original speech signal more completely, which conforms to the characteristics of human hearing. A schematic diagram of the language spectrum is shown in Figure 2.

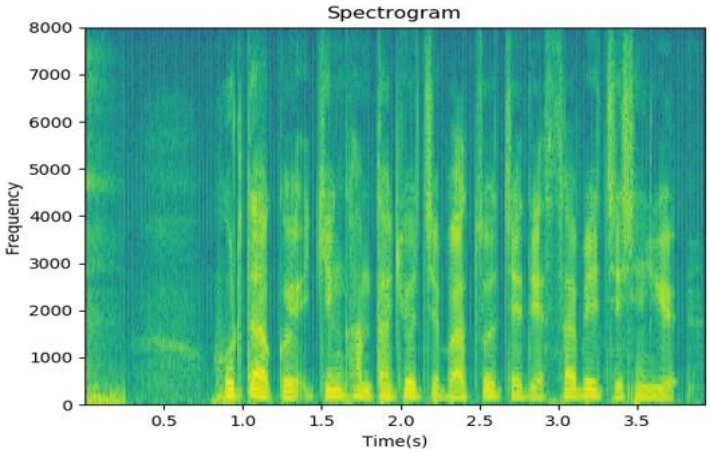


Figure 2. Structure of the language map.

As seen in the speech spectrum diagram, there are horizontal lines, vertical lines, and scrambled patterns, each of which has a different representation. The horizontal line is a resonance peak, the vertical line is a fundamental tone in the speech signal, and the clutter's depth indicates the noise energy distribution. The frequency and bandwidth of the horizontal lines can determine the frequency and bandwidth of the corresponding resonance peaks. The existence of horizontal lines in an audio signal's speech spectrum indicates whether it is a turbid tone. The vertical bars are perpendicular to the time axis, and each one represents a fundamental tone. The speech spectrogram shows the basic information of the speech signal and can be studied with image processing methods.

3.1.3. Residual Network

The residual neural network (ResNet) [33] was proposed by Kai-Ming He, Xiang-Yu Zhang, Shao-Qing Ren, and Jian Sun at Microsoft Research. ResNet uses residual blocks to improve the traditional convolution neural network, the network structure is shown in Figure 3. In the traditional network, each layer has a series of convolution, activation, and pooling operations, and the input of each layer is the output of the previous layer. In the residual block, there are two branches: one is identity mapping, and the other is nonlinear mapping. The identity mapping passes the input to the output directly. The nonlinear mapping transforms the input into a residual through convolution, activation function, and pooling. Then it adds the residual and input to obtain the output. The residual block can learn the difference between input and output and fit the data better.

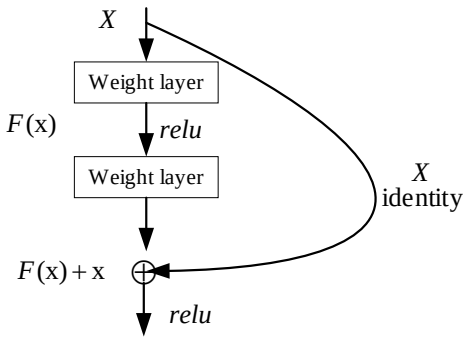


Figure 3. ResNet network structure diagram.

As can be seen from the above diagram: x is the input of the residual block, which is then copied into two parts, one of which is fed into a weight layer for inter-layer operations (equivalent to feeding x into a function for mapping), resulting in $F(x)$. The other part is used as a branching structure, and the output is still the original x. Finally, the outputs

of the two parts are superimposed ($F(x) + x$) and then processed through the activation function. This is the basic structure of the entire residual block. $F(x) = y - x$ is also called the residual term, and it is easier to obtain the $-x > y$ mapping close to a constant mapping, e.g., by learning the residual term $F(x)$ to zero than it is by stacking the neural network layers directly. Using the residual structure allows the network to be deeper, converge faster, and optimize more easily, while having fewer parameters and less complexity compared to previous models. This residual structure solves the degradation problem of deep networks that are difficult to train. Applicable to a wide range of computer vision tasks, the entire residual structure can be defined formally as $y = F(x, \{W_i\}) + x$, where $F(x, \{W_i\})$ refers to the fitted residual mapping. In the figure above, there are two fully connected layers, i.e., $F = W_2\sigma(W_1x)$, where δ refers to the non-linear activation function ReLU. When F and x are of the same dimension, they can be added directly element by element, but if they are different, it is necessary to add another linear mapping to x, mapping it to a vector of the same dimension as F. At this point, the whole residual structure is $y = F(x, \{W_i\} + W_sx)$, where W_s is a matrix used for dimension matching.

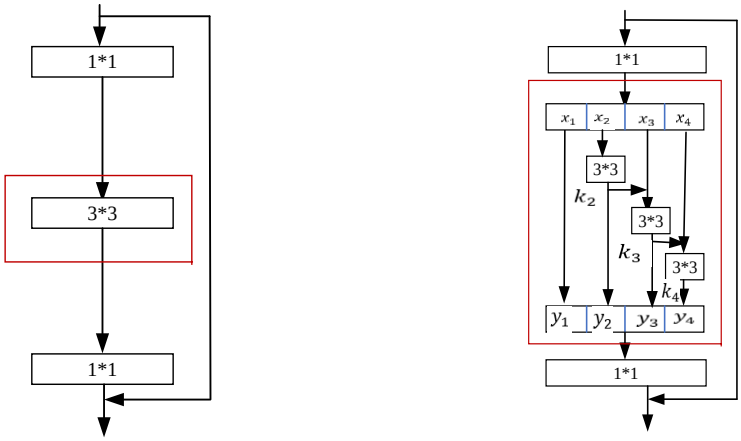
3.1.4. Res2Net Module

Nankai University proposed a new way of constructing convolutional networks Res2Net [7], which is achieved by constructing hierarchical connections inside a single residual block. Res2Net is the original ResNet with the middle 3×3 convolution replaced by the red part on the right, which is at least directly connected without going through 3×3 convolutions. At most, it will go through three 3×3 convolutions, which is the reason it seems to be wilder than the original structure. This is shown in Figure 4a,b below. After a 1×1 convolution, we partition the feature mapping uniformly into subsets of feature mappings, denoted by x_i, where $i \in \{1, 2, \ldots, s\}$. Compared to the input features, each feature submap X_i has the same spatial size but one-third of the number of channels, excluding x_1, and each x_i has a corresponding 3×3 convolutional transformation, denoted by $K_i()$. The outputs of the feature subgraphs $K_i()$ and X_i are summed and fed to $K_{i-1}()$. Thus, y_i can be written as:

$$y_i = \begin{cases} x_i & i = 1; \\ K_i(x_i) & i = 2; \\ K_i(x_i + y_{i-1}) & 2 < i \leq s. \end{cases} \quad (1)$$

It is worth noting that each 3×3 convolution operator $K_i()$ may receive feature information from all feature partitions $\{x_j, j \leq i\}$. Each time a feature is decomposed by a 3×3 convolution operator, the output may have a receptive field larger than x_j. Due to the combinatorial explosion effect, the output of the Res2Net module contains different numbers and different combinations of receptive field sizes/scales. In the Res2Net module, the decomposition is processed in a multi-scale manner, which facilitates the extraction of global and local information. To better fuse information at different scales, we combine all splits together and pass them through a 1×1 convolution. The splitting and cascading strategy allows for more efficient forced convolution to enhance processing. To reduce the number of parameters, we omit the convolution of the first split, which can also be considered a form of feature reuse. In this work, s was used as the control parameter for the scale size.

The Res2Net module achieves better feature extraction and higher classification accuracy by adding multiple branches with different scales and adopting multi-scale feature fusion. Multi-scale refers to multiple available receptive fields with finer granularity.



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

Figure 4. ResNet and Res2net network infrastructure diagram. (**a**) ResNet infrastructure diagram; (**b**) Res2Net infrastructure diagram.

3.1.5. SE-Res2Net Module

In 2020, J. Hu et al. [6] proposed the squeeze-and-excitation network (SENet) to obtain different weights on the channel dimension of the feature map to highlight important features and ignore useless features. Figure 5 shows the structure of the SE module network.

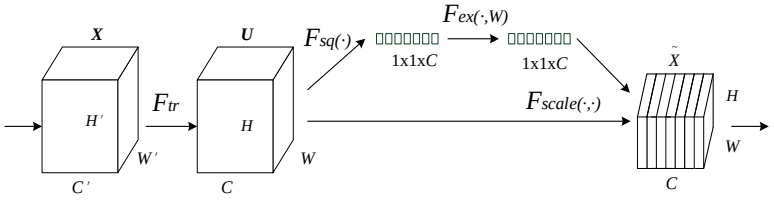


Figure 5. SE module structure.

The diagram above shows the network structure of the SE module. Given an input X with a feature channel count of C, a feature with a feature channel count of C is obtained by a series of general transformations. The previously obtained features are also rescaled by the following three operations: (1) The squeeze operation reduces the features to a single number per channel by compressing them spatially. This gives a global view of each channel. The output has the same dimension as the input channels. (2) The excitation operation is a gating mechanism similar to that in recurrent neural networks. The weights of each feature channel are generated by a parameter *W*, where the parameter w is learned to explicitly model the correlation between the feature channels. (3) The scaling operation is where the output of the weight by excitation is considered as the importance of each feature channel after feature selection. The weights of each feature channel are then multiplied one by one with the previous features to complete the original features and reconstructed in the channel dimension.

To prevent channel grouping from losing inter-channel correlation, the output y of the Res2Net module is fed into the SE module. The network structure is shown in Figure 6. In this module, the features $y \in R^{w \times h \times b \times c}$ are first compressed into $y^{'} \in R^{1 \times 1 \times 1 \times c}$ using global average pooling. Then, the correlation between channels is fitted using fully connected layers and finally normalized using a sigmoid activation function. Thus, the weight vector of the channels is $f^c = \sigma(FC(\delta(FC(y'))))$, where FC denotes the fully connected layer, σ denotes the ReLU function, and δ denotes the sigmoid function. The output of the SE module is $f = f' + x$. The rescaling of the original features in the channel dimension is implemented inside the residual unit to complete the feature adjustment.

finally, the input x of the residual unit is connected to the output f' of the residual unit by means of a jump connection to obtain the output of the SE-Res2Net module as $f' = f^c \cdot y$. The fusion of the SE module after the 1×1 convolution allows the advantages of the SE module to be enhanced by reassigning different weights to the channel features, eliminating invalid features, and allowing the single-layer features to be used to maximum effect. The SE-Res2Net module designed in this paper can emphasize the residual mapping, promote the convergence of the network, and improve the stability of the model.

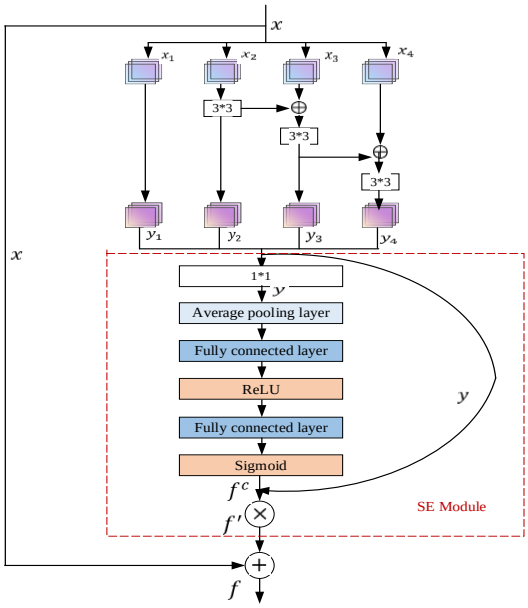


Figure 6. SE-Res2Net network structure diagram.

3.1.6. Bidirectional Long- and Short-Term Memory Network (BiLSTM)

The BiLSTM network consists of a forward and a backward LSTM, a bi-directional network structure. The forward LSTM learns messages before the current moment, and the backward LSTM learns messages after the current moment, so the network can learn the temporal background information contained in the speech sequence, thus making up for the shortcomings of CNN networks. The BiLSTM network [34] consists of four parts: the input layer, the forward LSTM, the backward LSTM, and the output layer, and its network structure is shown in Figure 7.

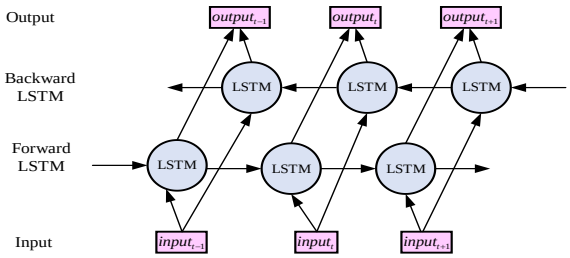


Figure 7. BiLSTM network structure.

In the BiLSTM network structure in Figure 4, $input_{t-1}$, $input_t$, and $input_{t+1}$ denote the inputs at moments $t-1$, t, and $t+1$, respectively, and $output_{t-1}$, $output_t$, and $output_{t+1}$ denote the outputs corresponding to moments $t-1$, t, and $t+1$, respectively. The forward LSTM refers to the calculation of the output corresponding to the forward moments along the forward order of the moments. Backward LSTM means calculating the output corresponding to the reverse moment along the reverse order of moments, and finally, calculating the output of both together as the final output at the corresponding moment.

3.1.7. Convolutional Block Attention Module (CBAM)

We use the CBAM [35] attention layer after SE-res2Net to focus on the features related to language and generate distinctive feature representations for language identification. The advantage is that we use attention to measure how important each high-level feature is for the language difference instead of simply aggregating a bunch of features over time.

The attention layer is located after the bidirectional LSTM. The output of the bidirectional LSTM is first passed through a softmax function to calculate the normalized weight α_t, calculated as shown in (2). The normalized weight α_t is then weighted and summed over h_t to obtain the language representation c, as shown in (3).

$$\alpha_t = \frac{\exp(W \cdot h_t)}{\sum_{t=1}^{T} \exp(W \cdot h_t)} \tag{2}$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{3}$$

where W denotes the weight value, h_t is the state at the current moment, and the language representation c is derived by computing Equations (2) and (3) and then passed to the all-connected layer to obtain a more profound representation of the language. A softmax classifier maps the language representation to N different spaces for classification, where N denotes the number of classes of the language.

The general network architecture of the convolutional chunk attention module is shown in Figure 8.

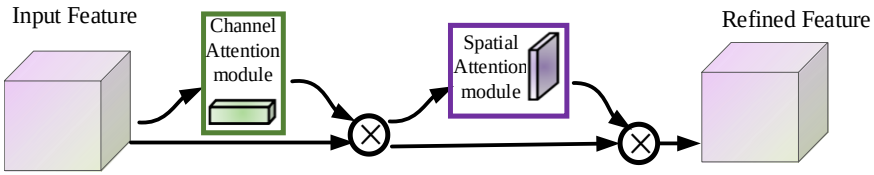


Figure 8. Convolutional block attention module.

Given a feature map, the CBAM can serially generate attentional feature map information in both the channel and spatial dimensions, and then the information from the two feature maps is multiplied with the previous original input feature map for adaptive feature correction to produce the final feature map.

As shown in the figure above, there is an input, a channel attention module, a spatial attention module, and an output. The input features $F \in R^{C \times H \times W}$, followed by the channel attention module $M_c \in R^{C \times 1 \times 1}$, multiply the result of the convolution by the original image, and the output of the channel attention module is used as input for the two-dimensional convolution of the spatial attention module $M_s \in R^{1 \times H \times W}$, and then the output is multiplied by the original image.

$$F' = M_c(F) \otimes F \tag{4}$$

$$F'' = M_s(F') \otimes F' \tag{5}$$

Equation (4) focuses on the features of the channel by keeping the channel dimension constant and compressing the spatial dimension, focusing on the meaningful information in the input image. Moreover, Equation (5) focuses on the features in the space by keeping the spatial dimension constant, compressing the channel dimension, and focusing on the location information of the target.

3.2. Loss Function

Generally, to solve the category imbalance problem, a weighting factor $\alpha \in [0, 1]$ is added before each category in the loss function to reconcile the category imbalance. Defining a in a similar way using p yields a binary balanced cross-entropy loss function.

$$CE(p_t) = -\alpha_t \log(p_t) \tag{6}$$

A larger imbalance between classes then results in the cross-entropy loss receiving an impact during training. Losses from the misclassification of easily classified samples account for the vast majority of the overall loss and dominate the gradient. Focal loss [36] adds a moderator to the balanced cross-entropy loss function to reduce the weight of easily classified samples, focusing on the training of difficult samples, as defined below.

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{7}$$

where $(1 - p_t)^{\gamma}$ is the modulation factor and $\gamma \geq 0$ is the adjustable focus parameter.

4. Experiments and Discussion

4.1. Dataset

① The Oriental language dataset used in this paper was provided by the AP17-OLR competition [8]. In this paper, we used the following five languages for our experiments: Mandarin (zh-cn), Vietnamese (vi-vn), Indonesian (id-id), Japanese (ja-jp), and Korean (ko-kr). For each language, 1800 speech data were extracted and divided into a dataset with a ratio of 7:2:1 (training set/validation set/test set), and the structure of the dataset is listed in Table 2.

Table 2. Oriental language dataset structure.

Language	Train	Validation	Test	Total
zh-cn	1260	360	180	1800
id-id	1260	360	180	1800
ja-jp	1260	360	180	1800
ko-kr	1260	360	180	1800
vi-vn	1260	360	180	1800

② Multilingual cocktail party dataset: We created a multilingual cocktail party dataset based on the Oriental language dataset to simulate a real multilingual overlapping speech scene. The process was as follows: First, we cut the original data into 4s segments. Second, we randomly selected some speech for each language and split it into target and non-target speakers. Third, we mixed them with different overlap rates according to the scenario needs. The result was a multilingual cocktail party dataset. For the language identification task, we assigned numerical labels to each language, such as "0" for "id-id", "1" for "ja-jp", and so on.

4.2. Network Parameters

The experiments in this study were conducted in a Linux environment using Python as the programming language, Pytorch as the deep learning framework, and CUDA version 11.4 on an NVIDIA GeForce GTX 3090 GPU and Intel(R) Xeon(R) Gold 6128 CPU @ 3.40 GHz.

During the training and validation phases, the language identification network model had a 224 × 224 spectral map as the input, a batch size of 32 × 32, and a learning rate of 0.0001, and we used the Adam optimizer and cross-entropy loss function. In each convolutional layer, the sizes and numbers of convolutional kernels were (7 × 7.16), (5 × 5.32), (3 × 3.32), and (3 × 3.32), respectively. The span of the convolution kernels was 1 and the

padding was 0. The size of the convolution kernels in the pooling layer was 3, the span was 2, and the padding was 0.

4.3. Performance Evaluation

The performance evaluation metrics used in the experiments were the accuracy, precision, recall, and F1 score values. When predicting a piece of speech, four statistical results will appear, namely, the target language is judged as the target language (TP), the target language is judged as the non-target language (FN), the non-target language is judged as the target language (TN), and the non-target language is judged as the non-target language (TN), as shown in Table 3.

Table 3. Results of statistical identification.

Confusion Matrix for Target and Non-Target Languages		Predicted Value		Total
		Target Languages	Non-Target Languages	
True value	Target languages	TP	FP	P
	Non-target languages	TN	FN	N
Total		T	F	-

According to Table 3, the calculation formulas of the accuracy, precision, recall, and F1 values can be obtained, as shown below.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

4.4. Experimental Results and Analysis

In this section, we used the SE-Res2-Net-CBAM-BiLSTM language identification model proposed in this paper to carry out experiments on the Oriental Corpus and our own multilingual cocktail party corpus. There were three experimental tasks.

Task 1: Three different models were selected and compared using the language spectrogram as input features, and their experimental results are shown in Figure 9.

As can be seen in Figure 9, for the four different models, the SE-Res2Net-CBAM-BiLSTM model had higher values than the other three models for the evaluation metrics corresponding to the AP17-OLR dataset, with an accuracy of 97.64%. The results in this paper show an accuracy improvement of about 3% compared to the baseline model CNN-CBAM-BiLSTM. This is because we replaced the CNN network with the SE-Res2Net network module in the baseline model. The SE-Res2Net network is a new multi-scale backbone network structure that can represent multi-scale features at a fine-grained level and increase the receptive field range of each network layer. These advantages can help SE-Res2Net-CBAM-BiLSTM networks to better understand and extract features so as to achieve better performance in language identification tasks.

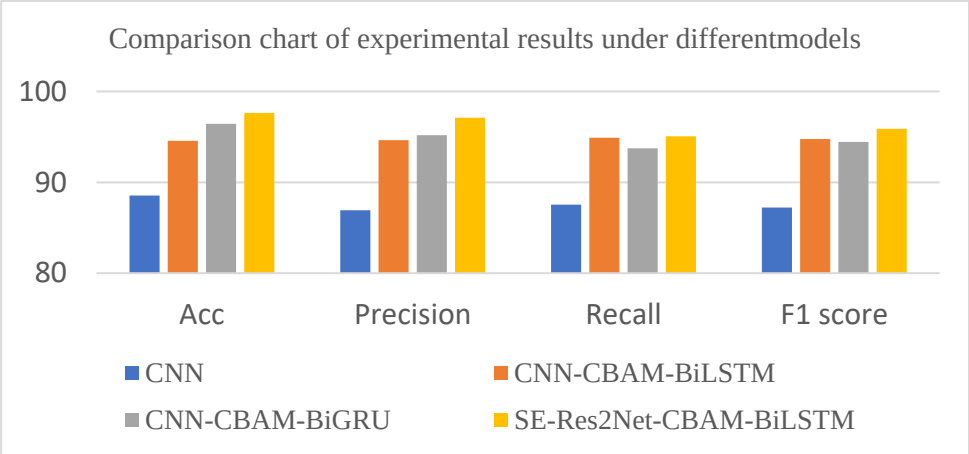


Figure 9. Comparison of experimental results under different models.

We conducted experiments on the CNN-CBAM-BiLSTM model and the SE-Res2Net-CBAM-BiLSTM model using an Oriental language dataset. Figures 10 and 11 show the accuracy and loss changes in the training and validation sets. Figure 10 shows the CNN-CBAM-BiLSTM model's curves, and Figure 11 shows the SE-Res2Net-CBAM-BiLSTM model's curves. The blue curves are for the training set, and the yellow curves are for the validation set.

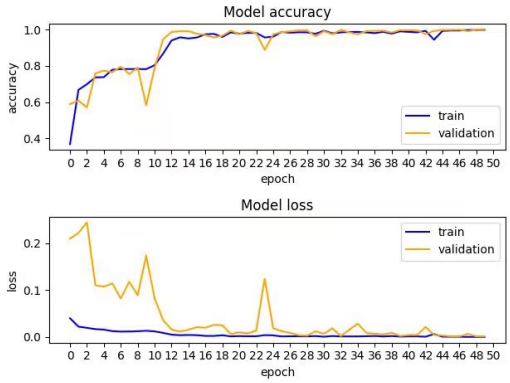


Figure 10. Accuracy loss curve of CNN-CBAM-BiLSTM model.

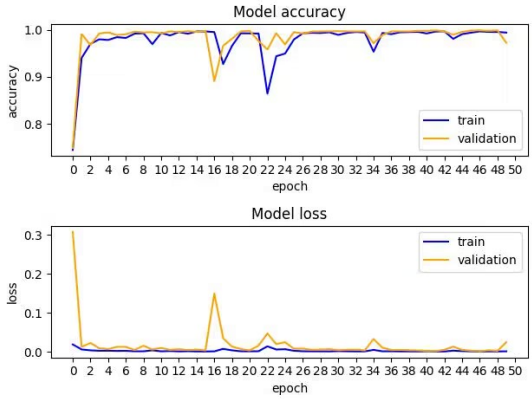


Figure 11. Accuracy loss curve of SE-Res2Net-CBAM-BiLSTM model.

Figures 10 and 11 show that on the CNN-CBAM-BiLSTM model, the accuracy and loss curves fluctuate, and thus, affect the stability of the model. In contrast, on the SE-Res2Net-CBAM-BiLSTM model, the accuracy and loss curve fluctuations were relatively minor and more stable.

Task 2: In this section, a comparative test on the multilingual cocktail dataset with a 100% overlap rate was carried out on the baseline model and the improved model. The accuracy of the baseline model CNN-CBAM-BiLSTM was 64%, and the specific results are shown in Table 4.

Table 4. Experimental results for multilingual cocktail party data with a 100% overlap rate.

Languages	Acc (%)	Precision (%)	Recall (%)	F1 Score
zh-cn (Mandarin)	61.51	63.71	60.62	62.12
id-id (Indonesian)	58.72	58.42	59.43	58.92
ja-jp (Japanese)	67.91	66.31	70.02	68.11
ko-kr (Korean)	56.12	43.83	73.34	54.87
vi-vn (Vietnamese)	66.23	68.41	64.45	66.37

The improved SE-Res2Net-CBAM-BiLSTM model was used on a multilingual cocktail party scenario in which the target language weight was 1.2 and the non-target language weight was 1. When the overlap was set to 100%, the accuracy of the model was 75%, as shown in Table 5.

Table 5. Experimental results of improved models on multilingual cocktail party data.

Languages	Acc (%)	Precision (%)	Recall (%)	F1 Score
zh-cn (Mandarin)	71.45	73.14	69.91	71.57
id-id (Indonesian)	69.42	68.52	70.13	69.32
ja-jp (Japanese)	79.14	77.41	81.24	79.37
ko-kr (Korean)	68.91	59.63	83.16	69.46
vi-vn (Vietnamese)	76.82	79.12	75.24	77.13

As seen in Tables 4 and 5, after using the model proposed in this article, the accuracy of the model on the multilingual cocktail party dataset was significantly improved by 11% compared to the baseline model. Thus, this also validates the effectiveness of the improved model proposed in this paper.

Task 3: Comparison of loss functions. In the previous language identification model, cross-entropy loss was used for classification, but it needed to take into account the problems of unbalanced data and confusion of languages. We used focal loss [36] to solve this problem effectively and improved the model performance. The results of language identification under different loss functions are shown in Table 6 below.

Table 6. Graph of language identification results with different loss functions.

Loss Function	Model	Acc (%)	Precision (%)	Recall (%)	F1 Score
Cross-entropy loss	CNN-CBAM-BiLSTM	94.57	94.64	94.91	94.77
Focal loss		95.65	95.68	94.95	95.31
Cross-entropy loss	SE-Res2Net-CBAM-BiLSTM	96.26	96.11	95.07	95.59
Focal loss		97.31	96.97	96.86	96.91

As can be seen in Table 6, the model with the focal loss function performed better when experimenting with both loss functions under the unified model. Compared to the cross-entropy loss, the recognition accuracy under the focal loss was improved by about 1%. Therefore, the experimental results show that the performance of the language identification network using the focal loss function was better than that using the cross-entropy loss function.

4.5. Limitation

This paper proposes a SE-Res2Net-CBAM-BiLSTM model, a language identification method based on a multi-scale feature extraction network. This model improves the

recognition performance compared to the baseline, but it still faces some difficulties in dealing with multilingual overlapping speech scenarios. One of the reasons is that this study only used public datasets and our synthesized multilingual cocktail dataset, which simulated the situation of different energy ratios with only two speakers. However, actual multilingual overlapping speech scenes are more complex and may involve three or more speakers and languages. Therefore, future research can explore more realistic and diverse datasets, as well as more advanced feature extraction and recognition techniques. The aim of this paper was to propose a novel language identification method that can handle multilingual overlapping speech effectively. We will continue to improve the performance of the model while studying more complex multilingual cocktail party scenarios.

5. Conclusions and Future Work

In this paper, we propose an improved SE-Res2Net-CBAM-BiLSTM method that can extract multi-scale features from speech signals. We evaluated our method on two datasets: AP17-OLR, a public Oriental language dataset, and a multilingual cocktail party dataset that we constructed with different energy ratios and speaker numbers. The experimental results show that the language identification accuracy of the improved SE-Res2-Net-CBAM-BiLSTM network on the AP17-OLR dataset was improved by about 3% compared to the baseline model CNN-CBAM-BiLSTM. On a multilingual cocktail party dataset of a 100% overlapped scenario, with a target language weight of 1.2 over the non-target language weight of 1, the accuracy of the proposed model was improved by 11% compared to the baseline model. In addition, the comparative experiments show that the accuracy, recall, and F1 values of the model in this paper were improved over the other three models, and the stable model performance indicates better robustness. Finally, different loss functions were also compared, and the focal loss method produced better results.

In the future, we will design new model frameworks to improve the performance of language identification networks. In addition, we will variously simulate multi-language cocktail party scenarios and conduct joint training experiments on source separation and language identification tasks.

Author Contributions: Conceptualization, Z.A. and M.A; methodology, Z.A. and A.H.; software, Z.A.; validation, M.A. and A.H.; formal analysis, Z.A.; investigation, Z.A. and M.A.; resources, M.A. and A.H.; data curation, M.A.; writing—original draft preparation, Z.A.; writing—review and editing, M.A. and A.H.; visualization, Z.A. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Strengthening Plan of the National Defense Science and Technology Foundation of China (grant number: 2021-JCJQ-JJ-0059) and the Natural Science Foundation of China (grant number: U2003207).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset supporting the conclusions of this article is available at https://www.speechocean.com, accessed on 11 March 2020.

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

References

1. Hazen, T.J.; Zue, V.W. Recent improvements in an approach to segment-based automatic language identification. In Proceedings of the 3rd International Conference on Spoken Language Processing (ICSLP 1994), Yokohama, Japan, 18–22 September 1994; pp. 1883–1886.
2. Navratil, J. Spoken language identification-a step toward multilinguality in speech processing. *IEEE Trans. Speech Audio Process.* **2001**, *9*, 678–685. [CrossRef]
3. Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 1505–1518. [CrossRef]
4. Wong, E. Automatic Spoken Language Identification Utilizing Acoustic and Phonetic Speech Information. Ph.D. Thesis, Queensland University of Technology, Brisbane City, Australia, 2004.

5. Lopez-Moreno, I.; Gonzalez-Dominguez, J.; Plchot, O.; Martinez, D.; Gonzalez-Rodriguez, J.; Moreno, P. Automatic language identification using deep neural networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 5337–5341.
6. 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.
7. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
8. Tang, Z.; Wang, D.; Chen, Y.; Chen, Q. AP17-OLR Challenge: Data, Plan, and Baseline. In Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia, 12–15 December 2017; pp. 1–5.
9. Baldwin, T.; Lui, M. Language identification: The long and the short of the matter. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, CA, USA, 1–6 June 2010; pp. 229–237.
10. Singh, G.; Sharma, S.; Kumar, V.; Kaur, M.; Baz, M.; Masud, M. Spoken language identification using deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 5123671. [CrossRef] [PubMed]
11. Toshniwal, S.; Sainath, T.N.; Weiss, R.J. Multilingual Speech Recognition with a Single End-to-End Model. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4904–4908.
12. Torres-Carrasquillo, P.A.; Singer, E.; Kohler, M.A.; Greene, R.J. Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP2002—INTERSPEECH 2002, Denver, CO, USA, 16–20 September 2002.
13. Zissman, M.A. Automatic language identification using Gaussian mixture and hidden Markov models. In Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, 27–30 April 1993; Volume 2, pp. 399–402.
14. Wang, X.L.; Wu, Z.G.; Zhou, R.H.; Yan, Y.H. Language-Pair Scoring Method Based on SVM for Language Recognition. *Appl. Mech. Mater.* **2013**, *333*, 737–741. [CrossRef]
15. Dehak, N.; Pedro, A.; Torres-Carrasquillo; Reynolds, D.; Dehak, R. Language identification via i-vectors and dimensionality reduction. In Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011, Florence, Italy, 28–31 August 2011; pp. 857–860.
16. Jiang, B.; Song, Y.; Wei, S.; Liu, J.-H.; McLoughlin, I.V.; Dai, L.-R. Deep bottleneck features for spoken language identification. *PLoS ONE* **2014**, *9*, e100795. [CrossRef] [PubMed]
17. Gonzalez-Dominguez, J.; Lopez-Moreno, I.; Sak, H.; Gonzalez-Rodriguez, J.; Moreno, P.J. Automatic language identification using long short-term memory recurrent neural networks. In Proceedings of the 15th Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014; pp. 2155–2159.
18. Fernando, S.; Sethu, V.; Ambikairajah, E. Factorized hidden variability learning for adaptation of short duration language identification models. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5204–5208.
19. Padi, B.; Mohan, A.; Ganapathy, S. End-to-end language identification using attention based hierarchical gated recurrent unit models. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5966–5970.
20. Lei, Y.; Ferrer, L.; Lawson, A.; McLaren, M.; Scheffer, N. Application of convolutional neural networks to language identification in noisy conditions. In Proceedings of the Odyssey 2014: The Speaker and Language Recognition Workshop, Joensuu, Finland, 16–19 June 2014; pp. 287–292.
21. Geng, W.; Wang, W.; Zhao, Y.; Cai, X.; Xu, B. End-to-end language identification using attention based recurrent neural networks. In Proceedings of the 17th Annual Conference of the International Speech and Communication Association, San Francisco, CA, USA, 8–12 September 2016; pp. 2944–2948.
22. Bartz, C.; Herold, T.; Yang, H.; Meinel, C. Language identification using deep convolutional recurrent neural networks. In Proceedings of the Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, 14–18 November 2017; pp. 880–889.
23. Cai, W.; Cai, D.; Huang, S.; Li, M. Utterance-level end-to-end language identification using attention-based CNN-BLSTM. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5991–5995.
24. Romero, D.; D'Haro, L.F.; Salamea, C. Exploring transformer-based language identification using phonotactic information. In Proceedings of the Fifth International Conference IberSPEECH 2021, Valladolid, Spain, 24–25 March 2021; pp. 250–254.
25. Benhur, S.; Sivanraju, K. Pretrained Transformers for Offensive Language Identification in Tanglish. *arXiv* **2021**, arXiv:2110.02852.
26. Nie, Y.; Zhao, J.; Zhang, W.-Q.; Bai, J. BERT-LID: Leveraging BERT to Improve Spoken Language Identification. *arXiv* **2022**, arXiv:2203.00328.
27. Borsdorf, M.; Li, H.; Schultz, T. Target Language Extraction at Multilingual Cocktail Parties. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Cartagena, Colombia, 13–17 December 2021; pp. 717–724.

28. Borsdorf, M.; Scheck, K.; Li, H.; Schultz, T. Experts Versus All-Rounders: Target Language Extraction for Multiple Target Lan-guages. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 846–850.
29. Borsdorf, M.; Scheck, K.; Li, H.; Schultz, T. Blind Language Separation: Disentangling Multilingual Cocktail Party Voices by Language. In Proceedings of the 23rd INTERSPEECH Conference, Incheon, Republic of Korea, 18–22 September 2022; pp. 256–260.
30. Ablimit, M.; Xueli, M.; Hamdulla, A. Language Identification Research Based on Dual Attention Mechanism. In Proceedings of the 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 16–18 July 2021; pp. 241–246.
31. Revathi, A.; Jeyalakshmi, C. Robust speech recognition in noisy environment using perceptual features and adaptive filters. In Proceedings of the 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 19–20 October 2017; pp. 692–696.
32. Davis, S.; Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
34. Bizzoni, Y.; Ghanimifard, M. Bigrams and BLSTMs two neural networks for sequential metaphor detection. In Proceedings of the Workshop on Figurative Language Processing, New Orleans, LA, USA, 6 June 2018.
35. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
36. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.