applied sciences

*Article*

# Multi-Scale Channel Adaptive Time-Delay Neural Network and Balanced Fine-Tuning for Arabic Dialect Identification

**Qibao Luo and Ruohua Zhou ***

Department of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
* Correspondence: zhouruohua@bucea.edu.cn

**Abstract:** The time-delay neural network (TDNN) can consider multiple frames of information simultaneously, making it particularly suitable for dialect identification. However, previous TDNN architectures have focused on only one aspect of either the temporal or channel information, lacking a unified optimization for both domains. We believe that extracting appropriate contextual information and enhancing channels are critical for dialect identification. Therefore, in this paper, we propose a novel approach that uses the ECAPA-TDNN from the speaker recognition domain as the backbone network and introduce a new multi-scale channel adaptive module (MSCA-Res2Block) to construct a multi-scale channel adaptive time-delay neural network (MSCA-TDNN). The MSCA-Res2Block is capable of extracting multi-scale features, thus further enlarging the receptive field of convolutional operations. We evaluated our proposed method on the ADI17 Arabic dialect dataset and employed a balanced fine-tuning strategy to address the issue of imbalanced dialect datasets, as well as Z-Score normalization to eliminate score distribution differences among different dialects. After experimental validation, our system achieved an average cost performance ($C_{avg}$) of 4.19% and a 94.28% accuracy rate. Compared to ECAPA-TDNN, our model showed a 22% relative improvement in $C_{avg}$. Furthermore, our model outperformed the state-of-the-art single-network model reported in the ADI17 competition. In comparison to the best-performing multi-network model hybrid system in the competition, our $C_{avg}$ also exhibited an advantage.

**Keywords:** Arabic dialect identification; multi-scale learning; channel adaptation; balanced fine-tuning; Z-Score normalization

check for updates

**Citation:** Luo, Q.; Zhou, R. Multi-Scale Channel Adaptive Time-Delay Neural Network and Balanced Fine-Tuning for Arabic Dialect Identification. *Appl. Sci.* **2023**, *13*, 4233. https://doi.org/10.3390/app13074233

Academic Editor: Javier Hernando

Received: 22 February 2023
Revised: 17 March 2023
Accepted: 25 March 2023
Published: 27 March 2023

## 1. Introduction

Dialect identification (DID) refers to the task of identifying the dialect category, which generally belongs to the same language family and can be considered as a special case of language identification. With the development of i-vector, x-vector, and neural networks, language identification has achieved significant success [1–3]. In recent years, accent and dialect identification have received increasing attention from speech researchers [4–9]. However, DID is often more challenging than the language identification task, as similar dialects often share similar feature spaces [10]. Dialect identification technology is commonly applied to the front-end of speech processing systems, such as automatic speech recognition (ASR) and multilingual translation systems. Therefore, the development of DID technology is essential in the era of speech interaction intelligence.

Currently, the x-vector-based architecture remains the most widely used method for DID [11]. The X-vector is a method that utilizes deep neural networks to extract features from speech signals, enabling the mapping of high-dimensional speech features to a fixed-length vector to represent speech information. Unlike traditional speech feature extraction methods, the X-vector is trained through a deep learning model and can better express the nonlinear characteristics of speech signals. With the continuous development of deep learning, numerous excellent x-vector architectures have been proposed for speech, including the time-delay neural network (TDNN) [12], extended time-delay neural network (E-TDNN) [13], and factorized time-delay neural network (F-TDNN) [14]. TDNN is a suitable approach for speech and time-series signal processing as it considers multiple consecutive frames of information during network input. E-TDNN utilizes a deeper network structure than TDNN to establish gradual context connections. F-TDNN decomposes the parameter matrix into smaller matrices, effectively reducing the layer parameters, and leveraging the advantages of network depth when overall parameters are comparable. Additionally, some researchers have proposed the ECAPA-TDNN architecture [15], which emphasizes channel attention, propagation, and aggregation, and has made significant progress in speaker recognition.

Arabic dialect identification (ADI) is one of the two tasks in the latest MGB-5 challenge [16]. The task involves identifying audio categories from 17 Arabic dialect audios collected from YouTube. Arabic dialects belong to various dialects within the same language family, making their distinction challenging and more difficult than other easy-to-understand dialects.

To enhance the performance of recognizing similar-sounding Arabic dialects, we propose a new network module named the multi-scale channel adaptive module (MSCA-Res2Block). The MSCA-Res2Block is designed to address issues in the ECAPA-TDNN, such as an insufficient receptive field, inflexible context information extraction, and inadequate local channel attention. It consists of two components, namely, multi-scale dilated convolutions and multi-scale channel attention. The former allows the network to capture receptive fields of different sizes to adapt to the varying speaking rates in different dialects, while the latter can dynamically calibrate channel attention features extracted at different scales. The unique aspect of the MSCA-Res2Block design is that it combines both global and local attention mechanisms, which selectively amplify informative features and suppress irrelevant features, thus achieving more effective feature representation. Such feature representation is particularly critical for dialect identification tasks, as the network needs to capture subtle pronunciation and intonation differences. The MSCA-Res2Block module can adapt to these changes and extract the most prominent features to improve the accuracy of recognition tasks. We use this module to improve the ECAPA-TDNN, which we call the MSCA-TDNN network.

In our experiment we proposed a balanced fine-tuning method aimed at addressing the issue of significant data imbalance in Arabic dialect datasets. In essence, we constructed a balanced subset of data for fine-tuning the model trained on the complete dataset, with the goal of reducing model bias caused by dataset unfairness. Inspired by score normalization in speaker recognition, we believed that the score distribution may vary between different dialects. Therefore, we introduced the Z-Score method to normalize the scores between dialects.

Our contributions are threefold:

1. We propose a dialect identification model that is based on the ECAPA-TDNN network as the backbone and that incorporates the proposed MSCA-Res2Block module, which enables multi-scale channel adaptation in TDNN. Our model improves $C_{avg}$ by 22% compared to ECAPA-TDNN.
2. We propose a balanced fine-tuning strategy to address data imbalance, which has potential applicability in other domains.
3. We investigate the complementary effects of training models with diverse data and introduce the Z-Score standardization approach to address the variations in score distribution among distinct dialects.

The remainder of this paper is organized as follows. Section 2 outlines the critical components of the ECAPA-TDNN, while Section 3 describes the structure of the proposed MSCA-Res2Block. We provide a detailed account of the experimental setup in Section 4, and Section 5 presents and analyzes the experimental results. Finally, in Section 6, we present a brief summary of our experiments and provide recommendations for future research.

## 2. Related Works

In this section, we mainly introduce the key module of ECAPA-TDNN and its advantages and disadvantages. ECAPA-TDNN is a novel speaker embedding extractor that uses the Time Delay Neural Network (TDNN) for speaker verification. This model builds on the original x-vector architecture and places additional emphasis on channel attention, propagation, and aggregation. ECAPA-TDNN demonstrates substantial improvements in speaker verification performance, despite its relatively low number of parameters. ECAPA-TDNN mainly consists of SE-Res2Block modules, multi-level inputs, multi-scale aggregation, and attention pooling modules.

The SE block in SE-Res2Net models the dependency between channels, while the Res2Net module enhances feature processing by constructing a hierarchical residual connection within the convolution layer. The multi-scale input and multi-scale aggregation aim to leverage features at different levels of the neural network. Attention pooling assigns different weights based on their importance to the task, allowing the network to focus more on speech features that are discriminative.

Although the ECAPA-TDNN architecture has the above-mentioned advantages, there are still some shortcomings in its application to dialect recognition. Firstly, the fixed convolution kernel in the SE-Res2Block may not capture broader contextual information, which could have a negative impact on model performance for dialect recognition tasks that require more contextual information. Secondly, although the attention mechanism in the SE-Res2Block can help the model focus on important features, it does not consider local channel information, which is also important for more informative features in dialect recognition tasks.

To address these issues, we propose the MSCA-Res2Block. This module consists of two parts: multi-scale dilated convolution and multi-scale channel attention. The multi-scale dilated convolution allows the network to learn features at different scales, better adapting to changes in speech speed; by comparison, the multi-scale channel attention can dynamically adjust the weights between different feature channels at different scales, allowing the network to better capture subtle feature information related to pronunciation and tone differences. The detailed structure design is introduced in Section 3. The following subsection provides a description of the main modules of ECAPA-TDNN.

### *2.1. SE-Res2Block*

The ECAPA-TDNN [15] incorporates one-dimensional SE [17] and Res2Net [18] to create the SE-Res2Block module. The SE module has demonstrated remarkable success in global channel interdependence in image vision applications, and has been adapted to one-dimensional SE for speech applications [19].

Res2Net is a deep convolutional neural network architecture that can be used for image classification and object detection. It replaces the traditional 3 × 3 filters with a set of smaller filter groups and uses hierarchical residual-style connections between these filter groups to enhance the network's feature representation capability. Specifically, the input feature map is divided into multiple groups, and each group, except for the first group, uses a filter to extract features. The output features of the previous group are then sent together with the input features of another group to the next filter. This operation can be expressed by the following formula:

$$y_i = \begin{cases} x_i & i = 1 \\ C_i(x_i) & i = 2 \\ C_i(x_i + y_{i-1}) & 2 < i \leq s \end{cases} \tag{1}$$

Here, $C_i$ represents the convolution operation, $x_i$ represents the $i$-th group input feature, $y_i$ represents the $i$-th group output feature, and $s$ represents the group size.

### *2.2. Multi-Level Inputs, Multi-Scale Aggregation, and Attention Pooling*

The approach for multi-level input involves utilizing the outputs of all prior SE-Res2Block blocks and the initial convolution layer as the input of each frame layer block [20]. To minimize the number of model parameters, feature map accumulation is selected over feature map connections. The following formula can be represented as:

$$out_n = \begin{cases} g_n(out_1) & n = 1 \\ \\ g_n(out_1 + \cdots + out_{n-1}) & n \geq 2 \end{cases} \tag{2}$$

Here, $out_n$ represents the output of the $n$-th bottleneck network layer, and $g_n$ denotes the $n$-th bottleneck network.

Recent research, such as [20,21], demonstrates that the aggregation of different network feature layers can enhance the accuracy of speaker embedding models in speaker verification tasks. Multi-scale aggregation in ECAPA-TDNN involves concatenating the output features of all SE-Res2Block modules. It can be expressed mathematically as follows:

$$out = \text{concat}[out_2, out_3, \ldots, out_n] \tag{3}$$

Furthermore, ECAPA-TDNN extends the temporal attention mechanism to the channel dimension and performs attention pooling on the aggregated features [22]. The activation of the last frame layer at time step $t$ is represented as $x_t$. It is noteworthy that the computation of the scores $e_{t,c}$ involves projecting the self-attention information using parameters $W$ and $b$ onto a smaller R-dimensional representation, followed by a non-linear activation function $f(\ )$ and a linear layer with weights $v_c$ and bias $k_c$. Subsequently, the importance $\alpha_{t,c}$ of each frame in a given channel is obtained by applying the softmax function along the temporal dimension, and can be described mathematically as:

$$e_{t,c} = v_c^T f(Wx_t + b) + k_c \tag{4}$$

$$\alpha_{t,c} = \frac{exp(e_{t,c})}{\sum_{\tau}^{T} exp(e_{\tau,c})} \tag{5}$$

For the attention statistics of each utterance, we weighted mean vector $\mu_c$ and weighted standard deviation vector $\sigma_c$. They can be expressed using the following formula:

$$\mu_c = \sum_t^T \alpha_{t,c} x_{t,c} \tag{6}$$

$$\sigma_c = \left( \sum_t^T \alpha_{t,c} x_{t,c}^2 - \mu_c^2 \right)^{\frac{1}{2}} \tag{7}$$

The final output of the global multi-scale attention pooling layer is obtained by connecting the vectors of weighted mean $\mu$ and weighted standard deviation $\sigma$. These modules were also incorporated into our model design.

## 3. The Proposed Model Architecture

In this section, we address some limitations of the ECAPA-TDNN architecture and propose a module that is more suitable for the DID task. In the SE-Res2Block module, the convolutional kernels of each layer are fixed and do not consider local channel information. However, for dialect identification, a larger receptive field and richer features are typically beneficial. To address the shortcomings of the SE-Res2Block module, we propose a multi-scale channel adaptive Res2Block module. We replace the SE-Res2Block module in the ECAPA-TDNN architecture with our MSCA-Res2Block module. For ease of description, we refer to this ECAPA-TDNN architecture with multi-scale channel adaptive module as the MSCA-TDNN architecture, as shown in Figure 1. We describe the structure of the MSCA-Res2Block module below.

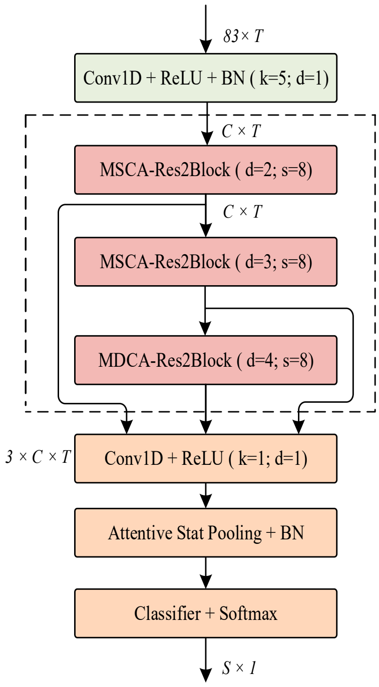
83× T
Conv1D + ReLU + BN ( k=5; d=1)
C × T
MSCA-Res2Block ( d=2; s=8)
C × T
MSCA-Res2Block ( d=3; s=8)
MDCA-Res2Block ( d=4; s=8)
3 × C × T
Conv1D + ReLU ( k=1; d=1)
Attentive Stat Pooling + BN
Classifier + Softmax
S × 1

**Figure 1.** MSCA-TDNN.

Among them, Conv1D represents one-dimensional convolution, $k, d$, and $s$ represent convolution kernel size, dilated convolution dilation rate, and grouping size, respectively, $C$ and $T$ correspond to the channel number and time dimension of feature map, respectively, and $S$ represents dialect classification number.

*MSCA-Res2Block*

The multi-scale channel adaptive (MSCA) module is an attention mechanism that allows convolutional neural networks to adaptively learn the importance of different scale channels, in order to improve the model's representation and generalization performance. The structure of MSCA, as shown in Figure 2, can be obtained through the following steps.

The first step of the MSCA module is to obtain multi-scale feature maps through n dilated convolutions of different size scales. Dilated convolution is a technique that increases the receptive field of the convolutional kernel while keeping the same number of parameters. This means that dilated convolution can capture information from larger regions of the input feature map, allowing the network to learn more meaningful features.

Assuming the input feature is $X$, the multi-scale feature maps obtained through dilated convolution can be described as a set of matrices, where each matrix represents the response of different filters to the input feature. The filters are defined by the convolution kernel size $k_i$ and dilation rate $d$, which determine the size of their receptive field. By using multiple filters with different receptive fields, the MSCA module can capture features of different scales $F_i$, allowing it to learn more complex patterns and improve the accuracy of dialect identification. This can be described as:

$$k_i = 2i - 1 \quad (i = 1, 2 \cdots n) \tag{8}$$

$$F_i = Conv(k_i, d)(X) \quad (i = 1, 2 \cdots n) \tag{9}$$

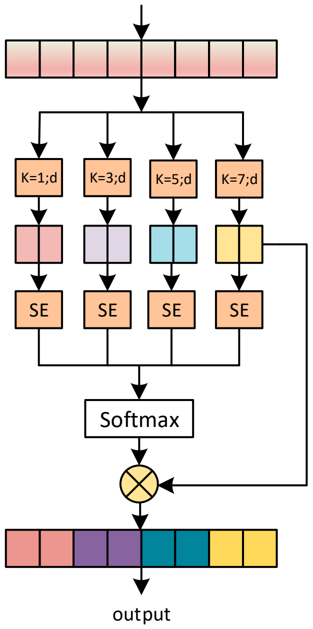

**Figure 2.** Structure of MSCA.

In order to reduce parameters and save computational resources, we used a dimension reduction $m$ in the channel dimension, which reduces the output channels to $1/m$ of the original feature map. In this experiment, we set the dimension reduction factor $m$ to 4 and the number of dilation convolution $n$ to 4.

The Squeeze-and-Excitation (SE) module is highly effective in enhancing the performance of neural networks. The SE module enhances the influence of useful feature channels by learning the weights of each channel. The basic idea is to use another neural network to learn the weights of each channel, and then apply the weights to each channel of the input feature map, allowing the neural network to focus on the useful feature channels.

In a multi-scale scenario, we can apply the SE module on different scale feature maps $F_i$ to obtain multi-scale channel attention $z_i$, which can help the network to better understand the input feature maps and improve the model performance.

We applied squeeze operations on multiple scale features $F_i$ to compute the average vector $s_i$ of frame-level features at different scales:

$$s_i = \frac{1}{T}\sum_{t=1}^{T} F_i(t) \quad (i = 1, 2 \cdots n) \tag{10}$$

Here, $T$ denotes the total number of frames on the scale, and $F_i(t)$ represents the feature vector of the $t$-th frame on the $i$-th scale.

Next, we perform the excitation operation to calculate the weights of the multi-scale channels, which can be defined as:

$$z_i = \sigma(W_2 f(W_1 s_i + b_1) + b_2) \quad (i = 1, 2 \cdots n) \tag{11}$$

Here, $\sigma$ represents the sigmoid function, $f$ represents a non-linear function, $b_1$ and $b_2$ are bias terms, and $W_1 \in \mathbb{R}^{R\times C}$, $W_2 \in \mathbb{R}^{C\times R}$ (where $C$ and $R$ represent the number of input channels and the reduction factor, respectively, and in our experiments, $C$ is set to 64 and $R$ is set to $1/2$).

After the multi-scale attention processing, the SE module obtains the channel attention of each scale feature map, but not all scales have equal importance for the task. Therefore, it is necessary to further calibrate the different scale feature maps to obtain more effective feature representations. To address this issue, this paper proposes a method for adaptively calibrating the channel attention vectors of different scale convolutions. Specifically, the softmax function is used to normalize the channel attention feature vector $z_i$ of each feature map, thereby obtaining the multi-scale channel adaptive weight $W_i$:

$$W_i = Softmax(z_i) = \frac{exp(z_i)}{\sum_{i=1}^{n} exp(z_i)} \quad (i = 1, 2 \cdots n) \tag{12}$$

After this step, each scale's feature map is adaptively weighted based on its contribution to the task, leading to more effective feature representation. Finally, we multiply the corresponding scale feature map $F_i$ with its adaptive weight $W_i$. To ensure that the number of channels remains the same at each layer, we concatenate in the channel dimension, yielding the multi-scale channel adaptive feature map $Y$, which can be expressed as:

$$Y = Concat[W_1F_1; W_2F_2; \ldots; W_nF_n] \tag{13}$$

We embed the MSCA module in Res2Net, as shown in Figure 3. From the above analysis, our proposed MSCA-Res2Block module pays attention to multi-scale space and channel information locally. It allows information to interact better and makes the network better adapt to the extraction of complex contextual information in dialect identification.

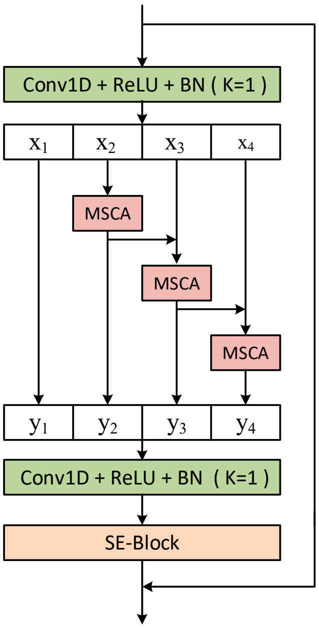

**Figure 3.** MSCA-Res2Block.

## 4. Experimental Setup

### *4.1. Dataset*

The ADI17 dataset is provided by MGB-5 [23]. We followed the dataset partitioning of the MGB-5 challenge without any modifications. The ADI17 dataset we used was collected from YouTube, consisting of approximately 3000 h of Arabic dialect speech from 17 Arab countries, with significant variation in data proportions across different languages. As the data were collected via specific country YouTube channels, label errors may exist. Therefore, the official dataset was first adjusted and verified manually for dialect labels from approximately 280 h of speech data, and then 58 h of data was chosen for performance evaluation as the development and test sets. The test set was grouped into three subcategories based on the duration of speech segments: short (<5 s), medium (5–20 s), and long (>20 s). The detailed statistical data of the ADI17 dataset are presented in Table 1.

**Table 1.** Speech data for Arabic dialect identification.

| | **Training** | **Dev** | **Test** |
|---|---|---|---|
| Dur (h) | 3033.4 | 24.9 | 33.1 |
| Utterances | 1,043,269 | 8955 | 12,615 |

**Speed Perturbation Data:** To enhance the performance of dialect identification, we applied data augmentation by using the open-source software SoX to generate speed perturbations with coefficients of 0.9, 1.0, and 1.1 for the training set before training. The official speech dataset provided varying durations, so we segmented them into 3 s speech clips and repeated shorter clips to obtain a total of 12,148,293 segments. For the development set and testing set, no manipulation was performed.

**Balanced Data:** We observed that the speech duration of the IRA dialect in the official raw data was 815.8 h, while the speech duration of JOR and SUD dialects was only 25.9 and 47.7 h, respectively. Therefore, we applied special processing to JOR and SUD. Specifically, we applied speed perturbation to JOR and obtained 98,189 segments of speech, followed by RIR data augmentation, resulting in 239,482 segments of speech. Similarly, we applied speed perturbation to SUD and obtained 186,789 segments of speech, followed by RIR data augmentation, resulting in 447,602 segments of speech. Although special data augmentation was applied to dialects with limited samples, the problem of severe data imbalance remained, greatly affecting training speed and the fairness of the model's identification of dialects with a smaller proportion. To eliminate the issue of unbalanced data, we randomly selected 200,000 3 s speech segments for each dialect to form the balanced training subset for this paper, while maintaining the same development and test sets as the official dataset.

### *4.2. Experimental Details*

To extract features from Arabic dialect audio, we used the Kaldi platform [24] to extract 80-dimensional Fbank features and 3-dimensional pitch features. The feature extraction used a frame length of 25 ms and a frame shift of 10 ms, without using energy VAD to filter out non-speech frames. Finally, we performed cepstral mean and variance normalization (CMVN) on the extracted 83-dimensional features to improve the system's robustness.

Our model was implemented using the PyTorch toolkit [25], with Adam [26] chosen as the optimizer and an initial learning rate set to 0.001. We used a cosine annealing schedule to decrease the learning rate, and the batch size was 32. We employed two NVIDIA V100S GPUs with 32 GB memory each. The models were trained for 30 and 20 epochs, respectively, using balanced and speed perturbation data. An additional 10 epochs were trained during the fine-tuning process for balancing. Due to the particularly large size of the dialect dataset, each experiment required a month-long cycle to complete. Additionally, we proposed a balanced fine-tuning and score normalization method in the experiments. In addition, we proposed a balanced fine-tuning method and a score normalization method in the experiment.

**Balanced Fine-Tuning:** In order to address the issue of imbalanced data, we evaluated our proposed model using three data processing methods: balanced data, speed perturbation, and balanced fine-tuning. The specific steps of balanced fine-tuning are as follows: first, a model is trained using unbalanced data augmented with speed perturbation, and then the model is fine-tuned using a previously constructed balanced dataset to correct the fairness issues caused by significant differences in training data volume across different dialects.

**Z-Score:** The score distributions differ among different dialects, and even the same dialect may have significant variations in scores due to differences in semantic content and noise environments during data collection. Therefore, it is necessary to normalize the scores for each dialect. In our experiments, we used Z-Score normalization, also known as standard score.

Z-Score normalization can be used to compare the values of different datasets, especially when they have different means and standard deviations. In Arabic dialect classification scoring, using Z-Score normalization can map the score of each dialect to a standard normal distribution, thereby reducing the influence of different score distributions of dialects and facilitating the comparison of scores of different dialects.

As shown in Figure 4, we randomly selected 2500 utterances from the original training data of each dialect and fed them into our trained model to generate a score file. Utilizing the known labels from the training set, we obtained the mean and standard deviation of each dialect:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{14}$$

$$\sigma_j = \left(\frac{1}{n}\sum_{i=1}^{n} (x_{ij} - \mu_j)^2\right)^{\frac{1}{2}} \tag{15}$$

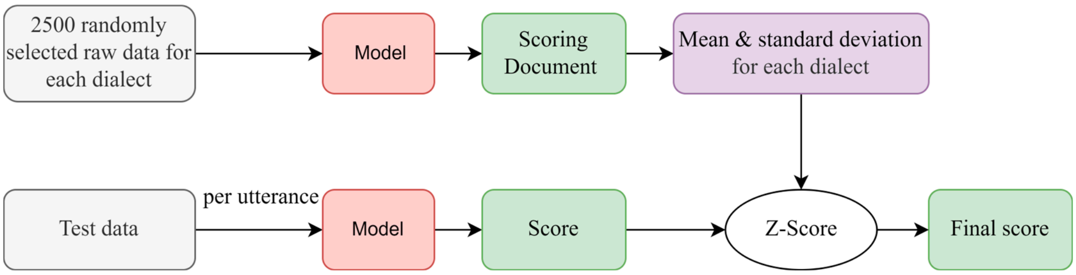
2500 randomly selected raw data for each dialect
Model
Scoring Document
Mean & standard deviation for each dialect
Test data
per utterance
Model
Score
Z-Score
Final score

**Figure 4.** Z-Score process.

Here, $X_{ij}$ represents the probability of the $j$-th dialect belonging to the $i$-th sentence in the speech corpus. $\mu_j$ represents the mean value of the $j$-th dialect, and $\sigma_j$ represents its standard deviation. $n$ represents the total number of speech utterances.

During the testing process, we input a complete speech sentence and perform softmax processing, resulting in 17 scores. We then use the previously obtained mean and standard deviation of each dialect to calculate the deviation from the mean (i.e., the difference between the test score and the mean), and divide it by the standard deviation. For the $j$-th dialect in the $i$-th speech sentence, the normalized value $Z_{ij}$ can be calculated using the following formula:

$$Z_{ij} = \left(X_{ij} - \mu_j\right)/\sigma_j \tag{16}$$

### *4.3. Evaluation Protocol*

To evaluate the performance of our model, we followed two official metrics in our experiments: overall accuracy and cost. The ADI17 task is a closed-set identification task, and the softmax output can be directly used as the score for each Arabic dialect. We calculate accuracy by selecting the highest score from the 17 dialects for each test utterance. For the cost metric, we used the average cost performance ($C_{avg}$) of each target/non-target pair defined in LRE NIST 2017 [27], with Ptarget set to 0.5.

## 5. Result and Analysis

### *5.1. Comparison of ECAPA-TDNN and MSCA-TDNN Systems*

We evaluated our system using the test set of ADI17. Table 2 shows the performance evaluation results of ECAPA-TDNN and MSCA-TDNN under three different data usage scenarios: balanced data, speed-perturbed data, and fine-tuned balanced data. Following the dialect evaluation standard, we used the average cost performance (Cavg) as the main metric. Overall, our proposed MSCA-TDNN system outperforms ECAPA-TDNN under different conditions. Finally, compared to ECAPA-TDNN, the Cavg of the MSCA-TDNN system relative improved by 22%. The Arabic dialects are very similar and lack distinct discriminative features. The superiority of MSCA-TDNN is due to its adoption of multi-scale dilated convolution to learn features at different scales, which better adapts to the speed variations in dialects. In addition, it also utilizes multi-scale channel attention to dynamically adjust the weights between different scale feature channels, enabling the network to better capture subtle phonetic and tonal differences, thereby improving the accuracy in dialect identification tasks. Although the balanced subset only used less than one-third of the entire dataset, its overall performance is still competitive. According to the analysis of the MSCA-TDNN test set, training with speed-perturbed data relative to using balanced data improved performance by 16%, whereas our proposed fine-tuned balanced data relative to using only speed-perturbed data improved performance by 13%.

**Table 2.** The average cost performance ($C_{avg}$ %) of ECAPA-TDNN and MSCA-TDNN under different data usage scenarios.

| Data Usage | ECAPA-TDNN | MSCA-TDNN |
|---|---|---|
| Balance data | 6.83 | 6.30 |
| Speed perturbation data | 6.09 | 5.28 |
| Balanced fine-tuning | 5.43 | 4.57 |
| Balance + speed+ fine-tuning | 6.00 | 5.54 |
| Balance + speed + fine-tuning + Z-Score | 5.39 | **4.19** |

We aimed to investigate whether models trained using three different data training methods (balanced data, speed perturbation data, and fine-tuned balanced data) have complementary effects. During testing, we fed the test speech into each of the three different models and then combined their output scores. However, we found that simple averaging of scores did not improve results. We speculate that this may be due to the influence of differences in dialect score distribution. Inspired by speaker recognition, we normalized scores using Z-Score before score fusion. Experimental results show that this method effectively improves dialect identification performance, which is evident in the last two rows of Table 2.

The confusion matrices in Figures 5 and 6 demonstrate the accuracy of the MSCA-TDNN model trained using speed perturbation data and balanced fine-tuning. As shown in Figure 5, the accuracy for most dialects exceeds 85%, while the JOR dialect has a lower accuracy of only 78.36%, indicating poorer performance. We believe this is due to the relatively small size of the JOR dialect dataset compared to the other dialects, which affected the fairness of the model. On the other hand, Figure 6 shows the confusion matrix after fine-tuning, which increased the accuracy of the JOR dialect to 84.88%. This indicates that balanced fine-tuning can correct the bias of the model and effectively improve the performance of dialect categories with limited data.

In Figure 5, another poorly performing dialect is SYR, which is most frequently confused with the EGY dialect, accounting for 4.47% of the total confusion pairs. We attribute this to the fact that Syria and Egypt are neighboring countries that merged to form the United Arab Republic in 1958, resulting in frequent communication between the two and possibly language convergence. The ADI17 dataset contains 119.5 h of SYR dialect data and 451.1 h of EGY dialect data. Therefore, when training the model, more emphasis is placed on the EGY dialect, which leads to poorer performance when evaluating the SYR dialect.

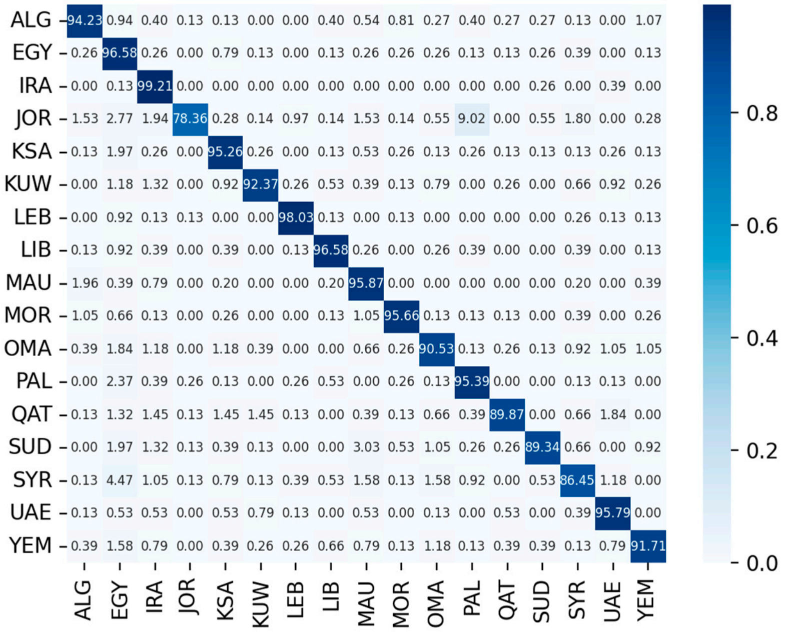

**Figure 5.** Dialect confusion matrix (speed perturbation).

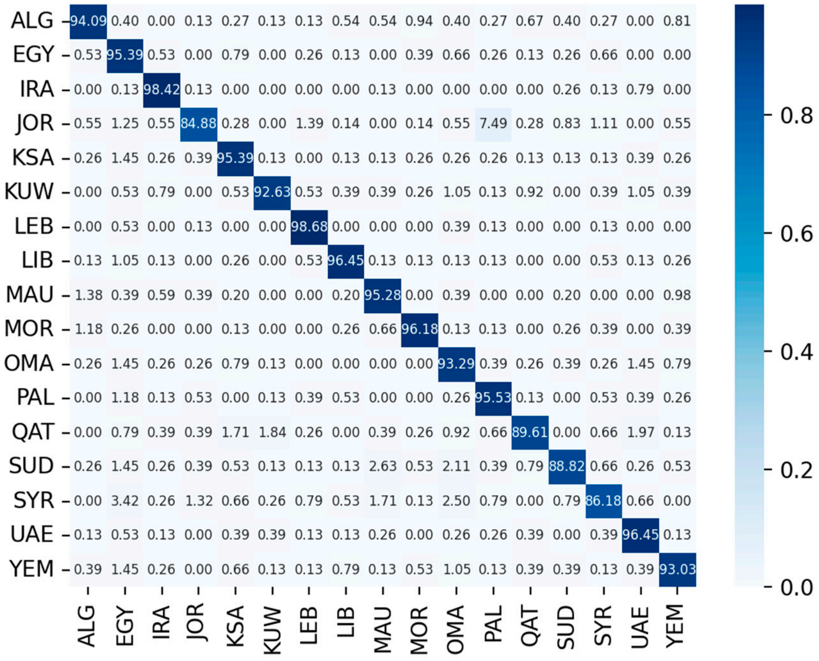

**Figure 6.** Dialect confusion matrix (balanced fine-tuning).

### *5.2. Comparison with Advanced Systems*

Table 3 presents the DID performance of some advanced systems and our proposed system on the ADI17 dataset. Results from different systems in the table show that the length of the speech has a significant impact on the test results, with longer speech typically resulting in better performance. The official best baseline is an end-to-end system that uses softmax layer output as the posterior probability of each dialect, which achieves an average cost performance $C_{avg}$ of 13.7% and an accuracy of 82% on the test set. Theoretically, transformer models can capture long-term information, which is beneficial for DID performance, but experimental results show that the improvement is very limited, indicating that overly long linguistic information cannot have a significant positive impact on Arabic DID tasks. DKU and Uken are two teams that participated in and won awards in the ADI17 competition. The DKU team that won first place achieved an accuracy of 93.8% on the test set using a single network model, while the system that combines multiple network models has an average cost performance $C_{avg}$ of 4.3% and an accuracy of 94.9%. The Uken system that won second place had an average cost performance $C_{avg}$ of 6.2% and an accuracy of 91.1% on the test set. Based on the characteristics of spectrogram information analysis and the shortcomings of ECAPA-TDNN in DID tasks, we propose a competitive MSCA-TDNN model. The model achieves an average cost performance $C_{avg}$ of 4.19% and an accuracy of 94.28% on the test set. Experimental results show that the multi-scale channel adaptive module is effective for dialect identification. The average cost performance $C_{avg}$ of our MSCA-TDNN system is the best result reported in the literature to date.

**Table 3.** Performance comparison of different systems. [$C_{avg}$ and Acc] (%).

| System | Overall | | <5 s | | 5∼20 s | | >20 s | |
|---|---|---|---|---|---|---|---|---|
| | $C_{avg}$ | Acc | $C_{avg}$ | Acc | $C_{avg}$ | Acc | $C_{avg}$ | Acc |
| E2E (Softmax) [23] | 13.7 | 82.0 | 18.8 | 76.2 | 10.9 | 85.1 | 6.7 | 90.4 |
| Transformer [10] | - | 82.54 | - | 76.21 | - | 86.01 | - | 90.58 |
| UKent [16] | 6.2 | 91.1 | 8.3 | 88.4 | 5.3 | 92.3 | 2.5 | 96.1 |
| DKU [16] | - | 93.8 | - | - | - | - | - | - |
| DKU (Fusion) [16] | 4.3 | **94.9** | **5.5** | **93.3** | 3.7 | **95.6** | 2.0 | 97.7 |
| MSCA-TDNN (Ours) | **4.19** | 94.28 | 5.64 | 92.23 | **3.45** | 95.22 | **1.98** | **98.12** |

## 6. Conclusions

This paper introduces for the first time ECAPA-TDNN, currently popular in speaker recognition, to recognize Arabic dialects. Starting from the goal of obtaining more information on channel and time dimensions for dialect identification, we propose a multi-scale channel adaptive time-delay neural network. We also address the data imbalance problem in the ADI17 dataset by using a balanced subset fine-tuning strategy, and standardizing the scores using the Z-Score before score fusion to mitigate the score distribution differences between dialects. We evaluate our proposed models and compare them with other systems submitted to the ADI17 challenge. Our experiments show that ECAPA-TDNN is highly suitable for Arabic dialect identification, and our proposed MSCA-TDNN further demonstrates more powerful performance. To the best of our knowledge, the performance of MSCA-TDNN is superior to the results of existing single network models. Our future work will focus on optimizing network architecture from the aspects of adaptively extracting contextual information and enhancing channels to better improve dialect identification performance.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** The Arabic Dialect Identification for 17 Countries (ADI17) dataset used in this paper can be accessed by the following link: (http://groups.csail.mit.edu/sls/downloads/adi17/, accessed on 24 March 2023).

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

## References

1. Dehak, N.; Torres-Carrasquillo, P.A.; Reynolds, D.; Dehak, R. Language Recognition via I-Vectors and Dimensionality Reduction. In Proceedings of the Interspeech 2011, Florence, Italy, 27 August 2011; pp. 857–860.
2. Snyder, D.; Garcia-Romero, D.; McCree, A.; Sell, G.; Povey, D.; Khudanpur, S. Spoken Language Recognition Using X-Vectors. In Proceedings of the The Speaker and Language Recognition Workshop (Odyssey 2018), Baltimore, MD, USA, 26 June 2018; pp. 105–111.
3. Richardson, F.; Reynolds, D.A.; Dehak, N. A Unified Deep Neural Network for Speaker and Language Recognition. In Proceedings of the Interspeech 2015, Dresden, Germany, 6–10 September 2015; pp. 1146–1150.
4. Biadsy, F. *Automatic Dialect and Accent Recognition and Its Application to Speech Recognition*; Columbia University: New York, NY, USA, 2011.
5. Jiao, Y.; Tu, M.; Berisha, V.; Liss, J. Accent Identification by Combining Deep Neural Networks and Recurrent Neural Networks Trained on Long and Short Term Features. In Proceedings of the Interspeech 2016, San Francisco, CA, USA, 8 September 2016; pp. 2388–2392.
6. Li, B.; Sainath, T.N.; Sim, K.C.; Bacchiani, M.; Weinstein, E.; Nguyen, P.; Chen, Z.; Wu, Y.; Rao, K. Multi-Dialect Speech Recognition with a Single Sequence-to-Sequence Model. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4749–4753.
7. Gao, Q.; Wu, H.; Sun, Y.; Duan, Y. An End-to-End Speech Accent Recognition Method Based on Hybrid CTC/Attention Transformer ASR. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7253–7257.
8. Li, Z.; Zhao, M.; Hong, Q.; Li, L.; Tang, Z.; Wang, D.; Song, L.; Yang, C. AP20-OLR Challenge: Three Tasks and Their Baselines. In Proceedings of the 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand, 7–10 December 2020; pp. 550–555.
9. Wang, B.; Hu, W.; Li, J.; Zhi, Y.; Li, Z.; Hong, Q.; Li, L.; Wang, D.; Song, L.; Yang, C. OLR 2021 Challenge: Datasets, Rules and Baselines. In Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 14–17 December 2021; pp. 1097–1103.
10. Lin, W.; Madhavi, M.; Das, R.K.; Li, H. Transformer-Based Arabic Dialect Identification. In Proceedings of the 2020 International Conference on Asian Language Processing (IALP), Kuala Lumpur, Malaysia, 4–6 December 2020; pp. 192–196.
11. Amani, A.; Mohammadamini, M.; Veisi, H. Kurdish Spoken Dialect Recognition Using X-Vector Speaker Embedding. In *Proceedings of the Speech and Computer*; Springer: Cham, Switzerland, 2021; pp. 50–57.
12. Peddinti, V.; Povey, D.; Khudanpur, S. A Time Delay Neural Network Architecture for Efficient Modeling of Long Temporal Contexts. In Proceedings of the Interspeech 2015, Dresden, Germany, 6 September 2015; pp. 3214–3218.
13. Snyder, D.; Garcia-Romero, D.; Sell, G.; McCree, A.; Povey, D.; Khudanpur, S. Speaker Recognition for Multi-Speaker Conversations Using X-Vectors. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5796–5800.
14. Povey, D.; Cheng, G.; Wang, Y.; Li, K.; Xu, H.; Yarmohammadi, M.; Khudanpur, S. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In Proceedings of the Interspeech 2018, Hyderabad, India, 2 September 2018; pp. 3743–3747.
15. Desplanques, B.; Thienpondt, J.; Demuynck, K. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In Proceedings of the Interspeech 2020, Shanghai, China, 25–29 October 2020; pp. 3830–3834.
16. Ali, A.; Shon, S.; Samih, Y.; Mubarak, H.; Abdelali, A.; Glass, J.; Renals, S.; Choukri, K. The MGB-5 Challenge: Recognition and Dialect Identification of Dialectal Arabic Speech. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 1026–1033.
17. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
18. 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.* **2021**, *43*, 652–662. [CrossRef] [PubMed]
19. Zhou, J.; Jiang, T.; Li, Z.; Li, L.; Hong, Q. Deep Speaker Embedding Extraction with Channel-Wise Feature Responses and Additive Supervision Softmax Loss Function. In Proceedings of the Interspeech 2019, Graz, Austria, 15 September 2019; pp. 2883–2887.
20. Lee, J.; Nam, J. Multi-Level and Multi-Scale Feature Aggregation Using Pretrained Convolutional Neural Networks for Music Auto-Tagging. *IEEE Signal Process. Lett.* **2017**, *24*, 1208–1212. [CrossRef]
21. Gao, Z.; Song, Y.; McLoughlin, I.; Li, P.; Jiang, Y.; Dai, L.-R. Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System. In Proceedings of the Interspeech 2019, ISCA, Graz, Austria, 15 September 2019; pp. 361–365.
22. Okabe, K.; Koshinaka, T.; Shinoda, K. Attentive Statistics Pooling for Deep Speaker Embedding. In Proceedings of the Interspeech 2018, Hyderabad, India, 2 September 2018; pp. 2252–2256.
23. Shon, S.; Ali, A.; Samih, Y.; Mubarak, H.; Glass, J. ADI17: A Fine-Grained Arabic Dialect Identification Dataset. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 8244–8248.
24. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. (Eds.) *The Kaldi Speech Recognition Toolkit*; IEEE Signal Processing Society: Piscataway, NJ, USA, 2011.

25. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv* **2019**, arXiv:1903.02428.
26. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
27. Sadjadi, S.O.; Kheyrkhah, T.; Tong, A.; Greenberg, C.; Reynolds, D.; Singer, E.; Mason, L.; Hernandez-Cordero, J. The 2017 NIST Language Recognition Evaluation. In Proceedings of the Speaker and Language Recognition Workshop (Odyssey 2018), ISCA, Les Sables d'Olonne, France, 26 June 2018; pp. 82–89.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.