INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

A Weight Moving Average based Alternate Decoupled Learning Algorithm for Long-Tailed Language Identification

Hui Wang[1], *Lin Liu*[2], *Yan Song*[1,*], *Lei Fang*[2], *Ian McLoughlin*[3], *Lirong Dai*[1]

[1]National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei, China.
[2]iFLYTEK Research, iFLYTEK CO., LTD, Hefei, China
[3]ICT Cluster, Singapore Institute of Technology, Singapore

wanghuii@mail.ustc.edu.cn, {linliu, leifang}@iflytek.com, {songy, ivm, lrdai}@ustc.edu.cn

Abstract

Language identification (LID) research has made tremendous progress in recent years, especially with the introduction of deep learning techniques. However, for real-world applications where the distribution of different language data is highly imbalanced, the performance of existing LID systems is still far from satisfactory. This raises the challenge of *long-tailed LID*. In this paper, we propose an effective weight moving average (WMA) based alternate decoupled learning algorithm, termed WADCL, for long-tailed LID. The system is divided into two components, a frontend feature extractor and a backend classifier. These are then alternately learned in an end-to-end manner using different sampling schemes to alleviate the distribution mismatch between training and test datasets. Furthermore, our WMA method aims to mitigate the side-effects of re-sampling schemes, by fusing the model parameters learned along the trajectory of stochastic gradient descent (SGD) optimization. To validate the effectiveness of the proposed WADCL algorithm, we evaluate and compare several systems over a language dataset constructed to match a long-tailed distribution based on real world application [1]. The experimental results from the long-tailed language dataset demonstrate that the proposed algorithm is able to achieve significant performance gains over existing state-of-the-art x-vector based LID methods.

Index Terms: deep learning, language identification, long-tailed distribution

1. Introduction

Language identification (LID) systems have been widely used as frontends for multilingual automatic speech recognition (ASR) systems to improve performance for a wide range of languages [2, 3]. Over the past decades, i-vector based systems have demonstrated their success for LID tasks [4, 5]. Existing systems typically consist of feature extractor and classifier components (as shown in Figure 1(a)), which can be learned separately with different criteria, following a decoupled learning strategy. More recently, motivated by the success of end-to-end deep neural network (DNN) models in ASR and speaker recognition [3, 6], generative i-vectors have been replaced by discriminative embeddings (*e.g.* x-vectors) [7, 8]. In this scenario, the DNN is first trained to minimize cross-entropy loss. Then the x-vectors are extracted from a hidden layer and used as inputs of a backend classifier such as a Gaussian Linear Classifier [9].

Despite the success of existing LID systems, applications in which the distribution of different language data is highly imbalanced, the performance is still far from satisfactory. This raises the challenge of deep learning based long-tailed LID, caused by the highly imbalanced distribution of different languages, *i.e.* some major languages have a large amount of data, yet only a few samples may be available for others. Specifically, with a highly imbalanced or long-tailed dataset, DNN training can be dominated by the propagating gradients of majority classes, while those from minority ones tend to be ignored. It is worth noting that long-tailed problems naturally exist in many real-world domains, including emotion recognition [10], sound event detection [11] and large scale image classification [1].

Several re-balancing methods, such as re-sampling [12], re-weighting [1] and distribution-aware learning [13, 14], have been proposed for long-tailed classification tasks. These methods improve performance by transforming the dataset into a balanced distribution. For example, under-sampling and over-sampling [12, 15] were proposed to address imbalanced distribution. However, the former potentially miss important information in majority classes, while the latter can easily result in over-fitting by repeatedly accessing data from minority classes. Re-weighting can mitigate these problems by directly adjusting the weights of different training samples to distinguish the importance of class errors. However, it is still an open question to determine how to assign the cost between individual classes.

In this paper, we propose an effective weight moving average (WMA) based alternate decoupled learning algorithm, known as WADCL, for DNN based long-tailed LID tasks, as shown in Figure 1(b). WADCL is able to learn the deep network structure in an end-to-end manner. Then, to compensate for mismatches between training and test distributions, a decoupled learning method is proposed, which learns the frontend feature extractor and backend classifier using different sampling schemes. A weight moving average (WMA) method, motivated by [16, 17], is designed to fuse model parameters along the trajectory of stochastic gradient descent (SGD) optimization. By ensembling models from different views of the original long-tailed distribution, WMA effectively mitigates the side-effects of re-sampling, such as under-fitting majority classes and over-fitting minority classes. To assess the effectiveness of WADCL, we construct a long-tailed language dataset for real-world application following the rule in [1], and then use this to evaluate a number of systems. WADCL achieves an equal error rate (EER) of 18.4%, 9.1% and 5.7% for 3s, 10s and 30s test conditions respectively. This significantly outperforms the 21.5%, 11.6% and 7.3% obtained by state-of-the-art x-vector based systems, effectively validating the WADCL algorithm.

*corresponding author.

http://dx.doi.org/10.21437/Interspeech.2021-776

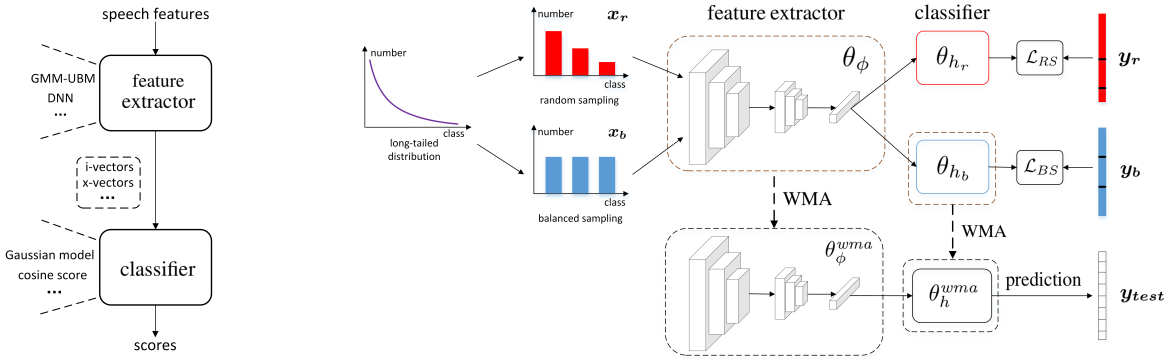

Figure 1: *(a) A conventional LID system. (b) Proposed WMA based alternate decoupled learning framework for long-tailed LID task.*

2. System description

2.1. Contribution 1: Decoupled learning

Let $\mathcal{X} = \{x_i\}, i \in \{1, \ldots, n\}$ be the language dataset, $\mathcal{Y} = \{y_i\}, i \in \{1, \ldots, n\}$ be the corresponding label set, where $|\mathcal{X}| = N$, $|\mathcal{Y}| = C$. A conventional LID system can be decoupled into two components, a frontend feature extractor $\phi(\cdot)$ that maps variable-length speech segments to an fixed-size embeddings (*e.g.* i-vector or x-vector), and a backend classifier $h(\cdot)$ that outputs decision scores. Most modern systems use a deep network to unify the two components, and directly output posterior probability of language labels from the last linear layer. Owing to end-to-end optimization, these methods demonstrate impressive recognition ability and bring excellent improvement. Typically, the network parameters are optimized by an objective loss, which takes the following form:

$$\mathcal{L} = \frac{1}{N} \sum_{c=1}^{C} \sum_{k=1}^{N_c} l(h(\phi(x_{ck})), y_{ck}) \tag{1}$$

where $l(\cdot)$ is the loss function. N_c is the number of training samples for class c.

When data is highly imbalanced, i.e., N_c varies from abundant to rare across different languages, the proportion of the majority classes in loss value is very high and the back propagation gradients of weights are dominated during iterations, leading to a decision bias. A simple solution is a re-sampling scheme, which balances the chance of sampling from all classes, to decrease the imbalance ratio. However, this also has side-effects of missing potentially useful information by under-sampling, or increasing the risk of over-fitting minority classes by over-sampling.

In fact, for imbalanced learning, such a joint optimization scheme with single data sampler, random sampler or balanced sampler, will make it unclear how good performance is achieved—is it achieved by learning a better embedding or is it achieved by readjusting the classifier decision boundaries? In [18, 19], it was found that re-sampling was effective at influencing the updating of classifier weights and paying more attention to the minority classes, to promote classifier learning of deep networks. However, re-sampling may also distort original distributions and degrade the representative ability of the learned embeddings. By contrast, simple random sampling can learn universal patterns from the original distributions and obtain better embeddings. In order to utilize the effects of different sampling schemes, we take one step back to decouple the network into two components again, and split the learning procedure into representation learning, and classifier learning. For the former, a feature extractor (backbone network) $\phi(\cdot)$ is obtained through training the model with random sampling schemes. For the latter, using the learned representation, the classifier $h(\cdot)$ is retrained with balanced sampling schemes to benefit classifier learning. Decoupled learning (DCL) thus separately manages training for both parts of the network, to achieve better overall performance for the long-tailed LID task.

2.2. Contribution 2: Weight moving averaging

The aforementioned side-effect of re-sampling schemes still exists with DCL. It does avoid disturbance from re-sampling to the backbone network, while producing more balanced decision boundaries – but is powerless to deal with the problem of information loss and over-fitting. To tackle those issues, we then propose a weight moving averaging (WMA) strategy. Such strategies have been successfully applied to other application domains [16, 17] as effective learning methods and achieve better generalization. WMA averages multiple nearby points in the weight space along the optimization trajectory of SGD, during the training procedure. Specifically, in SGD, the weights $\boldsymbol{\theta}$ are updated at each training step with one mini-batch of data. Then, over all the training steps, we capture all the weights using an, exponential moving average to get the final model $\boldsymbol{\theta}^{wma}$:

$$\boldsymbol{\theta}_t^{wma} = \alpha \cdot \boldsymbol{\theta}_{t-1}^{wma} + (1 - \alpha) \cdot \boldsymbol{\theta}_t \tag{2}$$

where α is a coefficient hyper-parameter and t is current training step. WMA can be interpreted as an approximation to ensemble learning, but in weight space. During each training step, a mini-batch of data is randomly drawn from the whole training set, in which the amount of different languages samples is balanced by re-sampling. Each mini-batch, in effect, under-samples the majority classes. But each is built from a different subset, and focuses on different perspectives of the majority classes. The model from a single training step in SGD might be one-sided with under-represented majority classes, but WMA is based on an ensemble, and can gather all models after every training step, therefore fully exploiting information of interest across the majority classes. Meanwhile, using a high constant learning rate, WMA can achieve faster and more stable convergence than SGD with a decaying schedule. Since some classes have very few samples, and the risk of overfitting increases as the learning rate decreases, this can effectively avoid overfitting

Algorithm 1 Full training algorithm of WMA based alternate decoupled learning

Input: Imbalanced training set: $\{\mathcal{X}, \mathcal{Y}\}, |\mathcal{X}| = N, |\mathcal{Y}| = C$;
Feature extractor: $\phi(\cdot)$; Classifiers: $h_r(\cdot)$, $h_b(\cdot)$;
Total iteration steps: T;
Mini-batch size: S;
Moving coefficient: α;
Output: The WMA model

Randomly initialize the weights θ_ϕ, θ_{h_r} and θ_{h_b};
Copy the weights $\{\theta_\phi, \theta_{h_b}\}$ to θ_0^{wma};
for $t \in [1, T]$ **do**
  Obtain (x_r, y_r) from the random sampler, and (x_b, y_b) from the balanced sampler;
  $\mathcal{L}_{RS} = l(h_r(\phi(x_r)), y_r)/S$;
  Optimize θ_ϕ and θ_{h_r};
  $\mathcal{L}_{BS} = l(h_b(\phi(x_b)), y_b)/S$;
  Optimize θ_{h_b};
  $\theta_t^{wma} = \alpha \cdot \theta_{t-1}^{wma} + (1-\alpha) \cdot \theta$, where $\theta = \{\theta_\phi, \theta_{h_b}\}$;
end for;
return WMA model θ_T^{wma};

the minority classes. Therefore, WMA can improve the performance on both groups. Additionally, WMA is extremely easy to implement, architecture-agnostic, and has almost no additional cost over conventional training. Compared to a conventional ensemble in model space, it can provide comparable generalization performance with a single model.

2.3. Contribution 3: WMA based alternate DCL algorithm

As discussed in Sections 2.1 and 2.2, DCL and WMA can help learn from imbalanced distributions in different ways. DCL is based on the complement of the effects of random sampling and re-sampling, while WMA directly mitigates the side-effects of re-sampling. We believe that combining WMA and DCL can contribute to better performance. Therefore, we propose a unified learning framework for the imbalanced problem as shown in Figure 1(b). Motivated by DCL, we construct two branch classifiers $h_r(\cdot)$ and $h_b(\cdot)$ for different distribution data. We apply random and balanced samplers to each of them separately and obtain two mini-batch samples (x_r, y_r) and (x_b, y_b) as the input. The backbone network is employed as the feature extractor $\phi(\cdot)$, which is shared among branches. There are two forward passes at each training step:
Firstly, (x_r, y_r), sampled by the random sampler, is sent to the feature extractor $\phi(\cdot)$ and the classifier $h_r(\cdot)$, whose weights are updated through standard cross-entropy loss with SGD;
Secondly, the feature extractor is fixed, and extracts a representation of balanced data (x_b, y_b) to train the weights of the classifier $h_b(\cdot)$.
After one training step, the weights of $\phi(\cdot)$ and $h_b(\cdot)$ are averaged according to Equation 2 to produce the final WMA model. The full WMA based alternate decoupled learning (WADCL) algorithm is described in Algorithm 1. During the test, only the WMA model is used for prediction. Therefore, compared to the standard classifier, the test model has no extra parameters.

3. Experiments

3.1. Dataset

We conduct experiments on a 6-language dataset provided by iFLYTEK, including Mandarin, MinNan, Cantonese, Taiwanese, English and Japanese. To evaluate the effectiveness of the proposed methods, we split the data into a long-tailed training dataset and a balanced test dataset. The long-tailed training dataset is constructed according to an exponential function $N_c = N_0 \mu^c$, where c is the class index (0-indexed). We use an imbalance factor β to denote the ratio between samples sizes of the most frequent class and least frequent class, i.e., $\beta = max_c\{N_c\}/min_c\{N_c\}$. β determines the severity of the imbalance and is set to 40 in our dataset. The resulting long-tailed training dataset details are given in Table 1. The test dataset is separated into 30s, 10s and 3s condition, each with 1000 utterances. As a closed-set LID task, we use accuracy and EER as the primary metrics for evaluation.

Table 1: *Statistics of the long-tailed LID training dataset.*

Language	Mandarin	English	Taiwanese
Duration (hrs)	58.0	25.7	14.0
Language	MinNan	Japanese	Cantonese
Duration (hrs)	7.8	2.6	1.4

3.2. Experimental setup

For all experiments, we use bottleneck features [20, 21] as input. An ASR acoustic model to extract 56-dimensional bottleneck features is trained using filterbank features with 25ms frames and a 10ms shift, on a 9000 hour Chinese corpus. A frame-level energy-based speech activity detector is applied to select features corresponding to speech frames. For comparison, we evaluate two baseline systems; one uses i-vectors and the other is an end-to-end deep learning baseline:

i-vector: For the i-vector system, a 2048 component full covariance GMM-UBM model is trained, along with a 300-dimensional i-vector extractor. Once extracted, the i-vectors are length normalized and their dimension reduced using linear discriminant analysis (LDA). Cosine scoring is used for final classification. The system is implemented using Kaldi toolkit [22].

ResNet: For the end-to-end system, a ResNet32 [23] model is used as the backbone network, followed by a statistics pooling [6] layer, to extract an embedding. Two fully-connected layers plus a softmax are used as classifier and directly output posterior probability of language labels. Batch normalization [24] and ReLu are added. The whole network is optimized using a standard cross-entropy loss via SGD with momentum of 0.9 and weight decay of 5e-4. The mini-batch size is set to 256. Random cropping is applied on the training data.

3.3. Results and analysis

3.3.1. Evaluation of the WADCL method

We evaluate the performance of three proposed method variants and three straightforward baselines. In addition to those systems described above, the configuration details of each of the other methods are as follows:

RS/BS: These two baselines are equivalent to the conventional supervised training using ResNet described in Section 3.2, with random sampling and balanced sampling schemes respectively. Total training steps, T, is 21000. Learning rate is initialized to 0.1 and decays by 0.1 every 7000 steps.

x-vector: Based on the ResNet model, an x-vector based DCL method is used. Concretely, a two-stage training strategy is designed to separately learn the two components of the network. In the first stage, the whole network is trained to converge

Table 2: *The performance of different methods for the three duration test conditions.*

	3s		10s		30s	
Method	Acc	EER	Acc	EER	Acc	EER
i-vector	47.7	30.9	64.8	22.5	73.2	16.4
RS	55.9	23.4	71.4	14.9	78.4	10.5
BS	57.3	22.7	73.4	13.6	79.6	10.2
x-vector	59.1	21.5	77.4	11.6	84.6	7.3
BS-WMA	60.9	20.3	77.9	10.8	84.9	7.4
WADCL	**63.1**	**18.4**	**80.8**	**9.1**	**87.9**	**5.7**

using random sampling with a decaying learning schedule. In the second stage, the x-vectors extracted from the fixed backbone network are used to re-train two fully-connected classification layers with balanced sampling.

BS-WMA: Extending the BS baseline noted above by applying WMA to average the weights of the network during iteration according to Equation 2. The coefficient hyperparameter α is set to 0.99. Training steps T is reduced to 10000 and the learning rate is fixed to 0.1.

WADCL: Keeping the internal structure of the backbone and classifier unchanged, the full WADCL algorithm presented in Algorithm 1 is implemented to combine WMA and DCL. The experimental hyperparameters are also the same as BS-WMA.

The results are presented in Table 2. As expected, end-to-end systems consistently perform better than the baseline i-vector system. Simply changing the sampling scheme, in the BS/RS systems, improves performance. BS performs slightly better since it adjusts the data distribution closer to the test. When we decouple the representation and classifier learning, x-vector based DCL achieves higher accuracy than RS or BS. Next, BS-WMA applies WMA to BS during training and tries to mitigate the side-effects, yielding a large improvement, at almost no additional cost. Finally, we combine WMA and DCL to take advantage of the effects of both. The further improvement this combination achieves confirms the complementary advantages of DCL and WMA. The proposed WADCL algorithm thus achieves best performance for all test conditions.

3.3.2. Evaluation of the WMA method

To understand and verify the effectiveness of WMA, we report more detailed results on four different experiments based on the end-to-end ResNet system. The configuration of RS, BS and BS-WMA are kept consistent with Section 3.3.1, while random sampling is used in the RS-WMA system. The results for the 30s test condition are shown in Table 3, including training loss and test accuracy. In addition to computing overall accuracy for all classes, we report the accuracy for three majority and three minority classes as listed in Table 1. This allows us to contrast the group-wise accuracy and explore the influence of each method on the imbalanced distribution.

From Table 3, it can be seen that RS achieves higher accuracy on the majority, yet worse performance on the minority. By contrast, BS improves the minority but at the expense of the majority. It is worth noting that WMA based methods achieve better and more robust performance on the test dataset, although they tend to have higher training loss. Looking at minority accuracy, BS-WMA yields a clear improvement over BS, suggesting that WMA is efficiently addressing the minority overfitting problem derived from the BS training. Furthermore, BS-WMA also boosts learning with the majority data, achieving comparable performance to RS. It has effectively mitigated most of the effects of under-representing the majority.

Table 3: *The performance of different methods, evaluated for the 30s test condition.*

Method	**Training loss** ($\times 10^{-2}$)	**Test accuracy**		
		Majority	Minority	All
RS	2.1	89.7	67.1	78.4
BS	3.5	88.3	70.9	79.6
RS-WMA	12.0	89.6	67.5	78.5
BS-WMA	14.3	89.5	80.3	**84.9**

Interestingly though, we find that RS-WMA provides little benefit over RS. There are two possible reasons. First, random sampling may harm the classifier weights by producing larger magnitudes for the majority, leading to a bias which cannot be solved in an ensemble manner. Second, every iteration of RS captures a similar view of the original dataset, negating one of the benefits that WMA provides to BS.

3.3.3. Discussion

By analyzing the evaluation results on all experiments we observe that the proposed WADCL method achieves significant improvement over the other learning methods. This gain is due to the competence of WMA at addressing the imbalanced distribution, the effective complement of WMA and DCL, and the unified end-to-end WADCL framework. Firstly, the WMA experiments of Section 3.3.2 clearly showed that promoting minority learning is the key to keeping overall performance good. As a commonly used method, BS only brings a limited improvement in this aspect. Working in the weight space, WMA can efficiently form an ensemble in one training process, with a high constant learning rate. This contributes to a substantial improvement over BS. Secondly, DCL decouples the network to focus on extracting discriminant embeddings and learning a discriminative classifier separately, and was found to be inherently complementary to WMA. Thirdly, although decoupled, WADCL introduces two branch classifiers which perform their own duties to yield a unified end-to-end framework that is able to promote more efficient learning for long-tailed data.

4. Conclusion

This paper proposed an effective learning algorithm to improve performance for long-tailed language identification. It was the first attempt at introducing the WMA into imbalanced learning. We explored how WMA effected the learning process for both the majority and minority classes. Motivated by different influence mechanisms of WMA and DCL, we then proposed a unified WADCL algorithm to combine both. This achieved significant performance gains on the long-tailed LID task.

In future we aim to evaluate WADCL on other real-world imbalanced datasets, including the Arabic dialect recognition task of the 2019 multi-genre broadcast challenge [14, 25] in which dataset class sizes vary by a factor exceeding 50.

5. Acknowledgement

This work was supported by National Natural Science Foundation of China (Grant No. U1613211), the Leading Plan of CAS (XDC08030200).

6. References

[1] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9260–9269.

[2] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, pp. 31–51, 2001.

[3] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, pp. 749–759, 2015.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[5] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *INTERSPEECH*, 2011.

[6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[7] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," *Proc. of Odyssey 2018 Speaker and Language Recognition Workshop*, pp. 105–111, 2018.

[8] X. Miao, I. McLoughlin, and Y. Yan, "A new time-frequency attention mechanism for TDNN and CNN-LSTM-TDNN, with application to language identification," in *INTERSPEECH*, 2019, pp. 4080–4084.

[9] A. Lozano-Diez, O. Plchot, P. Matejka, and J. Gonzalez-Rodriguez, "DNN based embeddings for language recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2018, pp. 5184–5188.

[10] Z. Li, L. He, J. Li, L. Wang, and W. Zhang, "Towards discriminative representations and unbiased predictions: Class-specific angular softmax for speech emotion recognition," in *INTERSPEECH*, 2019, pp. 1696–1700.

[11] T. Ashihara, Y. Shinohara, H. Sato, K. M. T. Moriya, T. Fukutomi, Y. Yamaguchi, and Y. Aono, "Neural whispered speech detection with imbalanced learning," in *INTERSPEECH*, 2019, pp. 3352–3356.

[12] X. Liu, J. Wu, and Z. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (cybernetics)*, vol. 39, pp. 539–550, 2009.

[13] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label distribution-aware margin loss," in *NeurIPS*, 2019, pp. 10–18.

[14] X. Miao and I. McLoughlin, "LSTM-TDNN with convolutional front-end for dialect identification in the 2019 multi-genre broadcast challenge," *arXiv preprint arXiv:1912.09003*, 2019.

[15] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic minority oversampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[16] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[17] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. Wilso, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[18] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9716–9725.

[19] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," *arXiv preprint arXiv:1910.09217*, 2019.

[20] B. Jiang, Y. Song, S. Wei, M. G. Wang, I. McLoughlin, and L. R. Dai, "Performance evaluation of deep bottleneck features for spoken language identification," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 143–147.

[21] Y. Song, X. Hong, B. Jiang, R. Cui, I. McLoughlin, and L. R. Dai, "Deep bottleneck network based i-vector representation for language identification," in *INTERSPEECH*, 2015.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[25] X. Miao, I. McLoughlin, and Y. Song, "Variance normalised features for language and dialect discrimination," *Circuits, Systems, and Signal Processing*, pp. 1–18, 2021.