Comparative Study on Spoken Language Identification Based on Deep Learning

Panikos Heracleous*, Kohichi Takai*, Keiji Yasuda†, Yasser Mohammad‡, Akio Yoneyama*
*KDDI Research, Inc., Japan
Email: {pa-heracleous,ko-takai,yoneyama}@kddi-research.jp
†Nara Institute of Science and Technology (NAIST), Japan
Email: ke-yasuda@dsc.naist.jp
‡Artificial Intelligence Research Center, AIST, Japan
Email: yasserm@aun.edu.eg

Abstract—**Spoken language identification is the process by which the language in a spoken utterance is recognized automatically. Spoken language identification is commonly used in speech translation systems, in multi-lingual speech recognition, and in speaker diarization. In the current paper, spoken language identification based on deep learning (DL) and the i-vector paradigm is presented. Specifically, a comparative study is reported, consisting of experiments on language identification using deep neural networks (DNN) and convolutional neural networks (CNN). Also, the integration of the two methods into a complete system is investigated. Previous studies demonstrated the effectiveness of using DNN in spoken language identification. However, to date, the integration of CNN and i-vectors in language identification has not been investigated. The main advantage of using CNN is that fewer parameters are required compared to DNN. As a result, CNN is cheaper in terms of memory and the computational power needed. The proposed methods are evaluated on the NIST 2015 i-vector Machine Learning Challenge task for the recognition of 50 in-set languages. Using DNN, a 3.55% equal error rate (EER) was achieved. The EER when using CNN was 3.48%. When DNN and CNN systems were fused, an EER of 3.3% was obtained. The results are very promising, and they also show the effectiveness of using CNN and i-vectors in spoken language identification. The proposed methods are compared to a baseline method based on support vector machines (SVM) and they demonstrated significantly superior performance.**

I. Introduction

Automatic spoken language identification is the process by which language in a spoken utterance is recognized automatically. Language identification is an important part of speech-to-speech translation systems, in multi-lingual speech recognition, and in the diarization of meetings. It can be applied in call centers to automatically route incoming calls to appropriate native speaker operators.

The current study focuses on spoken language identification based on deep learning (DL) methods and the i-vector paradigm. The first method is based on a conventional feed-forward fully connected neural network, which receives i-vectors as input features. In contrast, the second method is based on convolutional neural networks (CNNs) with i-vector features. However, few studies have presented experimental results using DL and i-vectors, and to date, the integration of i-vectors and DL for language identification has not been investigated exhaustively. Furthermore, in the current study the DNN-based method is also compared with CNN-based spoken language identification when i-vector features are used. CNNs have a different architecture from DNN, and were originally used in image recognition. Concerning spoken language identification, CNNs have been used with frame-level input features. The performance of a spoken language identification system when i-vector input features are used is, however, still an open research question.

The current study has been further expanded by investigating the performance of a spoken language identification system when DNN and CNN are fused into a complete system. In the current study, the two systems operate in parallel and the correct language is identified based on the maximum likelihood of the two individual scores.

II. Related Work

Several studies have investigated spoken language identification. The approaches presented are categorized based on the features they employ. Language identification systems are categorized into the acoustic-phonetic approach, the phonotactic approach, the prosodic approach, and the lexical approach [1]. In phonotactic systems [1], [2], sequences of recognized phonemes obtained from phone recognizers are modeled. In [3], a universal acoustic characterization approach to spoken language recognition is proposed. The main idea is to describe any spoken language with a common set of fundamental units, such as manner and articulation, which are used to build a set of language-universal attribute models. The vector space modeling-based phonotactic language recognition approach is demonstrated in [1] and presented in [4]. The key idea is to vectorize a spoken utterance into a high-dimensional vector, thus leading to a vector-based classification problem.

In acoustic modeling-based systems, however, each recognized language is modeled by using different features. Although significant improvements in LID have been achieved from phonotactic approaches, most state-of-the-art systems still rely on acoustic modeling.

In [5], an early attempt at language identification based on a neural network is presented. Similarly, neural network-

ISBN 978-90-827970-1-5 © EURASIP 2018

based language identification is addressed in [6]. In [7], the first attempt at language identification using deep learning is presented. In [8], automatic language identification based on deep neural networks (DNN) is also presented. The method shows superior performance compared to i-vector-based [9] classification schemes when a large amount of data is used. The method is compared to linear logistic regression, linear discriminant analysis-based (LDA), and Gaussian modeling-based classifiers. When limited training data are used, the i-vector yields the best identification rate. Another method based on DNN and using deep bottleneck features is presented in [10]. A method for identification in short utterances based on long short-term memory (LSTM) recurrent neural networks (RNN) is presented in [11]. In [12], the problem of language identification is addressed by using i-vectors with support vector machines (SVM) [13] and LDA. SVM with local Fisher discriminant analysis is also used in [14]. Similarly to the current study, the method is evaluated on the NIST 2015 i-vector Machine Learning Challenge task. The results obtained are very similar to those obtained in the current study when using SVM. In [15], deep neural networks-based language identification is also presented. The method is also evaluated on the NIST 2015 i-vector Machine Learning Challenge task.

III. METHODS

A. Data

In the NIST 2015 LRE i-Vector Machine Learning Challenge task, i-vectors, constructed from conversational and narrow-band broadcast speech, are given as training, testing, and development data. The task covers 50 languages and contains 15000 training i-vectors, 6500 test i-vectors, and 6431 development i-vectors. The training i-vectors are extracted from speech utterances with a mean duration of 35.15s. The training data and the test data are labeled, with the development i-vectors to be unlabeled. The set also includes i-vectors corresponding to out-of-set languages. In the current study, only the in-set languages are considered. In particular, 300 training i-vectors and 100 test i-vectors are used for each of the 50 in-set languages.

B. i-vector Paradigm

Gaussian mixture models (GMM) with universal background models (UBM) are widely used for speaker recognition. In this scenario, each speaker model is created by adapting the UBM using maximum a posteriori (MAP) adaptation. A GMM supervector is constructed by concatenating the means of the adapted model. Similar to speaker recognition, GMM supervectors can also be utilized for language identification.

The main disadvantage of GMM supervectors is the high dimensionality, which incurs high computational and memory costs. In the i-vector paradigm, the limitations of high dimensional supervectors (i.e., concatenation of the means of GMMs) are overcome by modeling the variability contained in the supervectors with a small set of factors. Considering automatic language identification, an input utterance can be modeled as:

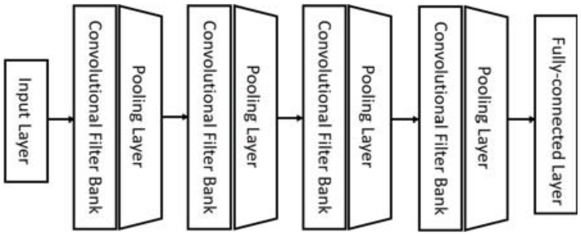
Input Layer
Convolutional Filter Bank
Pooling Layer
Convolutional Filter Bank
Pooling Layer
Convolutional Filter Bank
Pooling Layer
Convolutional Filter Bank
Pooling Layer
Fully-connected Layer

Fig. 1. The architecture of the proposed convolutional neural networks-based classifier.

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{1}$$

where $\mathbf{M}$ is the language-dependent supervector, $\mathbf{m}$ is the language-independent supervector, $\mathbf{T}$ is the total variability matrix, and $\mathbf{w}$ is the i-vector. Both the total variability matrix and language-independent supervector are estimated from the complete set of the training data.

C. Classification Approaches

1) Support Vector Machines (SVM): A support vector machine (SVM) is a two-class classifier constructed from sums of a kernel function *K(.,.)*

$$f(x) = \sum_{i=1}^{L} \alpha_i t_i K(\mathbf{x}, \mathbf{x_i}) + d \tag{2}$$

where the t_i are the ideal outputs, $\sum_{i=1}^{L} \alpha_i t_i = 0$, and $\alpha_i > 0$.

An SVM is a discriminative classifier, which is widely used in regression and classification. Given a set of labeled training samples, the algorithm finds the optimal hyperplane, which categorizes new samples. SVM is among the most popular machine learning methods. The advantages of SVM include the support of high-dimensionality, memory efficiency, and versatility. However, when the number of features exceeds the number of samples the SVM performs poorly. Another disadvantage is that SVM is not probabilistic because it works by categorizing objects based on the optimal hyperplane.

Originally, SVMs were used for binary classification. Currently, the multi-class SVM, a variant of the conventional SVM, is widely used in solving multi-class classification problems. The most common way to build a multi-class SVM is to use *K* one-versus-rest binary classifiers (commonly referred to as ”one-versus-all” or OVA classification). Another strategy is to build one-versus-one classifiers, and to choose the class that is selected by the most classifiers. In this case, *K(K-1)/2* classifiers are required and the training time decreases because less training data are used for each classifier.

2) Convolutional Neural Networks (CNN): A deep neural network is a feed-forward neural network with more than one hidden layer. The units (i.e., neurons) of each hidden layer take all outputs of the lower layer and pass them through an activation function. A convolutional neural network [16], [17] is a special variant of the conventional network, which

ISBN 978-90-827970-1-5 © EURASIP 2018

introduces a special network structure. This network structure consists of alternating convolution and pooling layers.

Convolutional neural networks have been successfully applied to sentence classification [18], image classification [19], facial expression recognition [20], and in speech emotion recognition [21]. Furthermore, in [22], bottleneck features extracted from CNN are used for robust language identification.

In the proposed CNN architecture, four convolutional layers with 64 filters and *ReLu* activation function were used. Each convolutional layer is followed by a max-pooling layer with *width* = 2. On top, a fully connected *Softmax* layer was used. The batch size was set to 64, and the dropout probability was set to 0.25. The epochs number was 200. Figure 1 shows the architecture of the proposed method.

3) Deep Neural Networks (DNN): Deep learning is behind several of the most recent breakthroughs in computer vision, speech recognition, and agents that achieved human-level performance in several games like go, poker etc. In the current study, four hidden layers with 64 units and *ReLu* activation function are used. On top, a *Softmax* layer with fifty classes is added. The number of batches is set to 512, and 500 epochs are used.

D. Evaluation measures

In the current study, the equal error rate (EER) (i.e., equal false alarms and false rejections) and the cost function are used as evaluation measures. Considering that in the current study only the in-set languages are being recognized, the cost function defined by NIST is modified as follows:

$$c_{avg} = \frac{1}{N}\sum_{k=1}^{N} P_{error}(k) \cdot 100 \qquad (3)$$

where

$$P_{error}(k) = \frac{No.\ of\ errors\ for\ class\ k}{No.\ of\ trials\ for\ class\ k} \qquad (4)$$

where N is the number of the target languages. In addition, the detection error tradeoff (DET) curves, which show the function of miss probability and false alarms, are also given.

IV. Results

These sections present the experimental results for automatic language identification using DNN, CNN, and SVM. Furthermore, the results obtained when integrating the classification methods are also presented. The experimental results show the performance of the proposed methods compared to SVM for the identification of 10, 20, 30, 40, and 50 in-set languages using the NIST 2015 LRE i-Vector Machine Learning Challenge task.

Table I shows the costs for 10, 20, 30, 40, and 50 target languages, respectively. The results demonstrate that in most cases, the lowest costs are incurred when CNN is used. For the identification of 10 target languages, the cost when using CNN was 4.6%, when using DNN the cost was 4.8%, and when using SVM a 5.4% score was obtained. In the case of identifying 50 languages, the costs were 13.7%, 13.6%, and 18.6% when using CNN, DNN, and SVM, respectively. The results obtained using DNN and CNN are very promising, and superior to those obtained in similar studies. The highest costs were incurred using SVM, demonstrating that SVM is less effective for this task.

TABLE I

Costs for different language subsets of NIST LRE 2015

No of Languages	Classification method		
	DNN	CNN	SVM
10	4.8	**4.6**	5.4
20	10.0	**9.7**	10.7
30	**11.1**	**11.1**	13.9
40	13.4	**13.3**	15.7
50	**13.6**	13.7	18.6

TABLE II

Equal error rates (EER) for different language subsets of NIST LRE 2015

No of Languages	Classification method		
	DNN	CNN	SVM
10	**1.89**	2.00	2.42
20	**3.30**	**3.30**	4.13
30	**3.02**	**3.02**	4.47
40	**2.95**	3.06	4.65
50	3.55	**3.48**	5.20

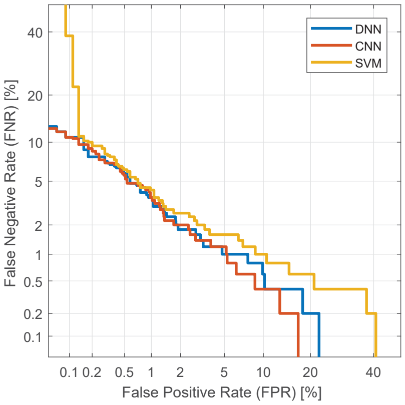

Fig. 2. DET curves for ten target languages.

Table II shows the EER when using the five subset target languages. As shown, when using the CNN- and DNN-based methods, the lowest EER is obtained. For the identification of 10 target languages, EERs of 2.0%, 1.89%, and 2.42% are obtained when using CNN, DNN, and SVM, respectively. In the case of 50 target languages, the EERs are 3.48%, 3.55%, and 5.2% when using CNN, DNN, and SVM classifiers, respectively. The results show that CNN- and DNN-based methods have higher robustness in terms of EER in relation to the number of target languages. Figure 2 and Figure 3 show the DET curves in the case of 10 and 50 target languages, respectively. As shown, superior performance is obtained using the proposed approaches. The graphs also show

ISBN 978-90-827970-1-5 © EURASIP 2018

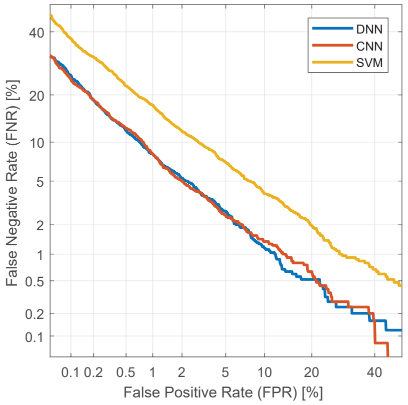

Fig. 3. DET curves for the fifty target languages.

TABLE III
COST USING TRAINING DATA OF DIFFERENT SIZES

No. of training i-vectors	Classification method		
	DNN	CNN	SVM
2500	26.7	**23.8**	27.9
5000	22.3	**20.8**	23.1
7500	**19.6**	20.0	22.0
10000	**18.0**	19.1	20.9
12500	18.0	**17.7**	20.1
15000	**13.6**	13.7	18.6

TABLE IV
EQUAL ERROR RATES (EER) USING TRAINING DATA OF DIFFERENT SIZES

No. of training i-vectors	Classification method		
	DNN	CNN	SVM
2500	5.9	**5.5**	8.3
5000	4.7	**4.5**	6.7
7500	4.3	**4.2**	6.4
10000	**3.9**	4.0	6.1
12500	**3.8**	3.9	5.8
15000	3.6	**3.5**	5.2

that when CNN and DNN are used, the performance is highly comparable, and superior to SVM.

To investigate the effect of training data size when using the three classifiers, an experiment was conducted using training data of reduced size. Table III shows the costs incurred in this case. As shown, the DNN and CNN classifiers show the lowest costs, followed by the SVM. The results also show that by increasing the training data, a lower cost is incurred. When using 50 training vectors per language, costs of 23.8%, 26.7%, and 27.9% costs are incurred in the case of CNN, DNN, and SVM, respectively. By increasing the number of training i-vectors, the costs decrease.

Table IV shows the EERs when a reduced amount of training data is used. As shown, in most cases the lowest EERs are obtained when using CNN. In the case of 50 training i-vectors per language, the EERs are 5.5%, 5.9%, and 8.3% using CNN, DNN, and SVM, respectively.

TABLE V
EQUAL ERROR RATES (EER) USING FUSION OF DIFFERENT CLASSIFIER

No. of languages	Classification method		
	DNN+CNN	DNN + SVM	CNN + SVM
10	1.80	1.91	1.87
50	3.39	2.84	2.79

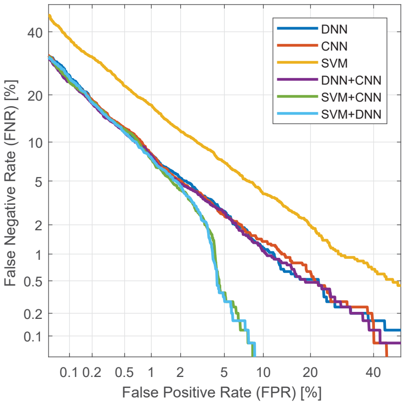

Fig. 4. DET curves for the fifty target languages when classifiers are fused.

The results obtained are very promising and show the effectiveness of using deep learning with i-vectors for language identification. Furthermore, the results justify the use of CNN along with i-vectors for this task, even in the case of limited training i-vectors.

In the current study, the performance of a language identification system when several classifiers are fused was also investigated. The classifiers operate in parallel and the language is hypothesized based on the classifiers' individual scores. Considering two classifiers, the total S score is computed as follows:

$$S = \alpha s_A + (1 - \alpha) s_B \tag{5}$$

where s_A and s_B are the scores obtained from the two classifiers, and α is a weight. In these experiments, the weight was adjusted empirically to 0.5 (i.e., equal importance of the two classifiers).

Table V shows the EERs when the classifiers are integrated in a parallel mode, and the final decision is made by adding the two individual scores. As shown, by using fusion, lower EERs are obtained compared when classifiers are used alone. In the case of 10 target languages, the lowest EER is obtained when CNN and DNN classifiers are integrated. In this case, an EER of 1.80% is achieved. In the case of 50 target languages, the lowest EERs are obtained when SVM is integrated with DNN and CNN. Specifically, when SVM is integrated with CNN the EER is as low as 2.79%. These results are interesting and promising and show the effectiveness of using fusion in spoken language identification compared to single classifiers. Figure 4

ISBN 978-90-827970-1-5 © EURASIP 2018

shows the DET graphs in the case of 50 target languages. As shown, when fusion is used, superior performance is obtained.

V. Discussion

This study focused on automatic language identification using the NIST 2015 LRE i-Vector Machine Learning Challenge task. Although many studies have investigated this area, problems remained. Recent advances in classification and feature extraction methods have resulted in significant improvements in identification rates. Modern approaches include the use of the i-vector paradigm along with the very popular and widely used SVM classifiers. Other studies focus on deep neural networks or other conventional approaches such as Gaussian mixture models and supervector-based identification methods. In the current study, spoken language identification based on CNN and DNN classifiers and i-vectors was experimentally investigated. The very few studies addressing language identification based on this framework, left this research area wide open. Moreover, in the current study CNN and i-vectors were integrated and compared using a DNN-based approach. A limitation of the current study was the very small amount of data used in the experiments. For each target language, 300 training vectors were used, and this may not be a sufficient number for high performance using deep learning-based classifiers. On the other hand, the proposed method was evaluated on the NIST 2015 LRE task, and the available data are still sufficient to demonstrate the effectiveness of the proposed methods in spoken language identification.

VI. Conclusion

In this study, two methods based on deep learning for language identification were experimentally investigated. Furthermore, the integration of different classifiers was also addressed. The methods are based on DNN and CNN using i-vector input features and were evaluated on the NIST 2015 LRE i-Vector Machine Learning Challenge task. For the identification of the 50 in-set languages, EERs of 3.6% and 3.5% were obtained using DNN and CNN, respectively. When CNN was fused with SVM, an EER of 2.79% was achieved. When DNN was fused with SVM, an EER of 2.84% was obtained. The results are promising and demonstrate the effectiveness of CNN and DNN in spoken language identification when i-vectors are used. Furthermore, the fusion of different classifiers resulted in additional improvements.

Acknowledgments

The authors would like to thank Professor Emeritus Péter Mihály, a former Professor of Eötvös Loránd University, Faculty of Humanities, Russian Department, Hungary, for fruitful discussions and his suggestions concerning current advances and future directions in spoken language identification.

References

[1] H. Li, B. Ma, and K. A. Lee, Spoken language recognition: From fundamentals to practice," *in Proc. of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[2] M. A. Zissman, Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4(1), pp. 31–44, 1996.

[3] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, Universal attribute characterization of spoken languages for automatic spoken language recognition," *Computer speech and language*, vol. 27, pp. 209–227, 2013.

[4] D. A. Reynolds, W. M. Campbell, W. Shen, and E. Singer, Automatic Language Recognition Via Spectral and Token Based Approaches," *in Springer Handbook on Speech Processing and Speech Communication, J. Benesty, Y. Hunag. M. M. Sondhi, Editors, SpringerVerlag*, 2008.

[5] R. Cole, J. Inouye, Y. Muthusamy, and M. Gopalakrishnan, Language identification with neural networks: a feasibility study," *in Proc. of IEEE Pacific Rim Conference*, pp. 525–529, 1989.

[6] M. Leena, K. Srinivasa Rao, and B. Yegnanarayana, Neural network classifiers for language identification using phonotactic and prosodic features," *in Proc. of Intelligent Sensing and Information Processing*, pp. 404–408, 2005.

[7] G. Montavon, Deep learning for spoken language identification," *in NIPS workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[8] I. L.-Moreno, J. G.-Dominguez, O. Plchot, D. Martinez, J. G.-Rodriguez, and P. Moreno, Automatic Language Identification Using Deep Neural Networks," *in Proc. of ICASSP*, pp. 5337–5341, 2014.

[9] N. Dehak, P. J. 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(4), pp. 788–798, 2011.

[10] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V.McLoughlin, and L.-R. Dai, Deep Bottleneck Features for Spoken Language Identification," *PLos ONE*, vol. 9(7), pp. 1–11, 2010.

[11] R. Zazo, A. L.-Diez, J. G.-Dominguez, D. T. Toledano, and J. G.-Rodriguez, Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks," *PLos ONE*, vol. 11(1): e0146917., 2016.

[12] N. Dehak, P. A.T.-Carrasquillo, D. Reynolds, and R. Dehak, Language Recognition via Ivectors and Dimensionality Reduction," *in Proc. of Interspeech*, pp. 857–860, 2011.

[13] N. Cristianini and J. S.-Taylor, Support Vector Machines," *Cambridge University Press, Cambridge*, 2000.

[14] P. Shen, X. Lu, L. Liu, and H. Kawai, Local Fisher Discriminant Analysis for Spoken Language Identification," *in Proc. of ICASSP*, pp. 5825–5829, 2016.

[15] S. Ranjan, C. Yu, C. Zhang, F. Kelly, and J. H. L. Hansen, Language recognition using deep neural networks with very limited training data," *in Proc. of ICASSP*, pp. 5830–5834, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. 2012, pp. 1097–1105, Curran Associates, Inc.

[17] O. Abdel-Hamid, A.R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1533–1545, 2014.

[18] Y. Kim, Convolutional Neural Networks for Sentence Classification," *in Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, 2014.

[19] W. Rawat and Z. Wang, Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review ," *Neural Communication*, vol. 29, pp. 23522449, 2017.

[20] X.-P. Huynh, T.-D. Tran, and Y-G. Kim, Convolutional Neural Network Models for Facial Expression Recognition Using BU-3DFE Database ," in *Information Science and Applications (ICISA) 2016. Lecture Notes in Electrical Engineering*, K. Kim and N. Joukov, Eds., vol. 376, pp. 441–450. Springer, 2013.

[21] W. Lim, D. Jang, and T. Lee, Speech Emotion Recognition Using Convolutional and Recurrent Neural Networks," *in Proc. of Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016.

[22] S. Ganapathy, K. Han, S. Thomas, M. Omar, M. V. Segbroeck, and S. S. Narayanan, Robust Language Identification Using Convolutional Neural Network Features," *in Proc. of Interspeech*, 2014.