An End-to-end Approach to Language Identification in Short Utterances using Convolutional Neural Networks

Alicia Lozano-Diez, Ruben Zazo-Candil, Javier Gonzalez-Dominguez, Doroteo T. Toledano, Joaquin Gonzalez-Rodriguez

ATVS-Biometric Recognition Group, Universidad Autonoma de Madrid, Madrid, Spain
{alicia.lozano, ruben.zazo} @uam.es

Abstract

In this work, we propose an end-to-end approach to the language identification (LID) problem based on Convolutional Deep Neural Networks (CDNNs). The use of CDNNs is mainly motivated by the ability they have shown when modeling speech signals, and their relatively low-cost with respect to other deep architectures in terms of number of free parameters. We evaluate different configurations in a subset of 8 languages within the NIST Language Recognition Evaluation 2009 Voice of America (VOA) dataset, for the task of short test durations (segments up to 3 seconds of speech). The proposed CDNN-based systems achieve comparable performances to our baseline i-vector system, while reducing drastically the number of parameters to tune (at least 100 times fewer parameters). Then, we combine these CDNN-based systems and the i-vector baseline with a simple fusion at score level. This combination outperforms our best standalone system (up to 11% of relative improvement in terms of EER).

1. Introduction

The Language Identification (LID) task consists of automatically recognizing which language is being spoken in a given utterance [1]. LID is daily used in different and varied applications such as interaction with devices in different languages [2] or in emergency call centers, where the rapid detection of the speaker language might be critical.

In the last years, the LID problem has been addressed following the i-vector scheme [3], which is also within state-of-the-art approaches for speaker recognition tasks [4]. However, the performance of i-vector based approaches significantly decreases when dealing with short test utterances [5]. Recently, systems based on deep learning approaches such as feed forward Deep Neural Networks (DNNs) or Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs) have demonstrated to outperform i-vector based approaches [6].

However, DNN-based systems need huge training datasets in order to be successful [5]. Furthermore, their training is computationally expensive and they still have a large number of parameters to be trained.

In this work, we propose the use of Convolutional Deep Neural Networks (CDNNs) since they have several advantages over other architectures. Their structure based on sharing weights among hidden units in order to extract the same features from different locations, reduces drastically the number of parameters to tune, in comparison with other deep architectures. Moreover, they have been succesfully applied to other related tasks [7] [8], and, recently, to LID [9] [10]. Unlike [9], where a CDNN trained for automatic speech recognition was used to replace the UBM in an i-vector based approach for LID, we propose using CDNNs as an end-to-end system. Our CDNN-based systems are trained from the begining to discriminate among a set of given languages and, thus, there is no need of a previous speech recognition stage.

We evaluate the performance of different CDNN-based systems on a subset of the NIST Language Recognition Evaluation 2009 (LRE'09), testing on short test utterances. We compare our proposal with an i-vector based system. The CDNN systems obtain comparable results to the i-vector approach, having much less free parameters. Further, the performance improves when a simple fusion is performed.

The rest of this paper is organized as follows. In Section 2 we describe the proposed CDNN systems and how we apply them to LID. The experimental framework used in this work is presented in Section 3. Section 4 is devoted to describe the obtained results both for individual and fusion systems. Finally, Section 5 presents the conclusions of this work.

2. Convolutional Networks for LID

2.1. Convolutional Network Architecture

Convolutional neural networks are a type of neural network where each hidden layer is split into two parts: convolutional layer and subsampling layer [11].

The convolutional layer performs feature extraction. Each unit in this layer is connected to a local subset of units in the hidden layer below, according to a given filter shape. It computes its activation by convolving the input with a linear filter (weights, W), adding a bias term (b) and applying a non-linear transformation (in our case, $\tanh$):

$$h = \tanh(W * x + b)$$

Moreover, groups of these units spatially related share their parameters and form what is called a feature map, since they extract the same features from different locations in the input. This sharing of parameters also decreases the number of free parameters of the whole network.

On the other hand, subsampling layer reduces the size of the representations obtained by convolutional layers. In our case, this phase is based on partitioning the input into non-overlapping regions (according to a given pool shape) and choosing the maximum activation of each region (*max-pooling*). This subsampling also makes the network invariant to small translations and rotations [12].

This structure makes convolutional networks easier to train than other DNNs, by using well-known supervised algorithms such as gradient descent [11]. Moreover, they are smaller in

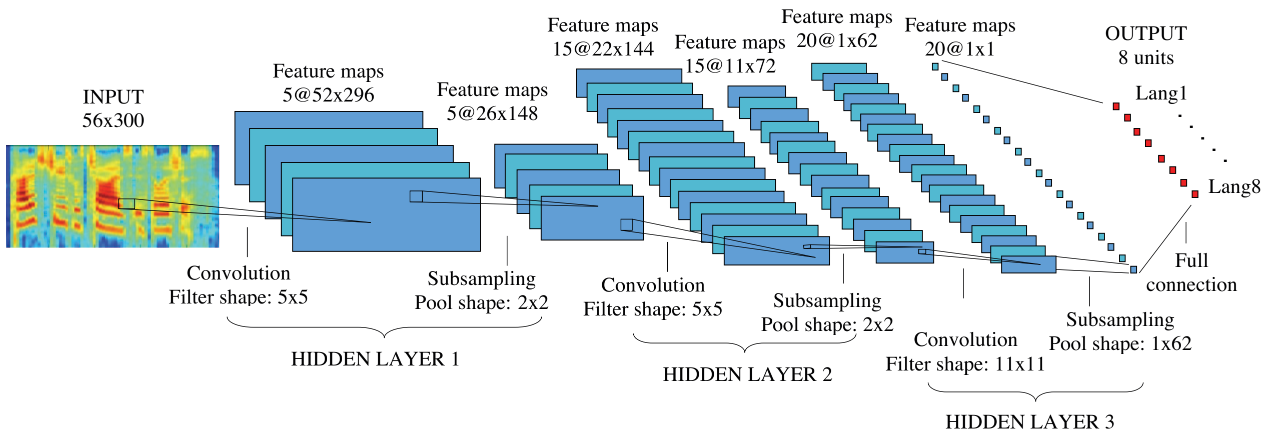


Figure 1: Representation of architecture used in the experimental part of this work with three hidden layers of 5, 15 and 20 filters respectively, and to discriminate among 8 languages. The other models used have the same structure but varying the number of filters in each layer.

terms of number of free parameters than the i-vector systems used typically for the problem of language identification.

Figure 1 shows an example of structure used in the experimental part of this work.

2.2. Proposed System

The details of the CDNN-based systems are as follows. We used speech segments of 3 seconds, which correspond with 300 frames, since we applied windows of 20 ms of duration with 10 ms of overlap. Then, we represented each frame with a vector of 56 MFCC-SDCs (with the configuration 7-1-3-7) [13] and fed the network with a 2-dimensional matrix of dimensions 56×300, corresponding with a given speech segment of 3 seconds long. Finally, we normalized the input to have zero mean and unit variance for each coefficient over the whole training set. Moreover, in order to suppress silences, we used a voice activity detector based on energy. This last filtering process made test segments contain less than 3 seconds of actual speech, which was a problem since the network input dimensions are constant. It was solved by applying a right padding by using the first frames of the segment to fit this requirement.

We built different networks depending on the number of filters (feature maps) considered for each hidden layer, which is related to the idea of how many different features are to be extracted in each layer. However, all of them have 3 hidden layers composed of the two stages mentioned in Section 2.1: convolution and subsampling. They have also in common the shape of the linear filters (5×5 for the first two hidden layers, and 11×11 for the third one) and the max-pooling regions (with a shape of 2×2 in the first two hidden layers, and 1×62 in the third one in order to have a single value as output of the last hidden layer).

We also evaluated different amounts of data to develop each network in order to study its influence in the performance (178h, 356h or 534h). These training sets are composed of approximately the same number of hours for each language involved in the experiment.

The differences among the structures are summarized in Table 1.

The output layer consists of a fully-connected layer that computes a softmax function according to the following expression:

	Configuration	**Development Data**	
ID	# Filters/Layer	Train	Validation
ConvNet 1	[20, 30, 50]	∼178h	∼31h
ConvNet 2	[5, 15, 20]	∼178h	∼31h
ConvNet 3	[5, 15, 20]	∼356h	∼63h
ConvNet 4	[5, 15, 20]	∼534h	∼63h
ConvNet 5	[10, 20, 30]	∼356h	∼63h
ConvNet 6	[10, 20, 30]	∼534h	∼63h

Table 1: Configuration parameters for the developed models.

$$P(Y = i|x, W, b) = softmax_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$

where i is a certain language, and W and b are the parameters of the model (weights and bias, respectively).

Then, the network outputs the probability that the test segment belongs to a certain language, among the languages involved in the experiment.

Regarding the training of the network, the algorithm that was used is the stochastic gradient descent algorithm with a learning rate of 0.1 and based on minibatches of 500 samples. The cost function that the algorithm tries to minimize is the negative log-likelihood, defined as follows:

$$NLL(\theta, D) = -\sum_{i=1}^{|D|} \log P(Y = y^{(i)}|x^{(i)}, \theta)$$

where D is the dataset, θ represents the parameters of the model ($\theta = W, b$, weights and bias respectively), $x^{(i)}$ is an example and $y^{(i)}$ its corresponding label, and P is defined as the output of the softmax function defined above.

Also, an "early stopping" technique was used during the training in order to avoid overfiting, so, each iteration, the performance of the model was evaluated in a validation set, and if the improvements over that set were not considered relevant, the training stopped.

All this development was done by using Python and, specifically, Theano [14], following the ideas of [15].

3. Experimental Framework

3.1. Dataset Description

The database used to perform the experiments was that provided by NIST in the Language Recognition Evaluation 2009 (LRE'09).

LRE'09 database includes data coming from different audio sources: conversational telephone speech (CTS), used in previous evaluations, and broadcast data containing telephone and non-telephone speech. That broadcast data consist of two corpora from Voice of America (VOA) broadcast in multiple languages (VOA2 and VOA3). Some language labels of VOA2 might be erroneous since they have not been audited. More details can be found in [16].

Both language and audio source labels were distributed to participants. In this work, just data belonging to VOA were considered in order to avoid unbalanced data from different sources (CTS and VOA).

The database includes data from 40 languages (23 target and 17 out-of-set). From them, we selected 8 languages as in [6]: US English (Eng), Spanish (Spa), Dari (Dar), French (Fre), Pashto (Pas), Russian (Rus), Urdu (Urd) and Chinese Mandarin (Chi).

Regarding evaluation data, segments of 3, 10 and 30 second of duration from CTS and broadcast speech data were available to test the developed systems. However, the experiments shown in this paper are focused on segments of 3 seconds (short duration), where i-vector systems obtain lower performances. Our test dataset includes 2942 test segments from the 8 languages mentioned before. Thus, we perform a closed-set task, without out-of-set test utterances.

3.2. Baseline i-vector System

In order to have a baseline to compare with, an i-vector based system was evaluated on the same test dataset.

The i-vector system is based on GMMs where a Total Variability (TV) modelling strategy is employed in order to model both language and session variability [17]. First, an Universal Background Model (UBM) composed of 1024 Gaussian components is trained from MFCC-SDC parameterization of the audio, with the configuration 7-1-3-7. Then, Baum-Welch statistics are computed over this UBM, and a TV space of 400 dimension is derived from them by using PCA followed by 10 EM iterations. All the process, from the parameterization of the audio to the i-vector computation has been done using Kaldi [18].

Regarding the classification stage, we used the classical cosine scoring scheme. Thus, given a test utterance i-vector w, and the i-vector model w_L (computed as the mean i-vector from all the utterances of the language L), the cosine similarity is computed as follows:

$$S_{w,w_L} = \frac{\langle w, w_L \rangle}{||w|| ||w_L||}$$

The classical Linear Discriminant Analysis (LDA) classification scheme gave us slightly lower performance than just the cosine scoring scheme in our experiments. This could be explained because we just used 8 languages in our experiments. Thus, if we used LDA, just 7 dimensions would remain from the i-vectors, and we could be losing information useful for discrimination.

		Performance	
ID	Size	EER_{avg} (%)	C_{avg}
i-vector	∼23M	**16.94**	**0.1535**
ConvNet 1	∼198k	22.14	0.2406
ConvNet 2	∼39k	25.90	0.2700
ConvNet 3	∼39k	24.69	0.2616
ConvNet 4	∼39k	23.48	0.2461
ConvNet 5	∼78k	21.60	0.2282
ConvNet 6	∼78k	21.11	0.2293
AllConvNets	-	17.93	**0.1836**
ConvNet 6+i-vector	-	**15.96**	**0.1433**
AllConvNets+i-vector	-	**15.04**	**0.1360**

Table 2: Individual and combined systems performance.

3.3. Performance Evaluation

The performance of the systems was evaluated according to two different metrics.

The first one is the cost measure C_{avg}, defined in the NIST LRE'09 evaluation plan [16]. This measure takes into account the false alarm and false rejection probabilities and the cost of a bad classification of the speech segment. Therefore, it evaluates the ability of the system for discrimination and calibration (i.e., the capacity of setting optimal thresholds).

Secondly, the classical Equal Error Rate (EER, in %) was considered. As we deal with a multiclass task in this work, we compute the EER for each individual system, and average them to obtain an EER_{avg} as a metric for the performance of the whole system.

Furthermore, we present the confusion matrix of our best system, typically used when assessing the performance in a multiclass classification task. With this matrix, we show the discriminative capacity of the system and the confusion among all the languages involved in our experiments.

4. Results

The experiments presented in this work are based on the 8 languages mentioned in Section 3.1. For the experiments based on CDNNs, we split the development data into two disjoint datasets (training and validation) in order to perform training and model selection with different data. The amount of data used for each CDNN-based system can be seen in Table 1.

The performances of standalone systems and combined systems are summarized in Table 2. In this table we can also see the size (in terms of number of parameters to be trained) of each system.

In order to calibrate and combine the systems, we use multiclass logistic regression from FoCal toolkit [19]. Its training was done using the evaluation scores themselves.

4.1. Individual Systems

As we can see in Table 2, the performance of the i-vector system is better than the one obtained by the standalone CDNN-based systems. However, the size of these models is between ∼100 and ∼600 times smaller with respect to the number of parame-

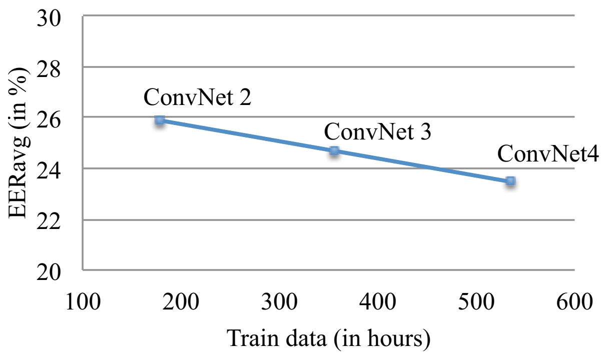
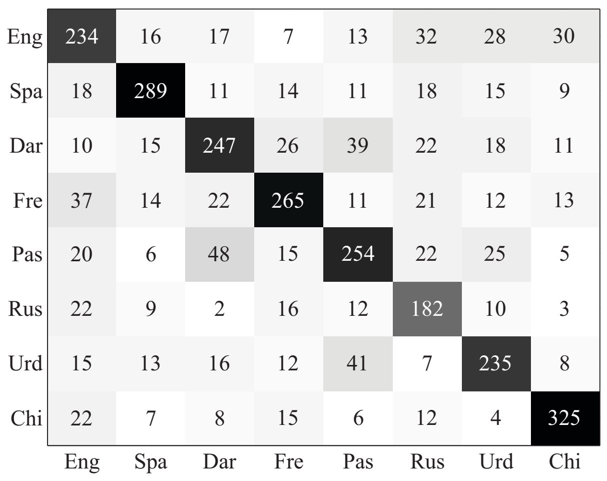


Figure 2: Influence of the amount of data used for training in the performance of the system (in terms of EER_{avg}). Note that the three convolutional networks shown in this graphic have the same topology (ConvNet 2, 3 and 4, with [5, 15, 20] filters per layer).



	Eng	Spa	Dar	Fre	Pas	Rus	Urd	Chi
Eng	234	16	17	7	13	32	28	30
Spa	18	289	11	14	11	18	15	9
Dar	10	15	247	26	39	22	18	11
Fre	37	14	22	265	11	21	12	13
Pas	20	6	48	15	254	22	25	5
Rus	22	9	2	16	12	182	10	3
Urd	15	13	16	12	41	7	235	8
Chi	22	7	8	15	6	12	4	325

Figure 3: Confusion matrix corresponding to the fusion of all CDNN-based systems and our baseline i-vector system.

ters that need to be tuned in the i-vector system. In our case, the baseline i-vector system presented in Section 3.2 has ∼23M of parameters, which is given by the number of Gaussian components of the UBM (1024), the feature space dimensionality (56 MFCC-SDCs) and the i-vector dimensions (400). In contrast, our biggest CDNN-based model is ∼100 times smaller (∼198k parameters). Moreover, the datasets used to train the CDNN systems are actually smaller than the one composed of 200h per language that we used to train the i-vector system. Therefore, the CDNN systems extract useful discriminant information even with less data and much less parameters. If we compare the different CDNN-based systems, we can see that the more data we introduce, the better the performance (see Figure 2).

Furthermore, increasing the number of filters per layer (and, thus, the size of the model) yields better performance even if the amount of data used to train remains constant (compare ConvNet 3 and 5 or ConvNet 4 and 6 in Table 1).

It should be hightlighted that increasing the number of parameters or the amount of data means higher cost in terms of time and memory needed to train the CDNN-based systems, although it is slightly noticeable in the testing stage. Nevertheless, their size is much smaller than the i-vector approach and they need less resources to be stored and tested.

4.2. Fusion Systems

The first kind of fusion we present in this work is the combination of the CDNN models. As it is shown in Table 2 (see All-ConvNets row), with this fusion we obtain comparable performance to i-vector system (17.93% of EER_{avg} versus 16.94%), and a ∼15% of relative improvement (in EER_{avg}) with respect to the best standalone CDNN system (ConvNet 6). From this result, we can draw the conclusion that even using the same type of architecture, varying just the number of filters per layer and the training dataset, CDNNs are able to extract complementary information.

On the other hand, when fusing the best CDNN system with the i-vector system, the combination outperforms our baseline by ∼6% in terms of EER_{avg}. Moreover, when fusing all the CDNN models with the baseline i-vector system, this relative improvement reaches up to a 11%. The confusion matrix for this last combination can be seen in Figure 3.

5. Conclusions

In this work, we proposed an end-to-end CDNN-based approach to the problem of LID and we tested it in segments of less than 3 seconds of speech (short duration).

The proposed models are lighter (in terms of number of parameters) than traditional approaches based on i-vectors and other deep learning architectures. Thus, using at least 100 times fewer parameters, our proposed systems obtain performances comparable to the i-vector baseline system.

Furthermore, by combining our CDNN-based systems and the baseline i-vector system, we obtain a relative performance improvement of ∼11%. This means our proposed systems extract information that complements the one extracted by our i-vector system.

6. Acknowledgments

This work has been supported by project *CMC-V2: Caracterización, Modelado y Compensación de Variabilidad en la Señal de Voz* (TEC2012-37585-C02-01), funded by *Ministerio de Economía y Competitividad*, Spain.

7. References

[1] Y. Muthusamy, E. Barnard, and R. Cole, “Reviewing automatic language identification,” *Signal Processing Magazine, IEEE*, vol. 11, no. 4, pp. 33–41, 1994.

[2] 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,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. PP, no. 99, pp. 1–1, 2014.

[3] M. Penagarikano, A. Varona, M. Diez, L. J. Rodriguez-Fuentes, and G. Bordel, “Study of different backends in a state-of-the-art language recognition system.” in *INTERSPEECH*, 2012.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-End Factor Analysis for Speaker Verification,” *Audio, Speech, and Language Processing, IEEE*

Transactions on, vol. 19, no. 4, pp. 788 – 798, February 2011.

[5] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic Language Identification using Deep Neural Networks," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 2014.

[6] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, 2014, pp. 2155–2159.

[7] M. McLaren, Y. Lei, N. Scheffer, and L. Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, 2014.

[8] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[9] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. ODYSSEY*, 2014.

[10] A. Lozano-Diez, J. Gonzalez-Dominguez, R. Zazo, D. Ramos, and J. Gonzalez-Rodriguez, "On the use of convolutional neural networks in pairwise language recognition," in *Advances in Speech and Language Technologies for Iberian Languages*. Springer, 2014, pp. 79–88.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Intelligent Signal Processing*. IEEE Press, 2001, pp. 306–351.

[12] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009, also published as a book. Now Publishers, 2009.

[13] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, and J. R. Deller, "Approaches to Language Identification Using Gaussian Mixture Models and Shifted Delta Cepstral Features," in *ICSLP*, vol. 1, 2002, pp. 89–92.

[14] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.

[15] LISA, *Deep Learning Tutorial*, University of Montreal, http://deeplearning.net/tutorial/.

[16] NIST, "The 2009 NIST SLR Evaluation Plan," www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf, 2009.

[17] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language Recognition via i-vectors and Dimensionality Reduction." in *INTERSPEECH*. ISCA, 2011, pp. 857–860.

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[19] N. Brümmer. Fusion and calibration toolkit [software package]. [Online]. Available: , http://sites.google.com/site/nikobrummer/focal.