Interspeech 2018
2-6 September 2018, Hyderabad, India

Learning Discriminative Features for Speaker Identification and Verification

Sarthak Yadav[1], *Atul Rai*[1]

[1]Staqu Technologies, India

sarthak.yadav@staqu.com, atul.rai@staqu.com

Abstract

The success of any Text Independent Speaker Identification and/or Verification system relies upon the system's capability to learn discriminative features.

In this paper we propose a Convolutional Neural Network (CNN) Architecture based on the popular Very Deep VGG [1] CNNs, with key modifications to accommodate variable length spectrogram inputs, reduce the model disk space requirements and reduce the number of parameters, resulting in significant reduction in training times. We also propose a unified deep learning system for both Text-Independent Speaker Recognition and Speaker Verification, by training the proposed network architecture under the joint supervision of Softmax loss and Center loss [2] to obtain highly discriminative deep features that are suited for both Speaker Identification and Verification Tasks.

We use the recently released VoxCeleb dataset [3], which contains hundreds of thousands of real world utterances of over 1200 celebrities belonging to various ethnicities, for benchmarking our approach. Our best CNN model achieved a Top-1 accuracy of 84.6%, a 4% absolute improvement over VoxCeleb's approach, whereas training in conjunction with Center Loss improved the Top-1 accuracy to 89.5%, a 9% absolute improvement over Voxceleb's approach.

Index Terms: speaker identification, speaker recognition, speaker verification, convolutional neural network, discriminative feature learning

1. Introduction

With the advent of large datasets that have significant applications in a multitude of real world scenarios, Deep Learning, spearheaded by Convolutional Neural Networks, has ascended as the go-to approach in the fields of computer vision [4], [5], speech recognition [6], [7] and other related fields due to their inherent capability to deal with real world, noisy datasets without the need for handcrafted feature engineering.

Text independent Speaker Recognition in unconstrained conditions is a challenging problem, due to extrinsic and intrinsic variations; extrinsic variations being background chatter, music, laughter, reverberations, faulty recording device and distortions caused by the transmission medium; whereas intrinsic variations include age, accent, emotion and intonation of the speaker, among others. [8]

The innate capability to learn discriminative features is crucial for any Speaker Recognition system to perform well. Recent works have focused on utilizing bottleneck features from DNNs [9], training deep embeddings for utilization by a PLDA backend [10] or training an end-to-end Deep CNN embedding using Constrastive Loss [3] or Triplet Loss [11]. These approaches require training a very diverse set of networks that try to minimize diverse sets of objective functions, which requires considerable training efforts and compute.

Recently introduced VGG family of CNNs [1] demonstrated that very deep CNNs, consisting solely of small 3x3 convolution filters achieved drastic improvement over previous baselines on the ImageNet [4] competition. This success was attributed largely to the usage of smaller receptive fields (3x3 kernel) as well as increased network depth, which is facilitated by the parameter efficient nature of 3x3 kernels.

In this paper, we propose a new Convolutional Neural Network Architecture based on VGG Config-A and Config-B [1], owing to their aforementioned characteristics, with key modifications to accommodate variable length spectrogram input along with a reduction in network parameters and storage space requirements.

The question then arises: since the tasks of Speaker Identification and Speaker Verification have the same underlying objective, which is, identifying a previously known speaker as well as learning to discriminate between distinct speakers, wouldn't it be beneficial to have a unified approach that is capable of solving both the problems? To this end, we also propose a unified deep learning system for both Text-Independent Speaker Recognition and Speaker Verification which trains the CNN under the joint supervision of Softmax loss and Center loss [2], thus mitigating the need for training distinct networks for the two tasks.

We utilize the Voxceleb [3] dataset for quantifying and comparing the performance of our proposed approach on both Speaker Identification and Verification. Voxceleb dataset is a large scale, gender balanced dataset comprising of over 140,000 utterances belonging to 1251 distinct celebrities of different ethnicities, in real-world conditions.

2. Related Works

Speaker recognition is a domain where Gaussian Mixture Models (GMMs) dominated the field for quite some time ([12], [13]), with Joint Factor analysis (JFA) [14] and i-vector based methods [15] surpassing them in more recent times. All the above mentioned methods rely upon low dimensional Mel Frequency Cepstrum Coefficients (MFCC) as input features. MFCCs are known to suffer from performance degradation under real world noise conditions as demonstrated by [16], [17], which has paved the way for the recent shift to Deep Convolutional Neural Networks for various speech based applications [18], [19], [20].

Speaker Recognition comprises of two subtasks: Speaker Identification and Speaker Verification. Usually, the methodologies for training networks pertaining to the two aforementioned tasks are different, where identification (in a closed set) is usually treated as a n-way classification problem and a classification model based on Softmax Loss is trained for the same [3], whereas Speaker Verification, which involves determining whether there is a match between a given utterance and a target model, is solved by training a discriminative embedding ([10], [11]) or using bottleneck features from classification models [9].

Recently, [19] examined various CNN architectures for

10.21437/Interspeech.2018-1015

Acoustic Scene Classification, establishing the effectivity of direct analogs of CNNs used for image classification in classification of acoustic scenes. By contrast, we have proposed an architecture developed specifically for Speaker Recognition, which is a much fine-grained classification task, as compared to acoustic scene classification. Also, the network architectures they evaluated, being direct altercations of prevalent Image Classification CNNs, couldn't handle variable length inputs, an important characteristic to have for Speaker Recognition.

[20] studied the optimal CNN design for speaker identification and clustering, as well as elaborated on how to apply transfer learning, viz., transfer a network trained for speaker identification to speaker clustering. However, as compared to the proposed architecture, their CNN architecture didn't support variable length input either.

3. Proposed Approach

The following subsections describe the proposed approach:

3.1. Model Architecture

Based on their recent success in a multitude of Computer Vision tasks, and relatively simplistic design as compared to more recent networks ([21], [22], [23]) we decided to base our CNNs on the VGG ConvNet config A and B (VGG 11 and VGG 13) networks as proposed in [1], with modifications to accomodate to the input features and provide support for variable length input. Our proposed network utilizes Batch Normalization [24] after every Conv-ReLU [25] pair. Inspired by [21], [22], we remove the fully connected layers right after the CNN feature extractor. The output features from the convolutional layers (final maxpool layers) are collapsed over the feature map dimensions and then averaged. This temporal averaging of CNN features yields a low dimensional vector allowing efficient and dense aggregation of variable length input, thus enabling the network to accomodate variable length inputs effectively. This is followed by a bottleneck layer of n dimensions (FC-ndims), on which center loss is applied (3). This setup drastically reduces the number of parameters of the network from 134M to 9.6M for the VGG 13 based network, much lower than the best performing network from [3] with ≈67M parameters. Other variations of the network with intermediate Fully connected layers were also tried, but offered no improvements. To aid with overfitting, Dropout [26] with a drop rate of 40% was applied before and after the bottleneck layer.

Network A and Network B are based on VGG-config A and VGG config B respectively (Table 1). ReLU and Batch Normalization layers are not shown in the table for clarity. We use only 2-D Convolutional Layers with 3x3 kernels, with both stride and padding equal to 1. Except the first Max Pooling layer, which has a 3x3 kernel and stride of 2, all the Max Pooling layers have a 2x2 kernel and stride of 2.

Our network design choices also lead to the following desirable characteristics of the proposed architecture:

1. **Parameter Efficiency**: The network only utilizes small 3x3 kernels. As shown in [1], 3x3 convolutions are much more parameter efficient as compared to 7x7 or 5x5 convolutions, with a 7x7 kernel requiring 81% more parameters.

2. **Model Size**: Most of the parameters of the original VGG Config-A and Config-B networks reside in their large fully connected layers. Latest CNN architectures have demonstrated that large FC layers are not a prerequisite for improved classification performance ([21], [22], [23]). Therefore, we decided to remove these FC layers. This reduced the number of parameters drastically, hence decreasing model disk space requirements as well as speeding up inference.

Table 1: *Proposed CNN Architecture*

Network A	Network B
based on VGG config-A	based on VGG config-B
input (1, 161, 301)	
conv3-64	conv3-64 **conv3-64**
maxpool $k = (3, 3), s = (2, 2)$	
conv3-128	conv3-128 **conv3-128**
maxpool	
conv3-256 conv3-256	conv3-256 conv3-256
maxpool	
conv3-512 conv3-512	conv3-512 conv3-512
maxpool	
conv3-512 conv3-512	conv3-512 conv3-512
maxpool	
Feature Averaging (batch, 512)	
dropout	
FC-$ndims$ (batch, ndims)	
dropout	
FC-1,251	
soft-max	

3.2. Feature Extraction

MFCC features demonstrate degraded performance under noisy conditions [16], [17], as well as by focusing only on the overall spectral envelope of short frames, MFCCs may be lacking in speaker-discriminating features (such as pitch information) [3]. Log-powered spectrograms, which have been popularized in recent Speech works ([3], [11], [27]), do not possess the aforementioned shortcomings of MFCCs, and therefore are used instead. Following the feature extraction process in VoxCeleb[3], all audio is first converted to single-channel, 16-bit streams at a 16kHz sampling rate. Spectrograms are then generated in a sliding window fashion using a Hamming window and mean and variance normalization. However, unlike VoxCeleb, we use a window width of 20ms, step size of 10ms and a 160 point fft. This yields us a spectrogram of size 1x161x301 for a 3 second audio clip, unlike VoxCeleb, which had a much larger spectrogram of size 1x512x300 for the same clip. This results in much lower memory footprint and helps us train large CNNs on a single Titan-Z GPU. Python's popular `LibROSA` module was used for feature extraction.

3.3. Joint Supervision Objective Loss function

In order to maximize the discriminative power of the CNN, we propose to train the network under the joint supervision of Softmax Loss and Center Loss [2]. Center Loss was originally proposed for Face Recognition tasks, where it performed remarkably on various benchmark datasets for face recognition such as

Labelled Faces in the Wild (LFW), Youtube Faces (YTF) and the MegaFace challenge. It was also demonstrated that the features learned in this manner worked on par with those learned using alternative Deep Metric Learning techniques, such as Triplet Loss or Constrastive Loss. The intuition behind Center Loss was that minimizing intra-class variations while keeping features of different classes separable was key to learning discriminative features. Center loss can be formulated as:

$$L_c = \frac{1}{2}\sum_{i=1}^{n} \|bottleneck_i - c_{y_i}\|_2^2 \quad (1)$$

where $c_{y_i} \in R^d$ denotes the y_i^{th} class center of deep features and $bottleneck_i$ denote the bottleneck features corresponding to the ith instance.

Since we adopt a joint supervision of softmax and center loss, the objective function can be formulated as:

$$L = L_s + \lambda L_c \quad (2)$$

Where L_s stands for the Softmax Multiclass Classification Loss and L_c stands for Center Loss. From (1) and (2):

$$L = -\sum_{i=1}^{n} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_{y_j}}} + \frac{\lambda}{2}\sum_{i=1}^{n} \|bottleneck_i - c_{y_i}\|_2^2 \quad (3)$$

where the scalar λ is used for balancing the two loss functions. In all our experiments, $\lambda = 5$ unless stated otherwise.

3.4. Implementation and Training Details

All our work was done using the PyTorch Deep Learning Framework, and training was performed on an NVIDIA Titan-Z GPU. Adam [28] optimizer was used for training the networks with default hyper-parameter values. To reduce overfitting, we augment the data by taking random 3-second crop in the time domain, just like [3], along with random noise injection. Using a fixed length input at training time reduces the memory footprint and computational requirements.

Identification vs Verification: The network architecture and training methodology (such as the loss function) is identical for both Identification and Verification tasks, unlike [3].

Testing: The network has an inherent capability of handling variable length sequences, courtesy of feature averaging prior to using fully connected layer(s).

4. Experiments

This section describes the experimental setup for both speaker identification and verification tasks, and compares the performance of the proposed approach with a number of methods whose performance has already been benchmarked in [3].

4.1. Experimental Setup

Speaker Identification: For Speaker Identification the Person Of Interests (POIs) for both training and testing remains the same (all 1,251 distinct speakers, Table 2). Therefore the task was treated as straight forward multi-class classification across 1,251 classes. The development/test split used was as provided by [3]. For each POI, speech segments from one video is reserved for test. For identification, *top-1* and *top-5* accuracies are reported.

Speaker Verification: Following [3], all POIs whose name starts with an 'E' are reserved for testing, leaving out 1,211 POIs for training (Table 3) Therefore, networks for the Verification task were trained as 1,211-way multiclass classification problem using the objective function as given in equation 2. At test time, the bottleneck features, $bottleneck_a$ and $bottleneck_b$ are calculated for the test pair (a, b) respectively, and cosine distance is used to measure the similarity between $bottleneck_a$ and $bottleneck_b$ vectors. We use Equal Error Rate (EER), a popular performance metric for Speaker Verification, to quantify the performance of the network on the test set.

Table 2: *Speaker Identification data set statistics*

Set	#POIs	# Vid.//POI	#Utterances
Dev	1,251	17.0	139,124
Test	1,251	1.0	6,255

Table 3: *Speaker Verification data set statistics*

Set	#POIs	# Vid.//POI	#Utterances
Dev	1,211	18.0	140,664
Test	40	17.4	4,715

4.2. Baselines

We compare our results with the following baselines, all of which were already evaluated in [3].

GMM-UBM: The GMM-UBM system uses MFCCs of dimension 13 as input. Cepstral mean and variance normalisation (CMVN) was applied on the features. Following the conventional GMM-UBM framework, a single speaker-independent universal background model (UBM) of 1024 mixture components was trained for 10 iterations from the training data.

I-vectors/PLDA: Gender independent i-vector extractors [15] were trained on the VoxCeleb dataset to produce 400-dimensional i-vectors. Probabilistic LDA (PLDA) [29] is then used to reduce the dimension of the i-vectors to 200.

Inference: For identification, a one-vs-rest binary SVM classifier was trained for each speaker m ($m \in 1...K$). All feature inputs to the SVM were L2 normalised and a held out validation set was used to determine the C parameter (determines trade off between maximising the margin and penalising training errors). Classification during test time was done by choosing the speaker corresponding to the highest SVM score. The PLDA scoring function [29] was used for verification.

VoxCeleb's Approach: VoxCeleb proposes a CNN architecture based on the VGG-M [30] CNN, with appropriate modifications to adapt to the spectrogram input. Spectrograms were generated using a sliding window of width 25ms, step 10ms and a 1024-point FFT, giving a spectrogram of size 512x300 for 3 second speech clip. Speaker Identification is treated as a straightforward multi-class classification task, whereas a Siamese network is trained with contrastive loss, which requires considerable training efforts.

4.3. Results

Results are provided in Tables 4 and 5, respectively.

Table 4: *Speaker Identification Results on VoxCeleb*

Accuracy	Top-1%	Top-5%
I-vectors + SVM	49.0	56.6
I-vectors + PLDA + SVM	60.8	75.6
CNN-fc-3s	72.4	87.4
CNN	80.5	92.1
Network A (ndims=128)	83.5	93.8
Network B (ndims=128)	**84.6**	**94.1**
Network A (ndims=128), Joint	88.3	96.8
Network B (ndims=128), Joint	**89.5**	**97**

Table 5: *Speaker Verification Results on VoxCeleb*

Metrics	EER%
GMM + UBM	15.0
I-vectors + PLDA	8.8
CNN-1024D	10.2
CNN-256D Embedding	7.8
Network B (ndims=128)	4.9

Table 4 provides the results on the Speaker Identification Task. The first four entries are baselines as evaluated in [3]. The proposed CNNs, Network A and Network B are trained using Softmax Loss and then again under the joint supervision of Softmax loss and Center Loss (marked Joint in the table) as given by (Equation 3). Both CNN networks outperform the existing benchmarks by a large margin, with Network-B standing out as the best performing network, for both softmax-only and joint training.

For identification, the Network-B CNN architecture with $ndims = 128$, trained using joint supervision performed the best, achieving a *top-1* classification accuracy of 89.5% over 1,251 POIs, which is an absolute improvement of 9% over the previous best result of 80.5% ("CNN" entry, Table 4).

Table 5 provides the results on the Speaker Verification Task. The first four entries are baselines as evaluated in [3]. Due to resource constraints, we only evaluate Network B, with different values of bottleneck dimensions. All the networks for verification were trained under the joint supervision of softmax loss and center loss. Therefore, as compared to [3], where they applied transfer learning by fine-tuning a classification model using constrastive loss using complex hard negative mining, our unified approach results in better performance with significantly shorter training times, without the need for complex pair mining techniques.

For verification, Network-B with $ndims = 128$ achieves an EER of 4.9 %, an absolute improvement of ≈3.0% over the previous best (CNN-256D Embedding, Table 5), a significant improvement considering the 50% reduction in embedding size, hence demonstrating the proposed training methodology's ability to improve the network's capability to learn discriminative features.

5. Conclusions

In this paper, we propose an end-to-end Deep Learning system using Convolutional Neural Networks (CNNs), trained under the joint supervision of Softmax loss and Center loss to obtain highly discriminative deep features suitable for both Text-Independent Speaker Recognition and Speaker Verification. Center loss was originally proposed for Face Recognition tasks, where it established its capability to improve CNNs capability to learning discriminative features. Our results demonstrate that networks trained using the proposed methodology outperform the current baselines on both Speaker Identification and Verification tasks, with much lesser number of parameters, establishing the effectiveness of the proposed approach for Text-independent Speaker Recognition as well.

6. References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[3] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[4] 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. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[6] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[7] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv preprint arXiv:1701.02720*, 2017.

[8] L. L. Stoll, *Finding difficult speakers in automatic speaker recognition*. University of California, Berkeley, 2011.

[9] J. Jorrin, P. Garcia, and L. Buera, "Dnn bottleneck features for speaker clustering," in *Proc. Interspeech 2017*, 2017, pp. 1024–1028. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-144

[10] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-620

[11] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[12] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[13] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[14] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, vol. 14, pp. 28–29, 2005.

[15] 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, no. 4, pp. 788–798, 2011.

[16] U. Yapanel, X. Zhang, and J. H. Hansen, "High performance digit recognition in real car environments," in *Seventh International Conference on Spoken Language Processing*, 2002.

[17] J. H. Hansen, R. Sarikaya, U. Yapanel, and B. Pellom, "Robust speech recognition in noise: an evaluation using the spine corpus," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[18] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[19] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[20] Y. Lukic, C. Vogt, O. Dürr, and T. Stadelmann, "Speaker identification and clustering using convolutional neural networks," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. IEEE, 2016, pp. 1–6.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[23] G. Huang and Z. Liu, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[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] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[30] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.