PAN
ARCHIVES OF ACOUSTICS
Vol. **45**, No. 4, pp. 573–583 (2020)
DOI: 10.24425/aoa.2020.134073

Research Paper

SpeakerNet for Cross-lingual Text-Independent Speaker Verification

Hafsa HABIB, Huma TAUSEEF*, Muhammad Abuzar FAHIEM
Saima FARHAN, Ghousia USMAN

Lahore College for Women University
Jail Road, Lahore, Punjab, Pakistan 54000
*Corresponding Author e-mail: humaiftikhar@hotmail.com

(*received February 6, 2020; accepted August 15, 2020*)

Biometrics provide an alternative to passwords and pins for authentication. The emergence of machine learning algorithms provides an easy and economical solution to authentication problems. The phases of speaker verification protocol are training, enrollment of speakers and evaluation of unknown voice. In this paper, we addressed text independent speaker verification using Siamese convolutional network. Siamese networks are twin networks with shared weights. Feature space can be learnt easily by training these networks even if similar observations are placed in proximity. Extracted features from Siamese then can be classified using difference or correlation measures. We have implemented a customized scoring scheme that utilizes Siamese' capability of applying distance measures with the convolutional learning. Experiments made on cross language audios of multi-lingual speakers confirm the capability of our architecture to handle gender, age and language independent speaker verification. Moreover, our designed Siamese network, SpeakerNet, provided better results than the existing speaker verification approaches by decreasing the equal error rate to 0.02.

Keywords: Convolutional Neural Network; Deep learning; Siamese network; speaker verification; text-independent; binary operation; Urdu speaker recognition.

1. Introduction

Biometrics consider physical or behavioral characteristics to identify a person. Physiological attributes relate to physical features of a person e.g. voice, palm, iris and face. Whereas, behavioral attributes are influenced by social and environmental factors related to activities of a person. Biometric Pattern recognition techniques work by extracting patterns from selected human trait like voice or finger prints into a digital signature. This signature is later used to recognize or verify a person. The use of biometrics is increasing widely to secure access to high profile places, data, services and other procedures. They offer convenience and efficiency in routine tasks and reduce chances of frauds and impersonation as biometric traits are difficult to imposter. While pass phrases and Personal Identification Number (PINs) are easily forgetTable and vulnerable to hackers. Even knowledge-based authentication questions can be answered just by knowing the person. With the emergence of Voice over Internet Protocol (VoIP) technology and mobile phones, voice is the most reachable biometric trait. It provides a relatively cheap and safer way of authentication as compared to passwords and pins. Voice features depend on physical attributes such as mouth, lips and nasal cavities that are used to speak. These are invariant for an individual but behavioral attribute may affect it. Voice recognition techniques are generally categorized as (1) automatic speaker verification and (2) automatic speaker recognition.

Speaker verification can be subcategorized into text dependent and text independent speaker verification. In text dependent verification, all speakers utter the same word or sentence then speaker is verified using features that are invariant to speech and specific to the speaker. In text independent scenario, speaker utterances are not limited to special words. It is more challenging to extract text independent speaker specific information in speech signal.

With the advent of machine learning and deep learning and their rapid adaptation to new problems, researchers have focused their research on text independent speaker verification. The speaker verification protocol consists of three phases: training, enrollment of speakers and evaluation of unknown voice. In

training, a universal background model is created to learn speaker representations. In enrollment phase, all speakers are listed in trained model and their respective speaker models are generated. Lastly in evaluation phase, the speaker utterances are matched with the previously saved model of claimed speaker for verification.

A speaker verifier can be efficiently built by training a Convolutional Neural Network (CNN) on speaker audios to extract embeddings. The embeddings are then used to train a Siamese network with twin fully connected layers which accepts two distinct speaker embeddings that are either similar or dissimilar. The sharing of weights and parameters between the twin networks guarantees that two extremely similar voices could not possibly be mapped to very different locations in feature space by their respective networks. The reason is that each network computes the same function. Extracted features from Siamese then can be classified using difference or correlation measures.

The main contribution of this paper is the use of binary operations in Siamese with convolutional layers, SpeakerNet. The proposed model can learn generic voice features and many other related properties that cover even minute inconsistency in voices from the training data. This learning capability contrasts with the methods based on hand crafted features. In contrary to other one-shot verification tasks (Koch *et al.*, 2015), the problem with voice audios is much complicated owing to subtle variations in speaker audios independent of scripts. These variations can also take in some degrees of forgery. Here we reduce that possibility by averaging the speaker embeddings.

The rest of the paper is organized as follows: in Sec. 2 we discuss previous studies in this domain. Sections 3 presents our proposed method SpeakerNet and its architecture. Moreover Sec. 4 illustrates our experimental validation and compares the proposed method with available state-of-the-art algorithms. Finally, in Sec. 4, we conclude the paper with a defined future direction.

2. Literature review

Traditionally, mathematical modeling techniques such as Gaussian Mixture Model-Universal Background Model (GMM-UBM) (Reynolds *et al.*, 2000) and I-vector (Dehak *et al.*, 2011) in combination with similarity measures such as log likelihood ratio have been used in biometrics and speaker verification successfully. Authors in (Czyżewski *et al.*, 2017; 2019; Szczuko *et al.*, 2019) have successfully implemented a similar multimodal biometric verification system called IDENT that utilizes face, voice and signature prints.

Mathematical speaker verification techniques such as I-vectors, show poor performance if the new enrolled speaker utterances are short. As these are unsupervised models, it is difficult for them to learn supervised speaker discriminative features hence they are not suiTable for verification process. This drawback is overcome by using any supervised model like Support Vector Machine (SVM) with Gaussian Mixture Model-Universal Background Model (GMM-UBM) (Campbell *et al.*, 2006) or Probabilistic Linear Discriminant Analysis (PLDA)-based I-vectors model (Lei *et al.*, 2014b) which showed promising results.

The popularity of deep learning techniques is evident as it is used in vast range of applications; such as disease identification (Shen *et al.*, 2015), automatic speech recognition (Lei *et al.*, 2014a; Hinton *et al.*, 2012), image recognition (Krizhevsky *et al.*, 2012) and network sparsity (Torfi, Shirvani, 2018). Several Deep Neural Network (DNN) based approaches have been proposed for Automatic Speaker Recognition (ASR) (Huang *et al.*, 2016; Lei *et al.*, 2014a). CNNs are also very popular in speech recognition and speaker verification (Hinton *et al.*, 2012, Shi *et al.*, 2018) inspired by their better performance in action recognition (Ji *et al.*, 2013) and scene understanding (Tran *et al.*, 2015). Authors in (Mobiny, Najarian, 2018), suggested Long Short-Term Memory (LSTM) Networks for speaker verification based on short utterances which provided good accuracy. Deep learning is state of the art problem solver for many prediction problems. We also investigated the use of transfer learning of CNN and Siamese networks in speaker verification. DNNs are used to learn continuous features called embeddings from categorical data. This provides additional benefit of low dimensionality by using only meaningful features. Authors in (Torfi, Shirvani, 2018; Shi *et al.*, 2018) extracted CNN embeddings and averaged them out to create speaker models. CNN requires very large data to be trained effectively that is why transfer learning provides better results. Transfer learning is a technique in machine learning that trains the network on one task and then uses the trained model to do another task that is relatively similar to the previous one. Transfer learning has shown good results in image classification (Zhang *et al.*, 2017), speech recognition (García-Salinas *et al.*, 2019; Huang *et al.*, 2016), ASR and verification (Lei *et al.*, 2014a; Hong *et al.*, 2017), medical image classification (Hermessi *et al.*, 2019) and face verification (Cao *et al.*, 2013).

Siamese like networks gained popularity for various tasks, such as, online signature verification (Bromley *et al.*, 1994), speaker verification (Soleymani *et al.*, 2018), face verification (Chopra *et al.*, 2005; Schroff *et al.*, 2015), one-shot image recognition and sketch-based image retrieval task (Qi *et al.*, 2016). However, to the best of our knowledge, till date, convolutional layers instead of distance calculation on top of Siamese has never been used for speaker verification.

Table 1. Datasets for speaker recognition.

Dataset	Year	Languages	Text Independent	# Speakers	# Utt*/speech recorded
Voxceleb (Nigrani *et al.*, 2017)	2017	English	yes	6,112	1,128,246/–
NIST SRE Corpora (Martin, Greenberg, 2010)**	2012	US English	yes	2000+	**
Fisher (Cieri *et al.*, 2004)	2004	US, Canadian English	yes	unknown	–/2742 hrs
RSR 2015 (Larcher *et al.*, 2014)	2014	Singaporean English	no	300	–/151 hrs
Deep Mine (Zeinali *et al.*, 2018)	2018	Persian, English	yes	1355	360 000/–
SITW (McLaren *et al.*, 2016)	2016	English	yes	299	2800/–
MIT Mobile (Woo *et al.*, 2006)	2006	–	no	88	7884/–
ANDOSL (Morrison *et al.*, 2012)	2012	Australian English	yes	204	33900/–
MGB Challenge Dataset (Bell *et al.*, 2015)	2015	English	yes	unknown	–/1600 hrs
MOBIO (McCool *et al.*, 2012)	2012	English	yes	150	–/61 hrs

* # Utt – Number of utterances, hrs – hours,

** NIST SRE corpora continually increase its size and #utterances over the years.

There are multiple datasets collected for speaker recognition and can be used in speaker verification as well. Details of some state-of-the-art datasets are given in Table 1.

3. Methodology

3.1. Data preprocessing

Data is preprocessed to convert it to a standard format before passing to the model. All audio files are converted into .wav format. In the next step each audio frame is classified as voice or not voice also known as Voice Activity Detection (VAD) (Ramirez *et al.*, 2007). Only voice parts of audio are passed to the next step. Sample conversion rate of 16 is applied on all audios which gives us 16 values per second. The audio signal values are then normalized between −1 to 1. Then the signal is divided into non overlapping frames each of 0.96 seconds and decomposed with a Short Time Fourier Transform (STFT) by applying 25 millisecond window with 10 millisecond step size. The resulting spectrogram is integrated into 64 Mel-spaced frequency bins, and the magnitude of each bin is log transformed after adding a small offset to avoid numerical issues and zero logs. This gives log-Mel spectrogram patches of 96×64 bins that form the input to the Neural Network (NN). Spectrogram represents signal strength and loudness in spectrum over time in various frequencies. Spectrograms are also called voiceprints or voicegrams. Spectrograms have varying types for example, STFT spectrograms and log-Mel spectrograms. Spectrograms usually work better with NN as compared to cepstrogram and chromograms (Korvel *et al.*, 2018).

3.2. CNN and Siamese Networks

Deep CNNs are multilayer NNs consisting of more than one convolutional layer having different stride and kernel sizes. The convolutional NN is known to be layered architecture. The transition of input from top layer to a bottom one is achieved by utilizing both the differentiable function and the neuron weight shared between those layers. Input at each layer is down sampled before going to next layer. The spectrogram of the shape 64×96 is passed to the input layers and spatial resolution of input is reduced by the kernel applied in convolution. This way it can find more generalized and abstract features of input data. Nonlinearity is introduced by using Rectified Linear Units (ReLU) activation function. Parameters and weight dimensions are reduced by using pooling layers in the network, an $n \times n$ pooling layer returns a single winning value for all $n \times n$ blocks. Pooling also helps in network robustness to deal with noise. Large datasets are passed to the network in batches which may results in overfitting according to the specific batch. This effect is neutralized by using batch normalization layers in between convolutional layers. The weights of the multiple layers are updated by applying backpropagation in the network. Several optimization techniques e.g., Adam optimizer is used in optimizing weights in backpropagation.

Siamese networks are special purpose networks that share two identical networks. These networks have same shape, parameters and configuration. Parameters are updated simultaneously in both networks during training. This framework has been successfully applied in verification problems like face verification and signature verification (Bromley *et al.*, 1994; Chopra *et al.*, 2005). These subnetworks are merged by a loss

function to compute similarity scores of the features calculated by each network. Contrastive loss is the most widely used loss function in Siamese networks (Chopra *et al.*, 2005). Unlike traditional approaches, binary labels are not assigned immediately to the outputs rather Siamese networks work in a fashion that brings similar impostor and original inputs together and push the dissimilar pairs far away from each other. If we see each branch of Siamese network as a function to map inputs into a space, this loss function has the property to map the different embeddings far from each other into the spaces while keeping the same embeddings near to each other. Both networks are joined with a merge layer. In order to decide if two audios belong to the same class, one needs to determine threshold value for the merged layer.

3.3. Proposed approach

Our proposed approach uses a CNN trained for speaker classification to extract speaker embeddings for the verification. It reconfigures the original Visual Geometry Group (VGG) NN (Simonyan, Zisserman, 2014) by changing its final layer and adding batch normalization to get audio embeddings. The fifth convolutional block is also removed thus reducing the total number of parameters from 144M weights and 20B multiplies to 62M weights and 2.4B multiplies. The final network architecture, illustrated in Fig. 1, consists of four convolutional blocks having convolution and max pooling layers. Followed by a fully connected block having two dense layers and an embedding layer. More details about this network can be found in (Hershey *et al.*, 2017).

Parametric values of filter size, height and width, strides and padding are provided in Table 2. In CNN, stride indicates the distance between subsequent samples achieved by applying convolutional and max pooling filters. ReLU activation is applied in all layers, activation decides whether a neuron should fire or not. Padding indicates the type of pixels applied on the boundary of the input so that

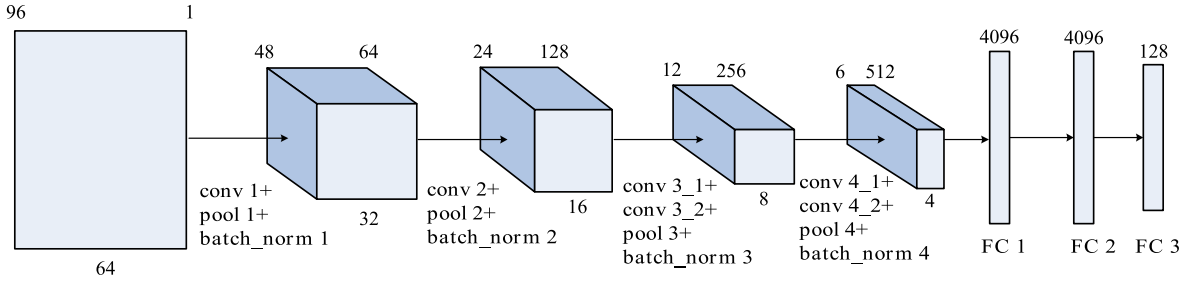

Fig. 1. CNN architecture to extract embeddings.

Table 2. CNN architecture.

Layers	CNN	Output	Parameters
Input	$64 \times 96 \times 1$	–	strides = 1, padding = 'same'
Conv 1	$64 \times 3 \times 3$		strides = 1, padding = 'same'
Pool 1	2×2	$32 \times 48 \times 64$	strides = 2, padding = 'same'
Batch_norm 1	–		–
Conv 2	$128 \times 3 \times 3$		strides = 1, padding = 'same'
Pool 2	2×2	$16 \times 24 \times 128$	strides = 2, padding = 'same'
Batch_norm 2	–		–
Conv 3_1	$256 \times 3 \times 3$		strides = 1, padding = 'same'
Conv 3_2	$256 \times 3 \times 3$		strides = 1, padding = 'same'
Pool 3	2×2	$8 \times 12 \times 256$	strides = 2, padding = 'same'
Batch_norm 3	–		–
Conv 4_1	$512 \times 3 \times 3$		strides = 1, padding = 'same'
Conv 4_2	$512 \times 3 \times 3$		strides = 1, padding = 'same'
Pool 4	2×2	$4 \times 6 \times 512$	strides = 2, padding = 'same'
Batch_norm 4	–		–
FC 1	4096	4096	–
FC 2	4096	4096	–
FC 3	128	128	–

conv – convolutional layer, pool – pooling layer,
FC – fully connected layer, batch_norm – batch normalization layer.

filters cover whole input. Same padding is applied to input in all layers so that output has same length as original input.

First convolutional layer filters the 96×64 input Mel spectrogram with 64 kernel windows of size 3×3. The second convolutional layer takes batch normalized and pooled output of first layer and apply 128 kernel windows of size 3×3. Third convolutional block has two consecutive convolutional layers followed by pooling batch normalization layer having 256 kernel windows of size 3×3 each. Forth convolutional block is connected to the third one with 2 convolutional layers of 512, 3×3 kernel windows each followed by pooling and batch normalization layers. This brings about the NN learning more high-level abstract features and less low-level specific features. The output of last convolutional layer is flattened and pass to the first fully connected layer having 4096 nodes. The second fully connected layer also has 4096 nodes. Finally, last fully connected layer has 128 nodes. This indicates that each audio is converted into embedding of size 128 each. We initialized the weights of model according to the work of (Hershey *et al.*, 2017). The model is trained using Adam optimizer for 10 epochs. Training starts with learning rate of 0.003 with hyper parameter epsilon 1e–8. All these values are shown in Table 3.

Table 3. Parameter values for Adam Optimizer.

Parameter	Value
Learning rate	0.003
Weight decay	0.0
Epsilon	Le–8
decay	0.0

The framework is implemented in Keras[1] library using Tensorflow[2] backend. The model is trained using GPU on cloud Google Colaboratory[3].

3.4. SpeakerNet

The trained model is used to generate embeddings for each speaker audio. Average is calculated for embeddings of 10 seconds for each speaker. These embeddings are passed to the Siamese network. Architecture for SpeakerNet is illustrated in Fig. 2.

Each branch of Siamese has fully connected layer of 128 nodes followed by dropout and batch normalization. The outputs of these branches are merged after taking dot product, sum, difference and square of difference as shown in Eqs (1), (2), (3), and (4), respectively.

$$\text{dot product} = \overrightarrow{\text{Original}} \otimes \overrightarrow{\text{Imposter}}, \tag{1}$$

$$\text{sum} = \overrightarrow{\text{Original}} \oplus \overrightarrow{\text{Imposter}}, \tag{2}$$

$$\text{difference} = \overrightarrow{\text{Original}} \ominus \overrightarrow{\text{Imposter}}, \tag{3}$$

$$\text{sq. of difference} = \left(\overrightarrow{\text{Original}} \ominus \overrightarrow{\text{Imposter}}\right)^2. \tag{4}$$

Output of above mentioned binary operations is stacked on top of each other, that makes a block of $4 \times 128 \times 1$. Convolution is applied on stacked output of last layer with 16 kernels of size 4×1 each. The output is again reshaped into $1 \times 16 \times 1$. Second con-

[1] http://keras.io/
[2] https://www.tensorflow.org/
[3] https://colab.research.google.com/

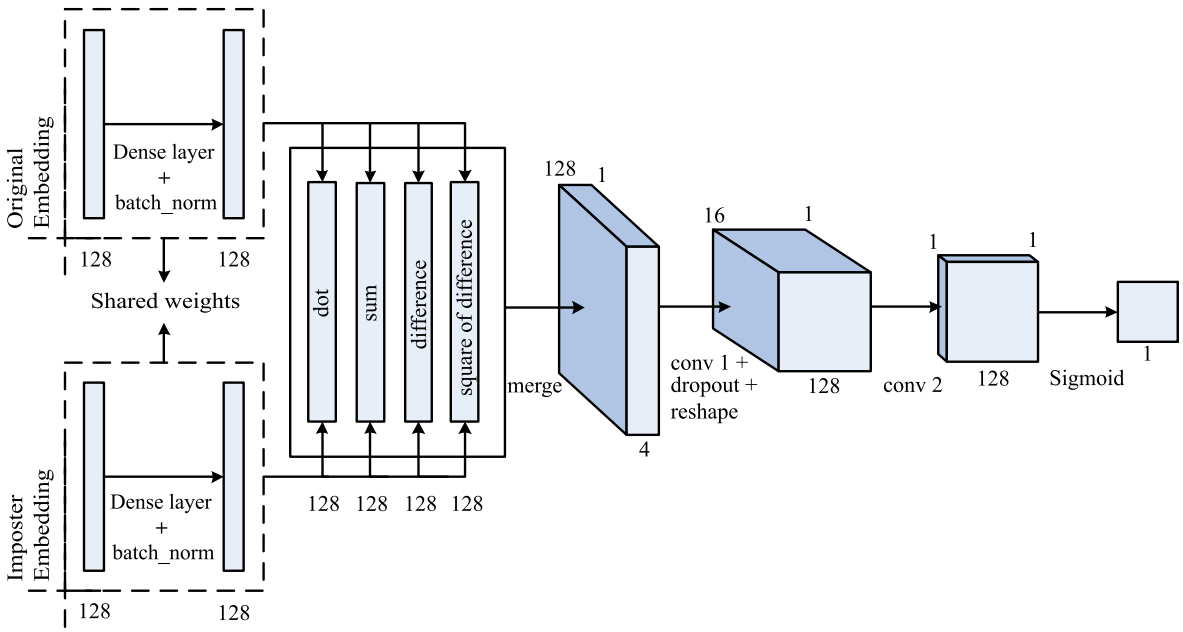

Fig. 2. Architecture for SpeakerNet.

Table 4. Configuration of SpeakerNet.

Layer	SpeakerNet	Output	Parameters
Dense Layer	128		activation = 'relu',
Dropout	Rate = 0.3		
Batch_norm	–		
Dot Sum Difference Sq. of diff	128 128 128 128	128 128 128 128	
Merge	128,128,128,128	$4 \times 128 \times 1$	
Conv1 Dropout reshape	$16 \times 4 \times 1$	 $16 \times 1 \times 128$	activation = 'linear', padding = 'valid'
Conv2	$1 \times 1 \times 16$	$1 \times 1 \times 128$	activation = 'relu', padding = 'valid'
Sigmoid	1	1	

conv – convolutional layer, batch_norm – batch normalization layer.

volution is applied on reshaped input with 1 kernel of size 1×16. This gives a vector of size 128. L1 kernel regularizer and ReLU activation is used in all of above layers. Lastly a fully connected layer with sigmoid activation is applied and result 0 is deduced as same person while 1 as fake. SpeakerNet architecture is presented in Table 4. Weights of all the layers in SpeakerNet are initialized with Glorot Normal Scheme (Glorot, Bengio, 2010). Optimization of SpeakerNet is performed using Adam with parameters presented in Table 3. Contrastive loss (Chopra *et al.*, 2005) is used in model training.

4. Experiments

In order to evaluate our speaker verification algorithm, we have used a benchmark dataset Voxceleb2 (Chung *et al.*, 2018) and a self-collected dataset comprising of multiple languages including Urdu, Arabic and English.

4.1. Datasets

4.1.1. Voxceleb2

Voxceleb2 contains over 1 million utterances for 6,112 celebrities on YouTube. The dataset is adequately gender balanced with 61% male population. The speakers stretch on a wide range of different ethnicities, accents, professions and ages. Audios present in the dataset are degraded with background chatter, laughter and varying room acoustics. Approximate length of utterances is 7.8 and there are average 185 utterances per person. All speakers are talking in English in their native accents. Voxceleb audio files are in compressed m4a codec and their bit rate is 71 kbps with 16 kHz sampling rate.

4.1.2. Self-collected Dataset

We collected our own dataset for this research from Pakistani speakers belonging to different ethnicities to cover different Pakistani accents. The Urdu and English accent varies across difference regions of Pakistan. Dataset is recorded in Urdu, Arabic and English languages. The speakers span over a wide range of accents, professions and age.

Female population consists of 60% of data making it fairly gender balanced. Data is recorded in several real time environments without setting any laboratory equipment. These environments range from quiet library to noisy outdoor parks. Data is recorded through multiple mobile devices and wav recorders. Later on audios were converted to wav codec for consistency. Other significant features of data are represented in the Table 5.

Table 5. Details of self-collected dataset.

Sampling rate	16 kHz
Bit depth	32 bit
Codec	Wav
Age	11–72 years
Female population	60%
Male population	40%
Average length per person	600 s

Data is divided into training and test sets. Speakers in training set have average recorded audio of 15 minutes. However, test set speakers have approximate length of 5 minutes.

4.2. Experimental protocol

Both datasets are originally in different formats, therefore they are converted to a standard format as described in Subsec. 3.1. Since our method is designed for text-independent speaker verification, we divided the above two datasets for training and evaluation of the proposed model. We randomly selected 20 speakers from voxceleb2 and 20 from our training set of our self-collected dataset. Both datasets are converted to similar standard format according to preprocessing steps described in Subsec. 3.1. We generated equal number of pairs for positive and negative samples for each batch. For the evaluation phase, 5 speakers from voxceleb2 and 5 from our own datasets are chosen. Both training and evaluation data is gender balanced. Total duration of samples in both phases is described in Table 6.

Table 6. Training and evaluation data.

Dataset	No. of speakers (male/female)	Speech recoded in seconds
Training	40 (20/20)	30,000
Evaluation	10 (5/5)	7,500
Total	50 (25/25)	37,000

Each audio is divided into 10 s length and Mel-spectrogram is calculated for each second then passed to CNN. The extracted embeddings from CNN are then averaged. The process is repeated on each audio again after adding gaussian noise to make it more robust. Contrastive loss margin is set to 1 in training.

4.3. Evaluation metrices

The performance of experiments is evaluated and compared using accuracy, Receiver Operating Characteristic (ROC) Area Under the Curve (AUC), and Equal Error Rate (EER). Accuracy, as shown in Eq. (5), is the ratio of true predicted values to the total number of input samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{FN} + \text{TP} + \text{FP}}. \tag{5}$$

ROC curve is a graph that shows overall performance of the model on all thresholds. The graph shows two measures, True Positive Rate (TPR) and True Negative Rate (TNR). While higher AUC depicts the superior performance of the model for distinguishing original and imposter speakers. EER is an algorithmic approach that measures error margin of a biometric system by utilizing TPR and TNR.

4.4. Baselines

Following models have been implemented as baseline.

1) L1 Siamese: We have implemented a simple L1 distance-based Siamese on top of our averaged embeddings as baseline model. L1 distance Siamese takes genuine and imposter pairs and calculate their L1 distances. A threshold is applied on the distances to accept or reject them. Contrastive loss is used to minimize loss of this network.
2) Cosine Siamese: It uses same configurations as L1 Siamese. The only difference is the use of cosine distance instead of L1 distance.
3) Binary vectors: Third baseline model calculates *b*-vector from the original and imposter embeddings by concatenating their sum, difference, dot products and square of difference. This long *b*-vector is passed through sigmoid layer to get output scores.

4.5. Results

The comparison of accuracies, AUC and EER of our proposed SpeakerNet with the baseline models are given in Table 7. Our system improves the accuracy by 8% and area under the curve by 9%.

The accuracy is lowest when we use L1 distance or cosine distance. But Table 7 also shows that area under the curve and EER for cosine distance is better than L1 distance. The reason for that is cosine distance maps the values closer to original labels i.e., 0 or 1. This behavior can be witnessed in the ROC curve in Fig. 3 as well. We analyzed the scores from L1 distance and cosine distance. It is evident that L1 distance does not draw the similar pairs together and different pairs far from each other. Instead they are closed enough hence it fails to find a proper threshold that hold for all values. While cosine distance marks the pairs closer to their labels and an appropriate threshold can improve results.

Table 7. Comparison of our system with baseline technique.

Model	Method	Acc [%]	AUC [%]	EER
Baseline	Embedding CNN + L1 distance Siamese	85	88	0.158
Baseline	Embedding CNN + cosine distance Siamese	85	97	0.073
Proposed	Embedding CNN + binary vectors	92.9	98	0.061
Proposed	Embedding CNN + SpeakerNet	93.08	98.5	0.026

Acc – accuracy, AUC – area under the curve, EER – equal error rate.

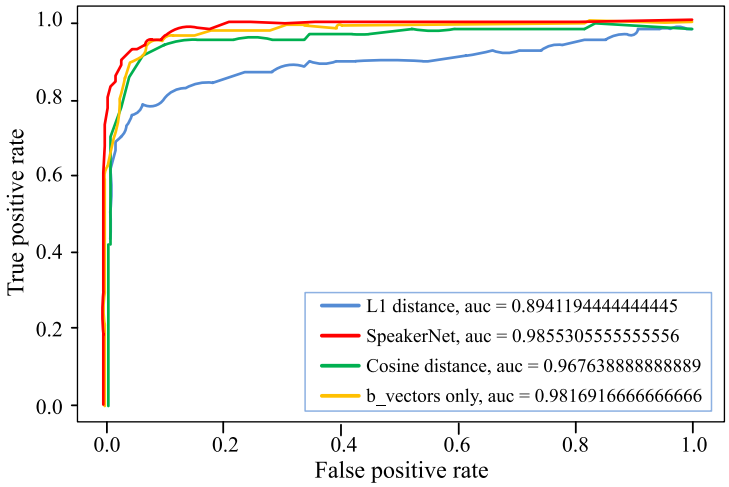

Fig. 3. ROC curve of baseline models and SpeakerNet.

As cosine distance can be interpreted as dot product of two vectors it is also called binary operation. We added more binary operations with cosine distance. Our aim was to detect a binary operation A($a!$, $a!$) which can not only be applied to any two vectors of dimensions $a! = [a!!, .., a!!]$ and $a! = [a!^{\wedge}, .., a!!]$, but is also served to do mapping between a non-empty set E and a function F, where F has output for all pair of elements in E individually and uniquely links with every pair of elements in Eq. (6)

$$E, E : F \times F \to F. \tag{6}$$

The feature representation based on binary operations does not need results from weak learners or any sub systems, whereas other methods including ensemble-based models use combination of similarity measures or discriminant scores from sub-systems. Binary operations-based feature representation acts as a package of information which a complex classifier can take and use what is helpful from that pack of information. In SpeakerNet, three basic binary operations are used to form vectors of higher dimensions. For example, the resultant vectors from addition and subtraction are concatenated. It is proved that the function that links vectors $a!$ and $a!$ to their binary vectors (in short b-vector) is injective (one to one) and surjective (onto). It is also worth mentioning that all of the above binary operations are commutative i.e. they are independent of the order. A binary vector is generated by taking dot, sum, difference and square of difference is calculated instead of a single distance calculation. This approach improved results significantly. It has improved accuracy from 85% to 92.9%, area under the curve by 1% and significantly improved EER by 1.2%.

We further analyzed the effect of stacking them on each other forming a multi-dimensional shape. Convolution is applied on the stack of these binary vectors. It gave better results than using just concatenated b-vectors. One reason for that is stacked b-vectors contain some patterns in them and convolutional layers tend to be the best to find image like patterns. SpeakerNet showed a significant decrease in EER on 0.4 threshold.

4.6. SpeakerNet vs other methods

Most of the state-of-the-art methods use CNN for embedding extraction then apply scoring or similarity measures on them. CNN embeddings provide better results than handcrafted feature extraction approaches. The results outshine because of improper selection of similarity metrics. Cosine similarity is widely used in verification tasks and provide better results. But in text independent speaker verification scenario a single threshold does not work quite well. Keeping the individual speaker specific thresholds to solve this issue affects the automation of the process and demands more memory consumption. We have compared our proposed model with other state of the art methods. A comparison in terms of Accuracy and AUC is given in Table 8. It is evident that our model performed better than all of the methods in terms of accuracy and AUC. Research has unveiled that the use of embedding networks in Siamese settings improves verification as compared to use of a single threshold in previous practice (Soleymani *et al.*, 2018). Embedding networks in Siamese work well for one-shot learning problems (Vinyals *et al.*, 2016) but treating text-independent speaker verification problem as one-shot learning does not yield promising results due to lack of generic speaker model.

Therefore, we have proposed a customized scoring scheme that utilizes capability of Siamese to apply distance measures with convolutional learning. Another difference with above mentioned state of the art methods is that all the other systems were trained only on English speakers but our proposed model is multi-

Table 8. Comparison of our method with sate of the art methods.

Paper	Method	Scoring measure	# Utt train/test	Input length [s]	Acc [%]	AUC [%]	EER
(Shi *et al.*, 2018)	ResNet with triplet loss	triplet	140,664/4,175	20	91.4	–	0.022
(Shi *et al.*, 2018)	ResNet with LGM loss (alpha = 1)	Mahalanobis distance	140,664/4,175	20	90.26	–	0.024
(Torfi, Shirvani, 2018)	3D CNN	cosine	–	–	–	87	0.220
(Soleymani *et al.*, 2018)	Prosodic-Enhanced Siamese CNN	Euclidean distance	2148/300	–	90	–	0.160
(Li *et al.*, 2017)	ResCNN, softmax (pre-train) + triplet	cosine	2,236,37/3,800	3.6–4.5	91	–	0.031
(Li *et al.*, 2017)	GRU, softmax (pre-train) + triplet	cosine	2,236,37/3,800	3.6–4.5	94.88	–	0.024
(Wang *et al.*, 2017)	*d*-vector + LDA	LDA	10,000/7,000	30	–	–	0.030
Proposed Model	CNN + SpeakerNet	proposed	30,000/7,000	10	93.08	98.5	0.026

Utt – number of utterances, Acc – accuracy, AUC – area under the curve, EER – equal error rate.

lingual. Our system is trained on Urdu, Arabic, and English utterances. We have compared our proposed model with the one in (Zhang *et al.*, 2018). It uses InceptionResNet v1 for speaker embeddings and applies PLDA and negative Euclidean distance scoring measures. It is evident from Table 9 that our model has a reduced EER.

Table 9. Comparison with transfer learning models.

Model	Scoring measure	# Utt	EER
Inception ResNetv1	Euclidean distance + PLDA	36k	0.085
Proposed	Binary vectors + convolution	37k	0.026

Utt – number of utterances, EER – equal error rate.

5. Conclusion

In this paper, we have proposed a model based on special Siamese like network for text-independent speaker verification which uses speech independent feature learning. This method does not rely on traditional feature engineering unlike its predecessors, instead it learns the features form raw audio signals. Experiments made on cross language audios of multilingual speakers that emphasize how well our proposed model detects the speakers form different accents of multiple speakers and forgers with diverse background and scripts. Moreover, the proposed SpeakerNet has achieved improved results, which is encouraging for further research in this direction. SpeakerNet can be used in combination with other embedding extraction models already trained on large speaker data for classification. It is observed that generalization of embedding CNN helps in better performance. The cross-language performance of SpeakerNet can also be increased by fine tuning on other language and accent-based datasets. The future implications can focus on the development of an enriched network model trained on larger multi lingual dataset. Additionally, different frameworks for verification task can also be explored in future. Another interesting application of SpeakerNet can be to use it with multimodal biometric verification including face, signature and iris prints.

References

1. Bell P. *et al.* (2015), The MGB challenge: evaluating multi-genre broadcast media recognition, *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 687–693, doi: 10.1109/ASRU.2015.7404863.
2. Bromley J., Guyon I., LeCun Y., Säckinger E., Shah R. (1994), Signature verification using a "Siamese" time delay neural network, *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS)*, pp. 737–744, Colorado.
3. Campbell W.M., Sturim D.E., Reynolds D.A. (2006), Support vector machines using GMM supervectors for speaker verification, *IEEE Signal Processing Letters*, **13**(5): 308–311, doi: 10.1109/LSP.2006.870086.
4. Cao X., Wipf D., Wen F., Duan G., Sun J. (2013), A practical transfer learning algorithm for face verification, *Proceedings of IEEE International Conference on Computer Vision*, pp. 3208–3215, doi: 10.1109/ICCV.2013.398.

5. Chopra S., Hadsell R., LeCun Y. (2005), Learning a similarity metric discriminatively, with application to face verification, *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 539–546, doi: 10.1109/CVPR.2005.202.
6. Chung J.S., Nagrani A., Zisserman A.J. (2018), *Voxceleb2: Deep speaker recognition*, arXiv preprint arXiv:1806.05622.
7. Cieri C., Miller D., Walker K. (2004), The Fisher Corpus: a resource for the next generations of speech-to-text, *Proccedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, Lisbon, pp. 69–71.
8. Czyżewski A., Bratoszewski P., Hoffmann P., Lech M., Szczodrak M. (2017), The project IDENT: Multimodal biometric system for bank client identity verification, [in:] *Multimedia Communications, Services and Security*, Dziech A., Czyżewski A. [Eds], Communications in Computer and Information Science, Vol. 785, pp. 16–32, Springer, Cham.
9. Czyżewski A., Hoffmann P., Szczuko P., Kurowski A., Lech M., Szczodrak M. (2019), Analysis of results of large-scale multimodal biometric identity verification experiment, *IET Biometrics*, **8**(1): 92–100, doi: 10.1049/iet-bmt.2018.5030.
10. Dehak N., Kenny P.J., Dehak R., Dumouchel P., Ouellet P. (2011), Front-end factor analysis for speaker verification, *IEEE Transactions on Audio, Speech, and Language Processing*, **19**(4): 788–798, doi: 10.1109/TASL.2010.2064307.
11. García-Salinas J.S., Villaseñor-Pineda L., Reyes-García C.A., Torres-García A.A. (2019), Transfer learning in imagined speech EEG-based BCIs, *Biomedical Signal Processing and Control*, **50**: 151–157, doi: 10.1016/j.bspc.2019.01.006.
12. Glorot X., Bengio Y. (2010), Understanding the difficulty of training deep feedforward neural networks, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, pp. 249–256.
13. Hermessi H., Mourali O., Zagrouba E. (2019), Deep feature learning for soft tissue sarcoma classification in MR images via transfer learning, *Expert Systems with Applications*, **120**: 116–127, doi: 10.1016/j.eswa.2018.11.025.
14. Hershey S. *et al.* (2017), CNN architectures for large-scale audio classification, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, pp. 131–135, doi: 10.1109/ICASSP.2017.7952132.
15. Hinton G. *et al.* (2012), Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *IEEE Signal Processing Magazine*, **29**(6): 82–97, doi: 10.1109/MSP.2012.2205597.
16. Hong Q., Li L., Zhang J., Wan L., Guo H. (2017), Transfer learning for PLDA-based speaker verification, *Speech Communication*, **92**: 90–99, doi: 10.1016/j.specom.2017.05.004.
17. Huang Z., Siniscalchi S.M., Lee C.-H. (2016), A unified approach to transfer learning of deep neural networks with applications to speaker adaptation in automatic speech recognition, *Neurocomputing*, **218**: 448–459, doi: 10.1016/j.neucom.2016.09.018.
18. Ji S., Xu W., Yang M., Yu K. (2013), 3D convolutional neural networks for human action recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(1): 221–231, doi: 10.1109/TPAMI.2012.59.
19. Koch G., Zemel R., Salakhutdinov R. (2015), Siamese neural networks for one-shot image recognition, *Proceedings of International Conference on Machine Learning (ICML) Deep Learning Workshop*, Lille, Vol. 2, pp. 1–8.
20. Korvel G., Treigys P., Tamulevicus G., Bernataviciene J., Kostek B. (2018), Analysis of 2d feature spaces for deep learning-based speech recognition, *Journal of the Audio Engineering Society*, **66**(12): 1072–1081, doi: 10.17743/jaes.2018.
21. Krizhevsky A., Sutskever I., Hinton G.E. (2017), Imagenet classification with deep convolutional neural networks, *Communications of the ACM*, **60**(6): 84–90, doi: 10.1145/3065386.
22. Larcher A., Lee K.A., Ma B., Li H. (2014), Text-dependent speaker verification: Classifiers, databases and RSR2015, *Speech Communication*, **60**: 56–77, doi: 10.1016/j.specom.2014.03.001.
23. Lei Y., Scheffer N., Ferrer L., McLaren M. (2014a), A novel scheme for speaker recognition using a phonetically-aware deep neural network, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, pp. 1695–1699, doi: 10.1109/ICASSP.2014.6853887.
24. Lei Z., Luo J., Yang Y. (2014b), A simple way to extract I-vector from normalized statastics, [in:] *Biometric Recognition, Lecture Notes in Computer Science, CCBR*, Sun Z., Shan S., Sang H., Zhou J., Wang Y., Yuan W. [Eds], Vol. 8833, pp. 366–374, Springer International Publishing, Cham, doi: 10.1007/978-3-319-12484-1_41.
25. Li C. *et al.* (2017), *Deep speaker: an end-to-end neural speaker embedding system*, arXiv preprint arXiv: 1705.02304.
26. Martin A.F., Greenberg C.S. (2010), The NIST 2010 speaker recognition evaluation, *Proceedings of Eleventh Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Chiba, pp. 2726–2729.
27. McCool C. *et al.* (2012), Bi-modal person recognition on a mobile phone: Using mobile phone data, *Proceedings of IEEE International Conference on Multimedia and Expo Workshops*, Melbourne, pp. 635–640, doi: 10.1109/ICMEW.2012.116.
28. McLaren M., Ferrer L., Castan D., Lawson A. (2016), The Speakers in the Wild (SITW) speaker

recognition database, *Proceedings of Seventeenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, San Francisco, pp. 818–822.

29. Mobiny A., Najarian M. (2018), *Text-independent speaker verification using long short-term memory networks*, arXiv preprint arXiv:1805.00604.
30. Morrison G.S., Rose P., Zhang C. (2012), Protocol for the collection of databases of recordings for forensic-voice-comparison research and practice, *Australian Journal of Forensic Sciences*, **44**(2): 155–167, doi: 10.1080/00450618.2011.630412.
31. Nagrani A., Chung J.S., Zisserman A. (2017), *Voxceleb: a large-scale speaker identification dataset*, arXiv preprint arXiv:1706.08612.
32. Qi Y., Song Y.-Z., Zhang H., Liu J. (2016), Sketch-based image retrieval via siamese convolutional neural network, *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Phoenix, pp. 2460–2464, doi: 10.1109/ICIP.2016.7532801.
33. Ramirez J., Górriz J.M., Segura J.C. (2007), Voice activity detection. Fundamentals and speech recognition system robustness, [in:] *Robust Speech Recognition and Understanding*, Grimm M., Kroschel K. [Eds], pp. 1–22, IntechOpen, Vienna, doi: 10.5772/4740.
34. Reynolds D.A., Quatieri T.F., Dunn R.B. (2000), Speaker verification using adapted Gaussian mixture models, *Digital Signal Processing*, **10**(1–3): 19–41, doi: 10.1006/dspr.1999.0361.
35. Schroff F., Kalenichenko D., Philbin J. (2015), Facenet: A unified embedding for face recognition and clustering, *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.
36. Shen W., Zhou M., Yang F., Yang C., Tian J. (2015), Multi-scale convolutional neural networks for lung nodule classification, [in:] *Information Processing in Medical Imaging (IPMI)*, Ourselin S., Alexander D., Westin CF., Cardoso M. [Eds], pp. 588–599, Springer, Cham, doi: 10.1007/978-3-319-19992-4_46.
37. Shi X., Du X., Zhu M. (2018), End-to-end residual CNN with L-GM loss speaker verification system, *Proceedings of 23rd IEEE International Conference on Digital Signal Processing (DSP)*, Shanghai, pp. 1–5, doi: 10.1109/ICDSP.2018.8631697.
38. Simonyan K., Zisserman A. (2014), *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556.
39. Soleymani S., Dabouei A., Iranmanesh S.M., Kazemi H., Dawson J., Nasrabadi N.M. (2019), Prosodic-enhanced Siamese convolutional neural networks for cross-device text-independent speaker verification, *Proceedings of 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Los Angeles, pp. 1–7, doi: 10.1109/BTAS.2018.8698585.
40. Szczuko P., Czyżewski A., Hoffmann P., Bratoszewski P., Lech M. (2019), Validating data acquired with experimental multimodal biometric system installed in bank branches, *Journal of Intelligent Information Systems*, **52**(1): 1–31, doi: 10.1007/s10844-017-0491-2.
41. Torfi A., Shirvani R.A. (2018), *Attention-based guided structured sparsity of deep neural networks*, arXiv preprint arXiv:1802.09902.
42. Tran D., Bourdev L., Fergus R., Torresani L., Paluri M. (2015), Learning spatiotemporal features with 3d convolutional networks, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, pp. 4489–4497, doi: 10.1109/ICCV.2015.510.
43. Vinyals O., Blundell C., Lillicrap T., Wierstra D. (2016), Matching networks for one shot learning, *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Barcelona, pp. 3630–3638.
44. Wang D., Li L., Tang Z., Zheng T.F. (2017), Deep speaker verification: Do we need end to end?, *Proceedings of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Kuala Lumpur, pp. 177–181, doi: 10.1109/APSIPA.2017.8282024.
45. Woo R.H., Park A., Hazen T.J. (2006), The MIT mobile device speaker verification corpus: data collection and preliminary experiments, *Proceedings of IEEE Odyssey – The Speaker and Language Recognition Workshop*, San Juan, pp. 1–6, doi: 10.1109/ODYSSEY.2006.248083.
46. Zeinali H., Sameti H., Stafylakis T. (2018), DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English, *Proccedings of Odyssey 2018 The Speaker and Language Recognition Workshop*, Les Sables d'Olonne, pp. 386–392, doi: 10.21437/Odyssey.2018-54.
47. Zhang C., Ranjan, S., Hansen J. (2018), An analysis of transfer learning for domain mismatched text-independent speaker verification, *Proceedings of Odyssey 2018 The Speaker and Language Recognition Workshop*, Les Sables d'Olonne, pp.181–186, doi: 10.21437/Odyssey.2018-26.
48. Zhang L., Yang J., Zhang D. (2017), Domain class consistency based transfer learning for image classification across domains, *Information Sciences*, **418**: 242–257, doi: 10.1016/j.ins.2017.08.034.