Learning noise robust ResNet-based speaker embedding for speaker recognition
Mohammad Mohammadamini, Driss Matrouf, Jean-François Bonastre, Sandipana Dowerah, Romain Serizel, Denis Jouvet

To cite this version:
Mohammad Mohammadamini, Driss Matrouf, Jean-François Bonastre, Sandipana Dowerah, Romain Serizel, et al.. Learning noise robust ResNet-based speaker embedding for speaker recognition. Odyssey 2022: The Speaker and Language Recognition Workshop, Jun 2022, Beijing, China. hal-03650549

HAL Id: hal-03650549
https://hal.science/hal-03650549
Submitted on 25 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Learning noise robust ResNet-based speaker embedding for speaker recognition

Mohammad MohammadAmini\textsuperscript{1,}, Driss Matrouf\textsuperscript{2,}, Jean-François Bonastre\textsuperscript{1}
Sandipana Dowerah\textsuperscript{2,}, Romain Serizel\textsuperscript{2,}, Denis Jouvet\textsuperscript{2}

\textsuperscript{1}LIA (Laboratoire Informatique d’Avignon)
University of Avignon, France
\textsuperscript{2}University of Lorraine
CNRS, Inria, Loria, F-54000, Nancy, France
mohammad.mohammadami, driss.matrouf, jean-francois.bonastre@univ-avignon.fr
sandipana.dowerah, romain.serize@loria.fr, denis.jouvet@inria.fr

1. Introduction

A speaker recognition (SR) system authenticates the identity of a claimed user from a speech utterance. The state-of-the-art systems mainly rely on Deep Neural Networks (DNN) for speaker modeling. From their emergence until now, numerous DNN architectures have been introduced in the speaker recognition realm in which TDNN\textsuperscript{1} and ResNet\textsuperscript{2} systems are among the well-known and efficient systems. Although DNN-based SR systems have shown a degree of robustness in the presence of background noise and reverberation, their performance reduces in severe conditions\textsuperscript{3}.

The problem of noise and reverberation is addressed at different levels of speaker recognition systems, including signal level\textsuperscript{4}, feature level\textsuperscript{5}, speaker modeling level\textsuperscript{6}, x-vector level\textsuperscript{7} and scoring technique adaptation\textsuperscript{8}. Data augmentation is another approach to making the speaker recognition systems robust against noise. Researches show that data augmentation brings a degree of robustness to the speaker recognition system\textsuperscript{1, 3}, but still, their performance degrades in noisy environments because there is no constraint on the speaker embedding system to extract identical or close x-vectors for pairs of noisy-clean version of the same signal.

Noise compensation in x-vector level, the estimation of clean x-vector from its corresponding noisy version, by doing a transformation between pairs of noisy/clean x-vectors is another approach that performed well in the compensation of artificial noise and reverberation\textsuperscript{9, 3}. Although this approach performs well in some cases\textsuperscript{7}, it doesn’t bring a significant improvement with all speaker embedding systems\textsuperscript{10} and in all environments. The behavior of different speaker embedding systems is different because they just consider the speaker classification accuracy (inter-speaker and intra-speaker distance) during optimization and they don’t put an explicit constraint on the noise impact. This characteristic makes noise compensation more difficult in some speaker embedding systems. To overcome this challenge, in the current paper, we propose two training strategies of ResNet-based speaker recognition systems that impose on the speaker embedding to extract x-vector for noisy signals that are close to their corresponding version in the clean environment.

In the first approach, the objective is to optimize the speaker embedding in a manner that converges toward the same point for the pairs of noisy/clean samples. In this system, two loss functions are used. The cross-entropy loss function is minimized for speaker classification and at the embedding layer, the mean square error (MSE) between pairs of noisy/clean x-vectors is optimized. Although the system improves for noise and reverberation, its performance is lower than the baseline system for clean environments.

To solve this problem, we propose a second system. In the second system firstly an optimal x-vector extractor for the clean environment is trained. After that, another speaker embedding is trained that jointly reduces the cross-entropy of the speaker classifier and mean square error (MSE) between the output of the embedding layer and an optimal clean x-vector extracted with the pretrained system. Since it is imposed on the output of the embedding layer to converge toward an optimum clean space, the performance of the speaker embedding is preserved for clean environments. Our proposed approaches, in all
cases with different types of simulated and real noises, outperforms the baseline ResNet system. For the sake of readability in the next parts of the paper, we call the first proposed system ResNet-MSE1 and the second system named ResNet-MSE2.

The rest of the paper is organized as: the related works are reviewed in section 2, in section 3 the architecture of the proposed systems is described, the experiment’s setup is described in section 4, the results are discussed in section 5.

2. Related works

Noise and reverberation are treated in the different parts of speaker recognition systems. Here we just review works in the speaker modeling level (x-vector extractor) and speaker embedding level (x-vector), which are directly related to our work.

In several works, the researchers tried to make the speaker embeddings robust to noise and reverberation. In [11] a domain adaptation technique is proposed that uses mean discrepancy distance (MMD) as a regularizer with speaker embedding that performs the adaptation between source and target domain. In this paper the proposed method is tested on language adaptation and its efficiency for noise and reverberation adaptation is not examined. In [12] an adversarial strategy was proposed to make the speaker embedding more robust against noise. In the standard x-vector extractors, after the embedding layer, a DNN speaker classifier is optimized. In this work, a second classifier is trained adversarially that accepts the type of noise in the output. In another work, a GAN-based speaker embedding was proposed that uses a binary discriminator to discriminate the noisiness of the x-vector alongside the speaker recognition classifier [13]. The main deficiency of adversarial speaker embedding systems is the labels that should be used in the discriminator. Another important point is that the training of the network in a manner that can not be able to discriminate the type of noise or the noisiness of an x-vector doesn’t guarantee that noisy x-vectors are close enough to their clean version.

Some works are done at the x-vector level to reduce the impact of noise and reverberation in SR systems. In [9] several denoising autoencoders (DAE) are proposed to remove the impact of additive noise from x-vectors. In [3] the impact of data augmentation alongside noise compensation is explored. Despite having good results for simulated noises this work doesn’t include real noise and reverberation. In another work two configurations are proposed to denoise different kinds of distortions such as noise, early reverberation, and late reverberation [7]. In this paper also the capability of doing noise compensation is not explored in real environments. Also, in all of them, denoising is done with TDNN-based speaker embedding. In another work, it is shown that noise compensation doesn’t bring a significant gain with x-vectors extracted from ResNet SR systems [2].

In the current paper, we introduce two training strategies that impose on the speaker embedding network to extract x-vectors for a noisy/reverberated signal that is close to the corresponding clean version.

3. Proposed system

In this section, the architecture of the baseline ResNet system and the proposed variants are described.

3.1. Baseline system

The baseline embedding extractor used in this paper is a variant based on ResNet [14]. The ResNet model for extracting embeddings consists of three modules: a set of ResNet Blocks, a statistics-level layer, and segment-level representation layers.

- ResNet (Residual Network) uses stacks of many Residual Blocks. A Residual Block is made up of two 2-dimensional convolutional Neural Networks (CNN) layers separated by a non-linearity (ReLU). The input of the Residual Block is added to its output in order to constitute the input of the next Residual Block.
- The statistics-level component is an essential component to convert a variable-length speech signal into a single fixed-dimensional vector. The statistics level is composed of one layer: the statistics-pooling, which aggregates over frame-level output vectors of the DNN and computes their mean and standard deviation.
- The segment-level component maps the segment-level vector to speaker identities. The mean and standard deviation are concatenated together and forward to additional hidden layers and finally to the softmax output layer.

The detailed topology of the used ResNet is shown in Table 1. Batch-norm and ReLU layers are not shown. The dimensions are (Frequency×Channels×Time). The input is comprised of 60 filter banks from speech segments. During training, we use a fixed segment length of 400. The speaker ResNet system is trained with Cross Entropy loss function (Eq. 1):

\[ L_{CrossEntropy} = -\sum_{c=1}^{N} y_{o,c} \log(p_{o,c}) \]  (1)

where N is the number of speakers, o is a speech signal, \( y_{o,c} \) is the truth label of o, and \( p_{o,c} \) is the output of the softmax activation function.

3.2. ResNet-MSE1

In this subsection, the first proposed system is described (Fig. 1). In the proposed system a noisy signal and its corresponding clean version are given to the network. At the embedding layer the mean square error (MSE) between the noisy and clean x-vectors is calculated at each minibatch (Eq. 2):

\[ L_{MSE} = \frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{D} (y_{ij} - x_{ij})^2 \]  (2)
where $B$ is the size of minibatch, $D$ is the size of x-vector, $x_j$ is the $j$th dimension of clean x-vector and $y_j$ is $j$th dimension of noisy x-vector.

Both versions of the signal (i.e. clean and noisy) are given to the classifier. In this system a combination of cross-entropy and mean square error is used as the loss function (Eq. 3):

$$L_{\text{ResNet-MSE}} = L_{\text{CrossEntropy}} + L_{\text{MSE}} \quad (3)$$

During the training process, the speaker embedding converges toward an intermediate space between clean and noisy environments. Minimizing the MSE distance between noisy and clean x-vectors improves the performance in noisy environments while the performance in clean environments becomes a little worse because the clean samples move toward the noisy sample. This attribute is shown in Figure 3a. In the second proposed system, we resolve this problem.

### 3.3. ResNet-MSE2

In the second proposed system, the performance degradation in the clean environment is resolved. To do that, we used an assistance pretrained network which is the same as the baseline system. Firstly a speaker embedding network with a mixture of clean and noisy data is trained. We used both noisy and clean data to train this network because the previous research shows using more data and data augmentation improves the performance of speaker recognition systems for both clean and noisy environments [3]. The pretrained network is used to extract x-vectors for the clean version of the training dataset. We assume these vectors as the best version we can achieve. After that, another network is used to shift the noisy version of x-vectors towards the clean version extracted in the previous step. In this step, the system is trained from scratch with samples containing noisy and clean signals. At the speaker embedding layer, the MSE between the fixed clean version and a given training sample (for both clean and noisy signals) is calculated. Both clean and noisy versions are given to the classifier and the weights are updated with all samples. This procedure is depicted in Figure 2. The steps of convergence towards the clean x-vector are shown in Figure 3b.

### 4. Experiments setup

#### 4.1. Dataset

In this subsection, all the corpora used in our paper are described.

- **Voxceleb.** In our experiments we used Voxceleb 2 [15] for training x-vector extractors. There are 1.2m samples from 5,994 speakers. The clean version is augmented with Musan corpus. The final version of training data included 5.9m samples.

- **Musan.** Musan is music, speech, and noise corpus comprising 109 hours of speech data. This corpus is used for data augmentation to training x-vector extractors [16].

- **Freesound.** This corpus included 3,000 RIR files and 4,275 noise files collected from Freesound noises. This corpus is used as artificial noise for evaluation protocols [17].

- **BBC Noise.** BBC Noise includes 16000 noise files, provided by BBC, these noises are used as artificial noise for an evaluation protocol \(^1\).

- **Fabiole.** Fabiole 1 is a French corpus that contains 7,000 files from 130 speakers [18]. We created two protocols from this dataset.

- **Robovox.** It is a French corpus collected from the Robovox project (A mobile robot). Each recording in this corpus has 5 channels. The fifth channel is a close microphone which we considered as clean and the third channel is the farthest channel that we consider as noisy and reverberated\(^2\).

---

\(^1\)http://bbcsfx.acropolis.org.uk

\(^2\)https://robovox.univ-avignon.fr/
4.2. x-vector extractors

All x-vector extractors are trained with Voxceleb in 10,000 iterations. The learning rate at the beginning of the training is set to 0.2 and weight decay equals to $2 \times 10^{-4}$. The momentum is set to 0.9. In all experiments, the stochastic gradient descent optimizer is used. The size of the feature maps is 32, 64, 128, and 256 for the 4 ResNet blocks.

- **Baseline.** In the baseline system, the training samples are chosen randomly from clean Voxceleb. The training data includes all clean files of Voxceleb and their augmented version with MUSAN Corpus and reverberated with a pool of RIR files. Kaldi toolkit is used for data augmentation [20]. The batch size is set to 128.

- **ResNet-MSE1.** In this system, a clean file from Voxceleb was chosen randomly. After that its augmented version was chosen. Because we have two versions of each file at each minibatch, we reduced the size of each minibatch to 64.

- **ResNet-MSE2.** In this system at each minibatch, an x-vector extracted from the baseline system was chosen. The modified system tries to reduce the distance between the clean x-vector and the x-vector extracted from the given signal.

4.3. Test protocols

- **Fabiole1.** In the first protocol, the Fabiole corpus is used. In this protocol 130 files (one file per speaker) are used as enrollment and 1,870 randomly chosen files are used for the test. In this protocol, the BBC noise files are added to the clean signal with different SNRs. In this protocol, the Kaldi toolkit is used to add noises to clean files.

- **Fabiole2.** The test and enrollment files in this protocol are the same as the previous one. But in this protocol, we used Freesound noises. We used Pyacoustics for data simulation.

- **Robovox.** In this protocol 26 files, one file per speaker, are used as the enrollment and 677 files are used as the test. The enrollment files are chosen from a close microphone with high quality but the test files are chosen from a far microphone between 1 and 3 meters.

- **VoıCes.** In this protocol 300 files, one file per speaker, are used as enrollment and 300 files are used as the test. The enrollment files adopted from the Librispeech (the clean version of VoıCes) and the test files are the replayed files in VoıCes recorded in room 4 in the presence of severe music noise and reverberation. We used mic 5, which is the farthest microphone in VoıCes.

The details of all protocols are summarized in Table 2.

5. Results and discussion

In this section, the obtained results are discussed. Our results show that in all cases our proposed systems are more robust in noisy environments. However, the performance of the first proposed system reduces in clean environments in comparison to the baseline system, the performance of the second system for clean environments remains stable.

In Table 3 the results for Fabiole1 protocol are presented. As it is shown, the modified systems improve significantly in comparison to the baseline system. The first column shows the results for clean environments and the other columns include the results with different SNRs. For example, when SNR is between 0 and 5 the EER with the second proposed system (ResNet-MSE2) improves 15% compared to the baseline system. In the case of SNR between 10 and 15, the ResNet-MSE2 improves 20% in terms of EER.

In the Fabiole2 protocol, we showed the generalizability of the proposed systems to other noises and RIR simulators. In this protocol, the same test and enrollment files as Fabiole1 protocol are used. But the Freesound noises dataset and Pyacoustics library are used for data simulation. The test and enrollment files are preserved as the previous protocol to show the robustness of the proposed systems against the impact of other noises and reverberations. Table 4 shows that in all cases the proposed systems reduce the impact of noise and reverberation. It deserved to be mentioned that in this protocol the results for the clean situation are worse in comparison to the Fabiole1 protocol because test files are truncated to 15 seconds for both clean and noisy situations. For the SNR between 0 and 5, the EER improves 14% with the ResNet-MSE2 system.

In order to extend the capability of the proposed systems to real environments, the experiments are done on the Robovox dataset in Table 5. The first column shows the results with the best channel with a close microphone. As it is shown in Fabiole protocol the ResNet-MSE1 system is worse in the clean environment in comparison to the baseline system and the results
for the ResNet-MSE2 system are the same as the baseline system. In the third column, the results are shown for the far microphone in the presence of noise and reverberation. Both adapted systems give better results in comparison to the baseline system. The last column shows the results with simulated noise in the Robovox protocol. In this experiment, the clean data from channel 3 is augmented with Freesound noises and Pyacoustics with SNR between 0 and 5. In this experiment the behavior of the proposed systems is the same. For real noise the relative improvement with the ResNet-MSE2 system is 16% and for simulated noise the gain is 10%.

Table 5: Robovox protocol (EER).

<table>
<thead>
<tr>
<th>System</th>
<th>Ch3</th>
<th>Ch3</th>
<th>Noisy (0-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>2.21</td>
<td>4.38</td>
<td>6.59</td>
</tr>
<tr>
<td>ResNet-MSE1</td>
<td>2.36</td>
<td>4.10</td>
<td>6.05</td>
</tr>
<tr>
<td>ResNet-MSE2</td>
<td>2.21</td>
<td>3.69</td>
<td>5.90</td>
</tr>
</tbody>
</table>

Finally, the systems are tested with a protocol created from the VoicEs dataset. The results are shown in Table 6. The experiments show that with the ResNet-MSE2 system there is 5% improvement of EER.

Table 6: Voices protocol (EER).

<table>
<thead>
<tr>
<th>System</th>
<th>Clean</th>
<th>Room 4 music</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.66</td>
<td>6.33</td>
</tr>
<tr>
<td>ResNet-MSE1</td>
<td>0.66</td>
<td>6.33</td>
</tr>
<tr>
<td>ResNet-MSE2</td>
<td>0.66</td>
<td>6.00</td>
</tr>
</tbody>
</table>

The results obtained from ResNet-MSE1 in a clean environment show that reducing the distance between noisy and clean environments without fixing the clean point makes the system worse in clean environments because it converges to a space between the noisy and clean space. To solve this problem, we fixed the clean x-vector in the second proposed system. Also, the results show that the second system is superior in noisy environments because during the process of reducing the distance between noisy and clean x-vectors the noisy x-vector moves closer toward the clean point to minimize the MSE.

The MSE between noisy and clean x-vectors is shown in Table 7. The big distance between noisy and clean x-vectors of the same signal in the Baseline system shows that despite having high accuracy in speaker classification, a noisy x-vector of the same signal can be far from its clean version because the speaker classification’s loss function doesn’t impose on the system to have close or identical representations for the noisy and clean version of a specific signal. The presented results show that imposing on the speaker embedding system to extract close x-vectors for noisy and clean x-vectors makes the system more robust against noise and reverberation.

Table 7: MSE distance between pairs of noisy-clean x-vectors.

<table>
<thead>
<tr>
<th>System</th>
<th>Fabiole1</th>
<th>Fabiole2</th>
<th>Robovox</th>
<th>VoicEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>2.85</td>
<td>7.14</td>
<td>4.05</td>
<td>10.23</td>
</tr>
<tr>
<td>ResNet-MSE1</td>
<td>0.06</td>
<td>0.015</td>
<td>0.09</td>
<td>0.02</td>
</tr>
<tr>
<td>ResNet-MSE2</td>
<td>0.66</td>
<td>1.71</td>
<td>1.01</td>
<td>2.67</td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper, we proposed two strategies to train ResNet-based speaker embeddings in order to make the speaker recognition systems more robust against additive noise and reverberation. In the first system, the network is updated to reduce the distance between pairs of noisy/clean x-vectors in the embedding layer. In the second system, an optimal clean point is fixed and at each iteration, the noisy and clean x-vectors given by the signal in the input are shifted toward the optimal point. Training x-vector extractors with the proposed strategies make the speaker embeddings more robust against additive noise and reverberation. The future potential work would be exploring other loss functions to reduce the difference between noisy and clean x-vectors and/or exploring different data augmentation techniques with proposed approaches to make the system more robust in a specific situation.

7. Acknowledgements

This work was supported by the Robovox ANR-18-CE33-0014 project.

8. References


