INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

ResNet and Model Fusion for Automatic Spoofing Detection

Zhuxin Chen, Zhifeng Xie*, Weibin Zhang, Xiangmin Xu*

School of Electronic and Information Engineering
South China University of Technology, GuangZhou, China
{chen.zhuxin,xie.zhifeng}@mail.scut.edu.cn, {eeweibin,xmxu}@scut.edu.cn

Abstract

Speaker verification systems have achieved great progress in recent years. Unfortunately, they are still highly prone to different kinds of spoofing attacks such as speech synthesis, voice conversion, and fake audio recordings etc. Inspired by the success of ResNet in image recognition, we investigated the effectiveness of using ResNet for automatic spoofing detection. Experimental results on the ASVspoof2017 data set show that ResNet performs the best among all the single-model systems. Model fusion is a good way to further improve the system performance. Nevertheless, we found that if the same feature is used for different fused models, the resulting system can hardly be improved. By using different features and models, our best fused model further reduced the Equal Error Rate (EER) by 18% relatively, compared with the best single-model system.
Index Terms: Replay attacks, Residual neural network, Model fusion, ASVspoof2017

1. Introduction

Intelligent personal assistants such as Siri, Alexa and Cortana are becoming more and more popular. For these agents, it is very important to recognize and understand what the user says. Another equally important issue is to recognize the user by his/her voice in order to provide personalized services such as schedule management and personal health management. Automatic speaker verification (ASV), a convenient biometric person authentication system that recognizes the speaker's identification based on the speech recordings, has gain more attention recently. ASV systems have been widely used in many commercial and civilian applications such as credit cards and telephone banking, e-commerce, secure building access and suspect identification. Especially where there is no face to face contact, speaker verification takes a crucial place [1, 2]. Approaches like Joint Factor Analysis (JFA) [3], i-vectors [4] and deep neural networks [5, 6] had achieved great performance in this task.

However, even state-of-the-art ASV systems are still weak for some spoofing attacks such as replay attack (RA), voice conversion (VC) and speech synthesis (SS). Voice conversion, which converts the speech without losing the target speaker's distinct characters, is one of the most accessible methods of attack. With enough training data, modern speech synthesis systems can 'speak' very similar to the target speaker. In replay attacks, audio recorded from the genuine speaker will be replayed by high-fidelity playback devices to attack recognition systems. Replay attacks can be highly effective. These spoofing voices differ very little from the genuine speaker's voice and thus have high possibilities of spoofing. Spoofing attacks have significantly increased the false acceptance rate of ASV systems [7, 8, 9, 10].

*Both authors contribute to this manuscript equally.

It is therefore very important to develop systems to automatically detect spoofing attacks — either in a joint modeling approach that can detect spoofing attack while at the same time perform the speaker verification task [11], or in a separate system that is used in conjunction with a speaker verification system [12]. In the challenges organized around the topic of spoofing detection, most of the methods focus on building more appropriate feature representations such as phase spectrum [13], linear prediction error [14], magnitude spectrum [15] and constant Q cepstral coefficients (CQCCs) [16]. According to the results reported in ASVspoof challenge 2015 [17], even though most the submitted systems have good behavior among known attacks, they failed to achieve good performance in unknown attacks. This suggests that the development of countermeasures that generalize well to unseen attacks still has a long way to go.

In this year, ASVspoof 2017 concentrates on replay attack, especially those encountered under 'unseen' conditions, for instance, using different replay environments, playback devices or talkers. The challenge provides training and development data, which contains the genuine and spoofed speech [18]. This paper describes our submission to the challenge.

This paper includes a number of contributions. First of all, as pointed out in [19], conventional MFCC features with mel-frequency scale filters along the speech bandwidth may miss some discriminative information between genuine and spoofed speech. Instead of allocating different number of filters to different sub-bands [19], we found that we can simply increase the number of filters along the whole speech bandwidth, especially when deep neural networks are used. Secondly, the depth of neural networks is theoretically a major determinant of model expressiveness [20]. We found that residual neural networks can help build deeper models that outperform conventional deep neural networks. Last but not least, model fusion is a good way to further improve the system performance. Nevertheless, we found that if the same feature is used for different fused models, the resulting system can hardly be improved. By using different features and models, the system performance can be significantly improved.

The rest of this paper is organized as follows. In Section 2, we describe the features used in our systems, i.e. MFCCs and Constant Q Cepstral Coefficients (CQCCs). The models explored are introduced in Section 3. Experimental setup and results will be given in Section 4. Finally, we draw a conclusion in Section 5.

2. Features

This section describes the features used in our experiments. Mel frequency cepstral coefficients (MFCCs) have been widely used in speech recognition and speaker recognition. Constant Q Cepstral Coefficients (CQCCs) are newly proposed for automatic spoofing detection. Both of them will be used in our experi-

http://dx.doi.org/10.21437/Interspeech.2017-1085

ments.

2.1. Constant Q Cepstral Coefficients

Instead of using Fourier transform, the Constant Q Cepstral Coefficients use Constant Q transform (CQT) with traditional cepstral processing to analyze speech signal. In contrast to the fixed time-frequency resolution of Fourier transform, CQT gives a higher frequency resolution for low frequencies and a higher temporal resolution for high frequencies. CQCCs have proven to be an effective spoofing countermeasure in ASVspoof 2015 Challenge, especially for SS and VC attacks. Experiments show that CQCC features have higher ability to capture the tell-tale signs of manipulation artefacts of spoofing attacks than other approaches. More details about Constant Q Cepstral Coefficients can be found in [16].

2.2. Mel Frequency Cepstral Coefficents

Mel Frequency Cepstral Coefficients (MFCCs) have been used in many speech applications such as speech recognition and speaker verification. Recently, some researchers [19] argue that the traditional MFCCs may miss some discriminative information between genuine and spoofed speech. In our preliminary experiments, we also found that conventional MFCC features do not work very well. Researchers in [19] proposed to allocate different number of filters to different sub-bands to overcome this problem. Since DNNs are able to take care of high-dimension input features, we found that we can also simply increase the number of filters to keep all the information needed to discriminate between genuine and spoofed speech.

3. Classifiers

Three classifiers were used in our experiments, i.e. the Gaussian mixture model, deep neural networks and residual neural network. We will introduce them in this Section.

3.1. Gaussian Mixture Models

Our first classifier is Gaussian mixture models (GMMs). All the genuine speech was used to train a genuine speaker model and the spoofed data were used to train a spoofed one. During testing, log-likelihood ratio (LLR) for each speech frame o_t is calculated by using the Equation (1). The final score for an utterance is calculated by summing all LLR scores, and normalized by the number of frames.

$$LLR = logP(o_t|M_{genuine}) - logP(o_t|M_{spoof}) \quad (1)$$

3.2. Deep Neural Networks

Deep neural networks (DNNs) have achieved tremendous success in many applications including spoof detection [5, 6]. We also used conventional feed-forward DNNs as our baseline. For example, when CQCCs were used, 32-dimension CQCCs, their delta, and double delta were used. Eleven consecutive CQCC frames were spliced together, resulting in 1056-dimension feature input for the network. The deep neural network we used has three hidden layers, with 512 units for each layer. Each hidden layer is followed by a batch normalization [21] layer in order to make the learning process less sensitive to the learning rate and initial parameters. The cross-entropy was used as the loss function. The standard backpropagation algorithm and dropout regularization [22] are used to train the network and prevent it from overfitting. The output layer was a softmax of dimension 2, one for the genuine speech, and the other one for spoof.

During testing, it is similar with the case of GMM models except that we use the posterior probabilities given by the network outputs.

3.3. Residual Neural Network

It is well known that the depth of a network is a determined factor of the network performance. Deeper and deeper networks are used in computer vision area. However, it is not easy to train a deeper network due to the notorious gradient vanishing problem.

To address this problem, the authors in [20] proposed a simple but effective method called residual neural network (ResNet). ResNet provides a training framework to ease the training of networks that are substantially deeper than those used previously. It was motivated by the counterintuitive experimental findings that adding more layers lead to higher training error. Theoretically, as the number of layers increased, the modeling capabilities of the Neural Networks should be better and therefore, the deeper networks should produce no higher training error. The authors suspect that this is because the gradients vanish after they are propagated many layers. Instead of adding parameterized gating functions to allow part of the earlier information to flow unimpededly to later layers through highway connections in Highway Networks [23], the authors in [20] proposed to simply add shortcut connections with identity functions to the networks. Experiments in [20] showed that ResNet greatly improves the training efficiency since the gradients can propagate many layers through the shortcut connection. In addition, ResNet allows deeper networks to be trained, resulting in models that usually perform better. We will use ResNets as the building block of our networks in the experiments.

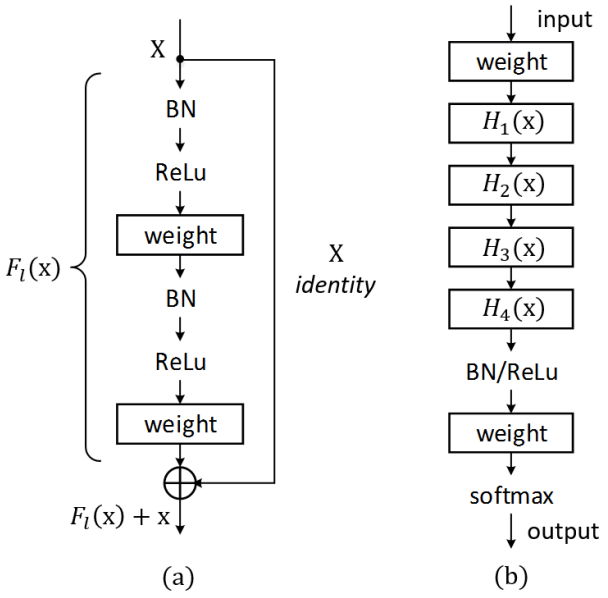

Figure 1: *(a) A building block of residual network where a conventional feed-forward network is paired with a shortcut connection. BN is a batch normalization layer. The l^{th} building block of a residual network learns the transformation $H_l(x) = F_l(x) + x$. (b) The network architecture of a full ResNet. There are four building blocks.*

Fig.(a) shows a building block of a ResNet. Formally, if the input of the l^{th} building block is x. Let H_l denotes the transformation of the l^{th} building block. That is, the desired output is $H_l(x)$. As shown in Fig.(a), the stacked nonlinear layers consists of two rectified linear layers and fully connected layers. Due to the shortcut connection as shown in Fig.(a), ResNet explicitly forces the output to fit a residual mapping, i.e., the stacked nonlinear layers are forced to learn the following transformation:

$$F_l(x) = H_l(x) - x \tag{2}$$

Therefore the transformation for the l^{th} building block is:

$$H_l(x) = F_l(x) + x \tag{3}$$

In our experiments below, we used residual networks with four building blocks, as shown in Fig.1(b). Each hidden layer in the ResNet also has 512 nodes. In addition, we add normalization layers to ease the training process. As in conventional DNNs, a context of eleven frames was used and the output is softmax.

3.4. Models Fusion

Model fusion provides a mechanism to combine the advantage of different models to further improve the system performance. It has been used in many applications [24, 25, 26]. We also incorporate it in our experiments.

To ensure that the output of different models can be linearly fused together, we firstly normalize them. Suppose x is the original output, we can calculate the mean μ and variance σ of x on the training data. Then the normalized score y is given by the following equation.

$$y = \frac{x - \mu}{\sigma} \tag{4}$$

The final score s for the fused model is a weighted sum of the normalized ys, i.e.

$$s = \Sigma_{i=1}^{N} w_i \times y_i \tag{5}$$

$$\Sigma_{i=1}^{N} w_i = 1$$

If two models are fused, then the weights will be α and $1 - \alpha$. All the weights will be tuned on the development data.

4. Experiments

4.1. Datasets

The ASVspoof 2017 focuses on spoofing attack detection with 'out in the wild' condition. The data released were mainly based on the RedDots corpus and its replayed version, which was a text-dependent database. The RedDots corpus served as the genuine speech, and its replay served as the spoofed data. The spoofed data was recorded through a variety of different environments in the ongoing H2020-dunded OCTAVE project2. Speech from a subset of the original *RedDots* corpus was replayed through different replay configurations consisting of varied devices, recording devices and loudspeakers.

The released datasets include the training and development data. Table 1 briefly gives the statics of different datasets. In our experiments reported below, we used all the training data to train the models and all the development data to tune the model parameters. Besides the primary labels (genuine/spoof), each audio file in the training and development data sets was also provided with information of the text context, speaker, recording environment, playback device and recording device. We only used the primary label to train and tune our models.

As for evaluation, 13,306 utterances were used. A larger number of speakers was expected in the evaluation set, with a much larger replay-to-genuine ratio comparing to the development set.

Table 1: *Number of speakers, genuine and spoofed utterances in the training and development sets.*

Subset	#speakers	#genuine-utterances	#spoofed-utterances
Training	10	1508	1508
Development	8	760	950

4.1.1. Training data

There were 3016 utterances in total from ten speakers in the training data. The spoofed speech was generated with three replay configurations in six different sessions. The whole training data set was used to train our model.

4.1.2. Development data

The development data set contained genuine and spoof speech from a total of 8 speakers. Ten different replay sessions with different playback and recording devices were used to produce the replay spoof samples for the development data. The devices in this part were mostly different from those used in the training data.

4.1.3. Evaluation data

The evaluation data set contains genuine and spoof speech. Only some of the replay conditions were the same as those in development and training data. Most of them were intentionally recorded and replayed in different unseen conditions to encourage research towards generalized spoofing countermeasure.

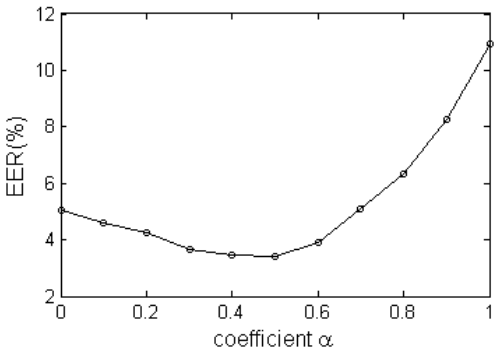

Figure 2: *The Equal Error Rates (EERs) and the corresponding fusing weights α when a ResNet with CQCC input was fused with another ResNet with MFCC input.*

Table 2: *EER(%) of different systems in development and evaluation data sets*

System description	EER (Dev)	EER (Eval)
CQCC GMM	10.83	28.46
CQCC DNN	5.18	19.41
CQCC ResNet	**5.05**	18.79
MFCC ResNet	10.95	**16.26**
CQCC DNN + CQCC ResNet	5.05	18.79
MFCC ResNet + CQCC ResNet	3.45	14.88
CQCC GMM + MFCC ResNet + CQCC ResNet	**2.58**	**13.30**
CQCC GMM + MFCC ResNet + CQCC DNN	2.76	13.44

4.2. Experimental setup

As stated above, MFCCs were used in our experiments. The speech signal was analyzed using an overlapping 25-ms Hamming window every 10-ms. Our preliminary experimental results showed that increasing the number of triangular mel-frequency filters can significantly improve the system performance. In our experiments, the number of filters was increased to 60 and we used 30 cepstral coefficients. Cepstral mean normalization was applied using a 300-frame sliding window. Finally, we augmented the features with their delta and double delta coefficients. As for the CQCC extraction, we followed the typical settings and the Matlab tool introduced in [16].

All the Gaussian mixture models were trained with 512 Gaussian components. Static features, together with their dynamic parts, were used as the input for GMM models. For the input of the DNNs, we used a context window of 11 feature frames (5 previous frames, current frame and 5 future frames). Categorical cross-entropy was used as the loss function.

For model fusion, the fusion parameter a was tuned by using the development data. Figure 2 shows the tuning process when a ResNet with CQCC input was fused with another ResNet with MFCC input.

4.3. Experimental Results

Table 1 shows the EERs of our systems on the development and evaluation data sets. The first column of the table describes the features and classifiers that each system used. The results given in the first part of the table show the performance of systems with a single model, followed by results of systems with model fusion.

Firstly, we investigated systems with CQCC features. As can be seen from the table, ResNet performs the best. It reduces the EER by about 34% relatively, compared with the GMM model on the evaluation data. ResNet also outperforms the conventional DNN. We believe that this is because ResNet enable deeper model to be trained and thus generalizes better to unseen test data. We further evaluate the best model with MFCC features. When the number of filters was increased from 23 (as a typical setting in speech recognition task) to 60, we saw a big improvement. Furthermore, the ResNet with the MFCC feature even outperforms the ResNet with CQCC feature on the evaluation data.

Different systems provides different perspective about the data. Model fusion is a good way to utilize the advantage of different systems to improve the performance. Model fusion may introduce feature diversity or model diversity into the fused system. To evaluate which one is more important, our first fused model used the same CQCC feature but different models (DNN and ResNet). As can be seen, this fused model hardly improves the system performance. We then go on to fuse a system with the same ResNet but different features (MFCC and CQCC). This time the fused system significantly reduces the EER by 8.5% relatively, compared with the best single-model system on the evaluation data. Maybe we can draw a conclusion that for model fusion, feature diversity is more important than model diversity.

The last two lines of Table 1 show two systems that are fused with three models. Fusing more models can still improve the performance. The best fused system outperform the best single-model system by about 18% relatively on the evaluation data.

5. Conclusion

This paper presents our submission to the automatic speaker verification spoofing challenge (ASVspoof 2017). Through our experiments, we found that conventional MFCC features with mel-frequency scale filters along the whole speech bandwith may miss some important information between genuine and spoofed speech, leading to poor system performance. To avoid information loss, we propose to simply increase the number of filters. On the evaluation data, MFCC features perform better than CQCC featues. Our second finding is that residual neural networks can help us build deeper models that outperform conventional deep neural networks. Finally, we demonstrated that model fusion is a good way to further improve the system performance. However, if the same featue is used for different fused models, the resulting system can hardly be improved. By using different features and models, the system performance can be significantly improved. In the future, we would like to explore the effectiveness of using more advanced generative adversarial networks for the spoof detection task.

6. Acknowledgement

This work is supported in part by the National Natural Science Founding of China (61601187, U1636218), the Fundamental Research Funds for the Central Universities (2017MS045), and Guangzhou Key Lab of Body Data Science (201605030011).

7. References

[1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[2] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[3] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[4] 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.

[5] J. Villalba, A. Miguel, A. Ortega, and E. Lleida, "Spoofing detection with dnn and one-class svm for the asvspoof 2015 challenge," in *Interspeech*, 2015, pp. 2067–2071.

[6] X. Zhao, Y. Wang, and D. Wang, "Deep neural networks for cochannel speaker identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4824–4828.

[7] J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *ICCST*. IEEE, 2011, pp. 1–8.

[8] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, T. Leino, and A.-M. Laukkanen, "I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry." in *Interspeech*. Citeseer, 2013, pp. 930–934.

[9] Z. Kons and H. Aronowitz, "Voice transformation-based spoofing of text-dependent speaker verification systems." in *Interspeech*, 2013, pp. 945–949.

[10] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2280–2290, 2012.

[11] A. R. Dhanush Kannangola, Suparna S, "Factor analysis methods for joint speaker verification and spoof detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017.

[12] F. Alegre, A. Amehraye, and N. Evans, "Spoofing countermeasures to protect automatic speaker verification from voice conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 3068–3072.

[13] L. Wang, Y. Yoshida, Y. Kawakami, and S. Nakagawa, "Relative phase information for detecting human speech and spoofed speech." in *Interspeech*, 2015, pp. 2092–2096.

[14] A. Janicki, "Spoofing countermeasure based on analysis of linear prediction error." in *Interspeech*, 2015, pp. 2077–2081.

[15] M. J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis, "Development of crim system for the automatic speaker verification spoofing and countermeasures challenge 2015." in *Interspeech*, 2015, pp. 2072–2076.

[16] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, 2017.

[17] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: a survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[18] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," *manuscript, submitted to Interspeech 2017*.

[19] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah, "Investigation of sub-band discriminative information between spoofed and genuine speech," *Interspeech 2016*, pp. 1710–1714, 2016.

[20] 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.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

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

[23] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[24] S. Chitroub, "Classifier combination and score level fusion: concepts and practical aspects," *International Journal of Image and Data Fusion*, vol. 1, no. 2, pp. 113–135, 2010.

[25] B. Solaiman, R. Debon, F. Pipelier, J.-M. Cauvin, and C. Roux, "Information fusion, application to data and model fusion for ultrasound image segmentation," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 10, pp. 1171–1175, 1999.

[26] M. Vatsa, R. Singh, and A. Noore, "Improving iris recognition performance using segmentation, quality enhancement, match score fusion, and indexing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 1021–1035, 2008.