ATTENTIVE FILTERING NETWORKS FOR AUDIO REPLAY ATTACK DETECTION

Cheng-I Lai[1,2], *Alberto Abad*[2,3], *Korin Richmond*[2], *Junichi Yamagishi*[2,4], *Najim Dehak*[1], *Simon King*[2]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA
[2]Centre for Speech Technology Research, University of Edinburgh, UK
[3]INESC-ID / Instituto Superior Técnico, University of Lisbon, Portugal
[4]Digital Content and Media Sciences Research Division, National Institute of Informatics, Japan

ABSTRACT

An attacker may use a variety of techniques to fool an automatic speaker verification system into accepting them as a genuine user. Anti-spoofing methods meanwhile aim to make the system robust against such attacks. The ASVspoof 2017 Challenge focused specifically on replay attacks, with the intention of measuring the limits of replay attack detection as well as developing countermeasures against them. In this work, we propose our replay attacks detection system - Attentive Filtering Network, which is composed of an attention-based filtering mechanism that enhances feature representations in both the frequency and time domains, and a ResNet-based classifier. We show that the network enables us to visualize the automatically acquired feature representations that are helpful for spoofing detection. Attentive Filtering Network attains an evaluation EER of 8.99% on the ASVspoof 2017 Version 2.0 dataset. With system fusion, our best system further obtains a 30% relative improvement over the ASVspoof 2017 enhanced baseline system.

Index Terms— ASVspoof, Anti-Spoofing, Spoofing Attack, Replay Attacks, Automatic Speaker Verification

1. INTRODUCTION

Automatic speaker verification (ASV) systems have become increasingly widespread in recent years with the advent of voice assistant and smart home devices. However, these systems may be vulnerable to presentation attacks, which are also known as spoofing attacks [1]. One way an impostor might attempt to fool an ASV system into accepting them as a target speaker is by mimicking the voice characteristics of a genuine user. In order to ensure the continued reliability of ASV technology, it is necessary to develop countermeasures to protect against such spoofing attacks. There are four types of spoofing attack [1]: Impersonation, Replay, Speech Synthesis, and Voice Conversion. The ASVspoof2017 Challenge is based on Replay attacks. The objective of the challenge is to develop countermeasures to defend against replay attacks, and to measure the limits of replay attack detection [2].

Previous work on the ASVspoof2017 Challenge data can generally be divided into three categories: Gaussian Mixture Model (GMM) and i-vector based systems [2, 3]; deep neural network-based systems [4, 5, 6]; and systems fusing the preceding models. In [2], Constant Q Cepstral Coefficient (CQCC) features together with a GMM classifier and i-vectors formed the enhanced baseline system for the Challenge. In [3], thorough experiments were conducted on GMMs, i-vectors and Multi-Layer Perceptrons. The authors of [4] employed Light Convolutional Neural Networks (LCNN) and stacked LCNNs with a recurrent neural network (RNN). In [5], the authors developed an evolution RNN for spoofing detection. We also noticed recently published work [7, 8, 9, 10, 11] on Version 1.0 and 2.0 dataset, and we have included and compared their results in our experiments section.

The goal of the work here is to develop a deep learning system that utilizes discriminative features in both the time and frequency domain for spoofing detection. The motivation is that clues for spoofing attacks may be time varying and only partially observable, perhaps in the noise between spoken words or the high frequency components of speech for example. However, we are not sure where these clues might be embedded within the feature spaces. Therefore, we desire a system that automatically acquires and enhances discriminative time and frequency features that are helpful for the detection of spoofing attacks. We achieve this by designing an attention mechanism-based filter to cancel or enhance features prior to a ResNet-based classifier.

We took inspiration from three prior studies when designing the attention mechanism-based filter. Stimulated training [12] encourages activations to group in an interpretable way by superimposing a phone set during neural network acoustic model training. This inspired us to look at ways of applying an attention mechanism in model training. The convolutional attention network [13] computes an attention matrix from speech spectrograms and embedded word sequences and multiplies it with the spectrogram prior to the classifier. This work inspired us to apply an attention mechanism prior to a classifier. The residual attention network [14] applies a bottom-up feedforward process and top-down attention feedback. This work inspired us to adopt similar processes within our filter.

As for the classifier, while a ResNet has already been applied to the Challenge data [6], we believed the network architecture may be further improved. Our ResNet-based classifier, which we term a Dilated Residual Network (DRN), uses convolution layers instead of fully connected layers, and we modify the residual units by adding a dilation factor. The filter and the classifier together compose our proposed method: Attentive Filtering Network (AFN).

The remainder of this paper is organized as follows: We first describe the Attentive Filtering Network in detail, including feature engineering, dilated residual network, attentive filtering, and optimization. Then, we briefly describe the three baseline systems that we re-implemented. This is followed by experimental setup, results and discussion. We end the paper with some concluding remarks.

2. ATTENTIVE FILTERING NETWORK

2.1. Feature Engineering

We created a unified time-frequency map from log power magnitude spectra (logspec) obtained via Fast Fourier Transform as the

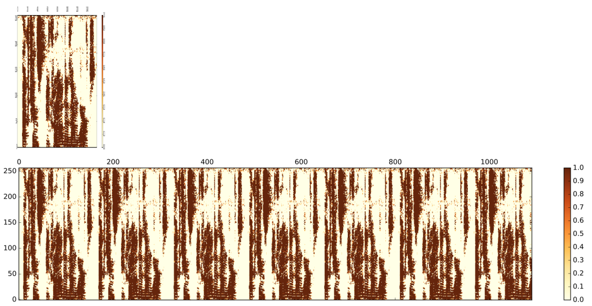

Fig. 1. An illustration of unified time-frequency map creation by extending the log power magnitude spectra of all utterances to the length of the longest utterance. **Top** is the original spectra and **Bottom** is the unified time-frequency map.

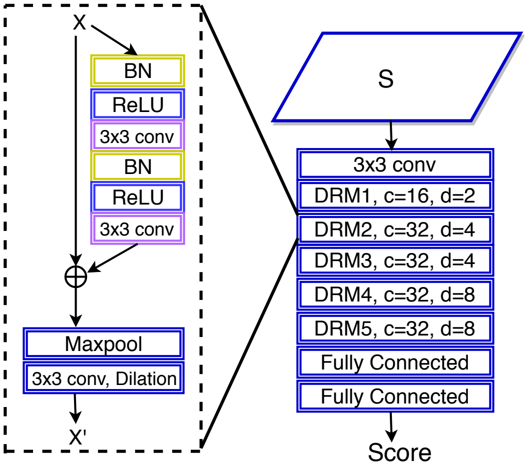

Fig. 2. Illustration of (**Left**) Dilated Residual Module (DRM) and (**Right**) Dilated Residual Network (DRN). S is the logspec input to the DRN, which is itself composed of five DRM blocks. Within each block, c is the channel dimension (16 & 32) and d is the dilation rate (2, 4 & 8). In addition to ReLU, we also experimented with ELU as an alternative activation function for the DRM.

input for our network. The dimension of logspec is 257. We kept all frames without applying voice activity detection, and applied mean normalization using a 3-second sliding window. The unified time-frequency map was created by extending all utterances to the length of the longest utterance by repeating their feature maps, illustrated in Figure 1. The resulting dimensionality of time-frequency maps for all utterances is 257 (frequency-domain) by 1091 (time-domain). The benefits of this feature engineering approach are that there is no need for truncating features [4], and since it is an utterance-level feature representation, there is no need for frame-level score combination.

2.2. Dilated Residual Network

A dilated residual network is composed of five Dilated Residual Modules (DRM), as shown in the right side of Figure 2. Each DRM has a 3×3 CNN based residual unit similar to [15], followed by a max-pooling layer and a dilated convolution layer, as illustrated in the left side of Figure 2.

The motivation for including a dilated convolution layer in the DRM arises from the observation that most of the previous ASVspoof 2017 Challenge-related research reported problems with generalizing to unseen conditions. In particular, the training set is small and the evaluation set contains very different conditions from those in the training and development sets. Therefore, prevention of overfitting is an important factor for obtaining good performance, and one idea to reduce the effects of overfitting in small datasets without compromising model capacity is to include dilation in convolution. Dilated convolution operation $*_d$ is defined as [16]:

$$(F *_d G)(\mathbf{n}) = \sum_{\mathbf{m_1}+d\mathbf{m_2}=\mathbf{n}} F(\mathbf{m_1})G(\mathbf{m_2}), \forall \mathbf{m_1}, \mathbf{m_2}, \qquad (1)$$

where F is the feature map, G is the kernel, d is the dilation rate, $\mathbf{m_1}, \mathbf{m_2}$ and $\mathbf{n}$ are vectors. With dilated convolution, the network's receptive field grows exponentially with layer depth such that it integrates knowledge for the wider and global context [16]. Max-pooling layers permit reduction of the spatial dimension of the feature maps.

Table 1. Configurations of the five Dilated Residual Modules in the Dilated Reisudal Network

Block	DRM1	DRM2	DRM3	DRM4	DRM5
Convolution	3×3	3×3	3×3	3×3	3×3
Dilation Size	2	4	4	8	8
Receptive Field	7×7	15×15	23×23	39×39	55×55
Input Channels	16	32	32	32	32
Output Channels	32	32	32	32	32

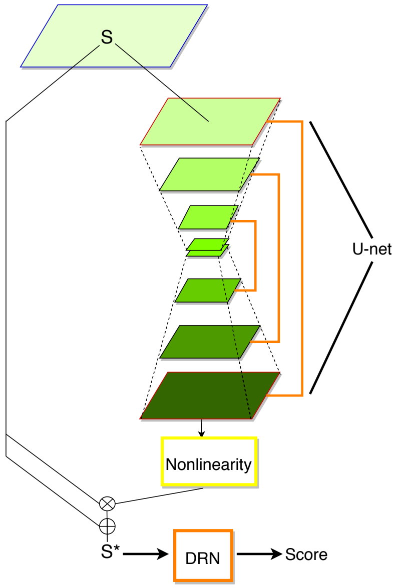

Fig. 3. Illustration of Attentive Filtering. Input **S** is a feature map, and output **S*** is the new input for the DRN. We experimented with four nonlinear transforms: *Sigmoid*, *Tanh*, *SoftmaxT* and *SoftmaxF*. Here, *SoftmaxT* means a *softmax* operation in the time domain, and a *SoftmaxF* operation means *softmax* in the frequency domain.

2.3. Attentive Filtering

Attentive Filtering (AF) accumulates discriminative features in frequency and time domains selectively. AF augments every input feature map $\mathbf{S}$ with an attention heatmap $\mathbf{A_s}$. The augmented feature map $\mathbf{S}*$ is then treated as the new input for the DRN as shown in Figure 3. For $\mathbf{S}, \mathbf{S}* \in \mathbb{R}^{F \times T}$, AF is described as:

$$\mathbf{S}* = \mathbf{A_s} \circ \mathbf{S} + \overline{\mathbf{S}}, \qquad (2)$$

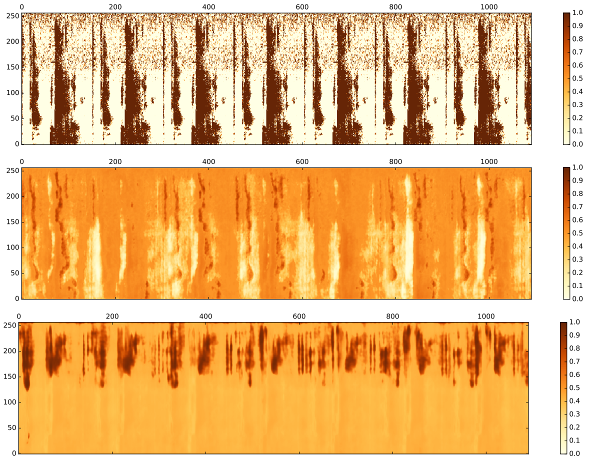

Fig. 4. Visualizations of attention heatmaps: the original feature map (top) and the corresponding attention heatmaps learned without $\overline{\mathbf{S}}$ (middle) and with $\overline{\mathbf{S}}$ (bottom).

where F and T are the frequency and time dimensions, $\circ$ is element-wise multiplication operator, $+$ is element-wise addition operator, and $\overline{\mathbf{S}}$ is the residual $\mathbf{S}$. In this work, we set $\overline{\mathbf{S}} = \mathbf{S}$. To learn the attention heatmap, $\mathbf{A_s}$ contains similar bottom-up and top-down processing as [14] and [17], and is described as,

$$\mathbf{A_s} = \phi(U(\mathbf{S})), \tag{3}$$

where ϕ is a nonlinear transform such as *sigmoid* or *softmax*, U is a U-net like structure, composed of a series of downsampling and upsampling operations, and $\mathbf{S}$ is the input. As in [14], we used max-pooling for downsampling and bilinear interpolation for upsampling. In addition, skip connections between the corresponding bottom-up and top-down parts are added to help learn the attention weights.

In contrast to [14], the attention weights in the model we propose are learned in the feature domain directly rather than in convolved domains. The motivation for this is twofold. First, the ASVspoof2017 dataset is much smaller than ImageNet and the approach of [14] could underfit with the amount of training data available. Second, and more importantly, attention heatmaps at the input-feature level are much more readily interpretable than at subsequent convolution levels.

Figure 4 shows attention heatmaps learned with and without $\overline{\mathbf{S}}$. As we can clearly see, the attention strongly focuses on high frequency components of speech segments when we trained the attention heatmaps with $\overline{\mathbf{S}}$, which is consistent with findings reported in literature. The intuitive explanation is that by giving DRN full information of $\mathbf{S}$, AF can focus on learning attention weights instead of learning a summary for $\mathbf{S}$. We can also see that with $\overline{\mathbf{S}}$, AF can selectively attend to and enhance not only high frequency segments but also any time and frequency segments.

2.4. Optimization

AF and DRN are the two components of the Attentive Filtering Network, and the network is trained end-to-end. The Attentive Filtering Network is initialized with Xavier initialization [18] and optimized with Adam with AMSGRAD [19]. We also performed model selection based on equal error rates (EERs) measured on the development set after every training epoch, as we found this to yield better results.

3. EXPERIMENTS

3.1. Baseline Systems

CQCC-GMM: CQCCs [20] are derived using the constant Q transform, a perceptually motivated time-frequency analysis tool, and have been shown to be especially effective for spoofing countermeasures [20]. The baseline of the ASVspoof2017 Challenge uses CQCC features with a standard 2-class GMM classifier for genuine and spoofed speech [2]. For each utterance the log-likelihood score is obtained from both models and the final system score is computed as the log-likelihood ratio.

i-vector: The i-vectors [21] pack a variable length speech recording into a fixed-dimension embedding. Following previous work [2, 3], we experimented with a 64-mixture Universal Background Model (UBM) with 100-dimension i-vectors and a 128-mixture UBM with 200-dimension i-vectors. The i-vector extractors were trained on 30-dimensional CQCC features from the ASVspoof 2017 training set. We also length-normalized speaker-level i-vectors. The i-vectors were then averaged within each class, giving one i-vector representation for genuine speech and another i-vector representation for spoofed speech. We used two simple classifiers for the i-vectors: Gaussian linear generative model [22] and cosine similarity [21].

LCNN: The best system submitted to the ASVspoof 2017 Challenge was based on the LCNN [4], where a Max-Feature Map activation is used for a CNN [23]. We re-implemented the LCNN for audio replay attack detection according to the specification described in [4] and tested it on the ASVspoof 2017 Version 2.0 dataset.

3.2. Experimental Setup

All experiments in this work were conducted on Version 2.0 of the ASVspoof 2017 dataset [2]. The dataset has 1507 replay and 1507 *bona fide* files in the training set, 950 replay and 760 *bona fide* files in the development (*dev*) set, and 12008 replay and 1298 *bona fide* files in the evaluation (*eval*) set. Details of the dataset can be found in [2]. We used only the training set to train our system. The development set was used for model selection during validation and for tuning the logistic regression for system fusion. In this paper, we compare our proposed system with prior work on both Version 2.0 and Version 1.0 of the ASVspoof 2017 dataset. Regarding implementation, the CQCC-GMM system was adopted from the official MATLAB code in [2]. Log-spectrogram and i-vectors were extracted with Kaldi [24]. Our Attentive Filtering Network was implemented in PyTorch, and the model implementation and training can be found in our github repository[1].

System fusion combines the strengths of different models to improve overall performance. Fusion at the score level has been extensively used in ASVspoof2017 [3, 4, 6]. In this work, we used the BOSARIS toolkit [25] for score fusion. The output of the network was first normalized using the *dev* set. Then, logistic regression was conducted to derive the fusion weights and a bias.

3.3. Experimental Results of Single Systems

Table 2 compares various systems in terms of Equal Error Rates (EER) (%). Overall, our proposed networks achieve competitive results on the Version 2.0 dataset. The DRN system with ReLU activation function achieves 10.3 EER on the *eval* set, and the DRN with ELU activation function achieves 10.16 EER. By adding AF to the DRN, we can reduce EER further. Examining the effect of

[1] github.com/jefflai108/Attentive-Filtering-Network

Table 2. EERs (%) of the *dev* and *eval* sets of our single systems and published single and fusion systems. Parentheses after AF and DRN denote the nonlinear transforms used in AF and DRN, respectively.

Systems	*dev* EER	*eval* EER	Diff.
Version 2 dataset			
AF(Sigmoid)-DRN(ReLU)	6.55	8.99	2.44
AF(SoftmaxT)-DRN(ReLU)	6.62	9.28	2.66
AF(SoftmaxF)-DRN(ReLU)	6.52	9.34	2.82
DRN(ELU)	7.49	10.16	2.67
AF(Tanh)-DRN(ReLU)	6.87	10.17	3.30
DRN(ReLU)	6.69	10.30	3.61
MDF(fusion) [11]	-	**6.32**	-
qDFTspec [7]	-	11.43	-
CQCC-GMM(CMVN) [2]	9.06	12.24	3.18
i-vectors (Cosine Similarity)	8.99	14.77	5.78
i-vectors (Gaussian) [22]	8.81	15.11	6.30
LCNN (Our implementation)	**6.47**	16.08	9.61
Evolving RNN [5]	18.7	18.20	-0.50
CQCC-GMM	12.08	29.35	17.27
Version 1 dataset			
DLFS(fusion) [8]	3.98	**6.23**	2.25
MDF(fusion) [11]	-	6.54	-
LCNN [4]	4.53	7.34	2.81
ConvRBM(fusion) [9]	**0.82**	8.89	8.07
Multi-task [10]	4.21	9.56	5.35
ResNet [6]	10.95	16.26	5.31

different non-linear activation functions, AF with *Sigmoid* achieves 8.99 EER, followed by 9.28 for *SoftmaxT*, 9.34 for *SoftmaxF*, and 10.17 for *Tanh*. The Table also shows the absolute difference between EERs on the *eval* set and *dev* set. This indicates the extent to which the model may have been over-fitted to training samples. We see these differences for the proposed systems are small, indicating that the models generalize well even from the give small, unbalanced dataset.

While [4] reported 7.37 EER on the *eval* set of the Version 1.0 dataset, we could not replicate their results on the Version 2.0 dataset. With their LCNN, we obtained 16.08 EER and the difference between the *dev* and *eval* sets is 9.61, which implies that LCNN could be over-fitting to the training data. Our i-vector baseline with cosine similarity backend achieves 14.77 *eval* EER, which is consistent with results reported in [2].

3.4. Experimental Results using Fusion

The previous section showed that the performance of the proposed system varies depending on the nonlinear activation function used in the AF. We found this empirically because the learned attentions behave differently. Figure 5 shows attention heatmaps for different nonlinear transforms ϕ in Equation 2. We see that *SoftmaxT* and *SoftmaxF* enforce sparse activation, while *Sigmoid* shows activations in multiple time and frequency bins. The difference in activations can be simply explained from how the nonlinearities scale each feature dimension. *Sigmoid* scales each dimension independently while *Softmax* scales each dimension dependently, implying that only a few dimensions are activated and most dimensions are suppressed (as shown in Figure 5 that most values are near 0). *Softmax* works well for classification task but not in our context, where the scale of each dimension could be useful for detecting replay attacks.

Figure 5 also indicates that different nonlinear activation functions in AF adopt different aspects of the task, and since they may be complementary, we decided to fuse multiple AF systems using *Sigmoid*, *SoftmaxF* and *SoftmaxT* activation functions. Table 3 gives the results of our fusion systems. As expected, the individual AF systems were complementary and fusing them reduced EER further. The best results for *dev* and *eval* were 6.09 and 8.54 respectively, obtained by fusing outputs of two AFNs with *Sigmoid* and *SoftmaxT* nonlinearity.

Table 3. Fusion system results. Multiple AF systems using *Sigmoid*, *SoftmaxF* and *SoftmaxT* were used to generate different attention maps. DRN(ReLU) was then used to calculate scores from each of the attention maps. We then fused the scores.

Fusion Systems	*dev* EER	*eval* EER
AF(Sigmoid)+AF(SoftmaxF)	6.37	8.80
AF(Sigmoid)+AF(SoftmaxT)	**6.09**	**8.54**
AF(SoftmaxT)+AF(SoftmaxF)	6.39	8.98
All	6.29	8.67

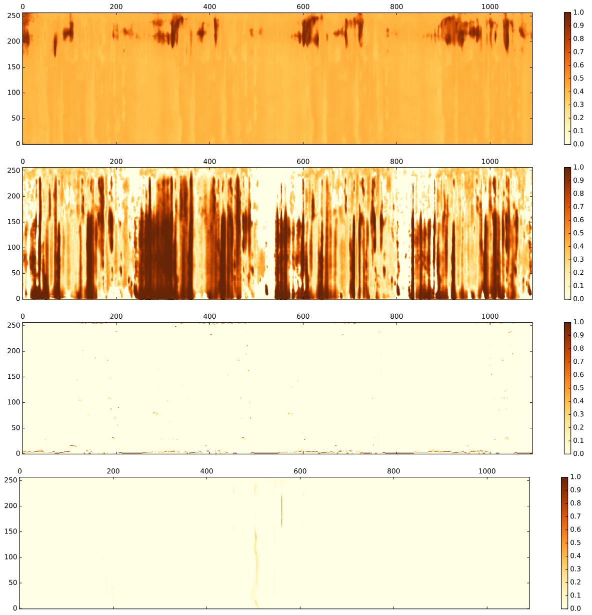

Fig. 5. Visualizations of attention heatmaps corresponding to different nonlinearities (from top to bottom): *Sigmoid*, *Tanh*, *SoftmaxF*, and *SoftmaxT*.

4. CONCLUSIONS

This paper presents our system for counteracting audio replay attacks. Our Attentive Filtering Network is composed of Attentive Filtering, which attends to and enhances input feature representation, and a Dilated Residual Network. Experiments conducted on the ASVspoof 2017 Version 2.0 dataset show the effectiveness of this model in replay attack detection, and furthermore, visualizing the attention heatmaps provides evidences for the network's feature enhancement behaviour. Our best single system achieved a competitive 8.99% evaluation EER, and our best fusion system provided 8.54% evaluation EER, providing a 30% relative improvement over the enhanced baseline system.

Acknowledgments This work was done at the University of Edinburgh; the authors thank the CSTR group there for helpful discussions. CL was supported by the Johns Hopkins Vredenburg Scholarship. JY was supported by JSPS KAKENHI Grant Numbers 18KT0051 and by JST CREST Grant Number JPMJCR18A6.

5. REFERENCES

[1] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[2] Héctor Delgado, Massimiliano Todisco, Md Sahidullah, Nicholas Evans, Tomi Kinnunen, Kong Aik Lee, and Junichi Yamagishi, "Asvspoof 2017 version 2.0: meta-data analysis and baseline enhancements," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 296–303.

[3] Mohammad Adiban, Hossein Sameti, Noushin Maghsoodi, and Sajjad Shahsavari, "Sut system description for anti-spoofing 2017 challenge," in *Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017)*, 2017, pp. 264–275.

[4] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin, "Audio replay attack detection with deep learning frameworks," in *Proc. Interspeech*, 2017, pp. 82–86.

[5] Giacomo Valenti, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, and Laurent Pilati, "An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 288–295.

[6] Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu, "Resnet and model fusion for automatic spoofing detection," in *Proc. Interspeech*, 2017, pp. 102–106.

[7] Md Jahangir Alam, Gautam Bhattacharya, and Patrick Kenny, "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 393–398.

[8] MS Saranya and Hema A Murthy, "Decision-level feature switching as a paradigm for replay attack detection," .

[9] Hardik Sailor, Madhu Kamble, and Hemant Patil, "Auditory filterbank learning for temporal modulation features in replay spoof speech detection," *Proc. Interspeech 2018*, pp. 666–670, 2018.

[10] Hyejin Shim, Jeeweon Jung, Heesoo Heo, Sunghyun Yoon, and Hajin Yu, "Replay attack spoofing detection system using replay noise by multi-task learning," *arXiv preprint arXiv:1808.09638*, 2018.

[11] Gajan Suthokumar, Vidhyasaharan Sethu, Chamith Wijenayake, and Eliathamby Ambikairajah, "Modulation dynamic features for the detection of replay attacks," *Proc. Interspeech 2018*, pp. 691–695, 2018.

[12] A Ragni, Chunyang Wu, MJF Gales, J Vasilakes, and Katherine Mary Knill, "Stimulated training for automatic speech recognition and keyword search in limited resource conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4830–4834.

[13] Chan Woo Lee, Kyu Ye Song, Jihoon Jeong, and Woo Yong Choi, "Convolutional attention networks for multimodal emotion recognition from speech and text data," *arXiv preprint arXiv:1805.06606*, 2018.

[14] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang, "Residual attention network for image classification," *arXiv preprint arXiv:1704.06904*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[16] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[18] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[19] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar, "On the convergence of adam and beyond," 2018.

[20] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.

[21] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[22] David Martinez, Oldřich Plchot, Lukáš Burget, Ondřej Glembek, and Pavel Matějka, "Language recognition in ivectors space," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[23] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[24] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[25] Niko Brümmer and Edward de Villiers, "The bosaris toolkit user guide: Theory, algorithms and code for binary classifier score processing," *Documentation of BOSARIS toolkit*, 2011.