The SJTU Robust Anti-spoofing System for the ASVspoof 2019 Challenge

Yexin Yang†, Hongji Wang†, Heinrich Dinkel, Zhengyang Chen, Shuai Wang, Yanmin Qian, Kai Yu

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

{yangyexin, jijijiang77, richman, feixiang121976, yanminqian, kai.yu}@sjtu.edu.cn, chenzhengyang117@gmail.com

Abstract

The robustness of an anti-spoofing system is progressively more important in order to develop a reliable speaker verification system. Previous challenges and datasets mainly focus on a specific type of spoofing attacks. The ASVspoof 2019 edition is the first challenge to address two major spoofing types - logical and physical access. This paper presents the SJTU’s submitted anti-spoofing system to the ASVspoof 2019 challenge. Log-CQT features are developed in conjunction with multi-layer convolutional neural networks for robust performance across both subtasks. CNNs with gradient linear units (GLU) activations are utilized for spoofing detection. The proposed system shows consistent performance improvement over all types of spoofing attacks. Our primary submissions achieve the 5th and 8th positions for the logical and physical access respectively. Moreover, our contrastive submission to the PA task exhibits better generalization compared to our primary submission, and achieves a comparable performance to the 3rd position of the challenge.

Index Terms: anti-spoofing, spoofing detection, variational autoencoder, convolutional neural network

1. Introduction

As a convenient and reliable method for identity authentication, automatic speaker verification (ASV) [1] has attracted researchers’ attention in recent years and gradually become mature, which makes it commercialized such as applications in call centers, security measures, etc. However, the ASV technologies are vulnerable, which makes ASV systems exposed to various spoofing attacks. Therefore, researchers manage to develop effective anti-spoofing systems, also known as presentation attack detection (PAD) systems, to protect ASV systems from malicious spoofing attacks.

At the beginning stage, researches were carried out in diverse datasets using different evaluation metrics, which made the results incomparable. In order to gather a community with standard databases and performance measures, a series of anti-spoofing competitions were born, for example, the Automatic Speaker Verification Spoofing and Countermeasures (ASVspoof) challenges that serve as special sessions in INTERSPEECH 2013 [2], 2015 [3], 2017 [4] and 2019, respectively. ASVspoof 2013 aimed at raising this serious spoofing problem, but no specific or appropriate solution was proposed.

ASVspoof 2015 focused on speech synthesis (SS) and voice conversion (VC), known as logical access condition (LA), while ASVspoof 2017 was designed to develop countermeasures capable of discriminating between bona fide (genuine) audios and replay ones, known as physical access condition (PA). Equal error rate (EER) is the common metric shared by them. ASVspoof 2019 covers both LA and PA but is divided into two separate subtasks.

To enhance the performance of anti-spoofing systems, recent works mainly focus on two approaches. One is to improve the front-end features extracted from audios [5, 6, 7, 8], where GMMs or LightCNN models are usually used as the classifiers. Another approach is to design new deep learning models [9, 10, 11, 12, 13] that learn discriminative representations for this task. Both of these two methods have been shown effective, which suggests that using appropriate front-end features as well as excellent deep learning models are both vital to the spoofing detection.

The rest of the paper is organized as follows, Section 2 briefly introduces the task of ASVspoof 2019 challenge, and Section 3 describes the features we used in the challenge. Section 4 will present the CNN based models and further explore the capabilities of GLU activations. The experiment details and results are given in Section 5. Section 6 concludes the whole paper.

2. Task Description

For better assessment of countermeasures for various spoofing attacks, ASVspoof 2019 challenge comprises two subtasks: logical access (LA) and physical access (PA).

2.1. Logical Access

Logical access (LA) spoofing attacks refer to spoofed speech generated with text-to-speech (TTS) and voice conversion (VC). As the widely use of neural-network-based systems in TTS and VC communities, the quality of generated speech is comparable to human speech, which brings new challenges to the spoofing detection system.

In the ASVspoof 2019 challenge, training data includes spoofed utterances generated according to two voice conversion and four speech synthesis algorithms, while spoofed algorithms in evaluation data are all unseen in the training set. Strong robustness is a requirement for our proposed spoofing detection systems.

2.2. Physical Access

Physical access (PA) spoofing attacks, also known as replay attacks, are performed at the sensor level. Since the somewhat uncontrolled setup in ASVspoof 2017 challenge makes the results difficult to analyze, the acoustic and replay configurations are carefully simulated and controlled in ASVspoof 2019 challenge, thus bringing some new insights into the replay spoofing problem.
In this work, we experiment with features extracted from the phase spectrogram ($e^{j\omega}$). Specifically, log-CQT and LMS features are extracted from the phase spectrogram in addition to the traditional magnitude spectrogram.

VAE log-CQT refers to use Variational Autoencoder (VAE) to extract genuine speech specific feature. All bona fide LA log-CQT features are used to train a VAE, which encodes data to 32-dim vectors and then try to reconstruct. Those vectors are our desired features, which are supposed to be meaningful on genuine data and be randomly distributed on spoofing speech.

4. CNN based Spoofing Detection

Convolutional neural network (CNN) based models are used as our classifiers because of their promising performance in [17, 18]. In addition to the heavily investigated models such as ResNet and LightCNN, the use of gated linear unit activation within CNNs for spoofing detection is proposed.

4.1. ResNet

A standard 18-layer ResNet comprised of 8 residual blocks is adopted as one of our single systems. The detailed configuration can be found in Table 1.

4.2. ResNet with i-vector

In order to enhance the generalization capability of our neural network model, i-vector is concatenated to the ResNet embedding layer as an additional feature for joint training. Compared to the naive GMM approach, i-vector is a factor analysis based method which can reduce the impact of spoof-independent factors. The architecture is depicted in Figure 1. In this work, the 400-dim i-vector extracted from log-CQT features is concatenated to a 128-dim ResNet18 embedding.

Table 1: Detailed Configuration of ResNet model. T denotes the frame number of input utterance and D denotes the feature dimension. Kernel sizes are set to 3 × 3.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Output Size</th>
<th>Channels</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv</td>
<td>$T \times D$</td>
<td>16</td>
<td>-</td>
</tr>
<tr>
<td>Res1</td>
<td>$T \times D$</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Res2</td>
<td>$T_2 \times D$</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>Res3</td>
<td>$T_4 \times D$</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>Res4</td>
<td>$T_8 \times D$</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>Average</td>
<td>128</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Linear (embedding)</td>
<td>128</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 1: The proposed ResNet + i-vector architecture. The inputs to the ResNet model and i-vector extractor are features (which are log-CQT + phase and log-CQT in this work, respectively) extracted from the same utterance.

4.3. LightCNN with multi-task outputs

Following the best system in ASVspoof 2017 challenge [17], a 9-layer LightCNN with max filter map (MFM) activation function is proposed. The general architecture of LightCNN model using multi-task outputs is shown in Table 2. The outputs of FC8.output refer to the spoofing labels (1 bona fide node and 1 spoofing node), while the outputs of FC8.output2 are the replay configuration labels (1 bona fide node and 9 replay configuration nodes, seen in Section 5.1). The sum of the outputs in both bona fide nodes is regarded as the detection score.

4.4. Context Gate CNN

In this work we further explore the capabilities of gated linear unit (GLU) activations. This activation function has been used in related tasks such as audio event detection (AED) [19], sound event detection [20], speech recognition[21] as well as natural language processing [22]. GLU can be seen as an alternative to the MFM activation used in the LightCNN. In this work, GLU halves the input tensor over the CNN filter dimension ($B$ and $A$) and uses one of those filters as weights and applies those weights on the other $f(A, B) = \sigma(A) \times B$ (see Figure 2). Here $\times$ is the Hadamard product of two tensors and $\sigma$ is the sigmoid activation function.

This activation acts as a context-gate for each filter, which is the reason to denote this network as context gate CNN (CGCNN). A single context gate of our network can be seen in Figure 2. The context gate architecture in this work strictly follows our LCNN approach (see Table 2), however small changes
the use of random oversampling the minority class (bona fide) many bona fide and spoofed utterances. Therefore we adopt ones, one needs to ascertain that the trained model sees equally within the training data set is only a fraction of the bona fide in stratified fashion. Since the number of spoofed utterances into a 90% training and 10% held-out cross-validation portion

\[ \text{model} \] nal evaluation. Before training, we split the given train dataset lowest cross-entropy loss on the held-out set was chosen for fi-

\[ \text{Model training for all experiments was ran for at most 200} \]

\[ \text{CGCNN and predicted posterior probabilities.} \]

\[ \text{CGCRNN. This GRU model was fed abstract features from the} \]

\[ \text{GLUs in order to avoid over-fitting (} C_1 = 48, C_2 = 96, C_3 = 192). 2) No multi-task training was utilized. 3) Statistics Pooling referred to mean pooling only.} \]

\[ \text{Moreover, for the final system fusion of our LA submission, we also incorporated a bidirectional gated recurrent unit (BGRU) model into the CGCNN model, further referred as CGCRNN. This GRU model was fed abstract features from the CGCNN and predicted posterior probabilities.} \]

\[ \text{5. Experiments} \]

\[ \text{Model training for all experiments was ran for at most 200} \]

\[ \text{epochs using adam optimization where the model producing the} \]

\[ \text{lowest cross-entropy loss on the held-out set was chosen for fi-

\[ \text{nial evaluation. Before training, we split the given train dataset into a 90% training and 10% held-out cross-validation portion in stratified fashion. Since the number of spoofed utterances within the training data set is only a fraction of the bona fide ones, one needs to ascertain that the trained model sees equally many bona fide and spoofed utterances. Therefore we adopt the use of random oversampling the minority class (bona fide) during training.} \]

\[ \text{5.1. Dataset and performance measures} \]

\[ \text{All experiments were conducted on the ASVspoof 2019 dataset respecting the official protocols on training/development divisions. For the LA subtask, 2,580 genuine and 22,800 spoofed speech utterances generated by one of 6 TTS/VC algorithms are used for training. The same spoofing algorithms in training set are used to create the development set, while the algorithms to generate the evaluation dataset are different. For PA task, the training set contains 5,400 genuine speech and 48,600 replay spoofing speech comprising 9 different replay configurations (3 categories of attacker-to-speaker recording distance times 3 categories of loudspeaker quality). The evaluation set for PA task has the same replay spoofing manner as training and development data, with different acoustic configurations. More details of the dataset can be found in ASVspoof 2019 evaluation plan}\]

\[ \text{To evaluate the performance of countermeasure, minimum} \]

\[ \text{tandem detection cost function (t-DCF) [23] is adopted as the} \]

\[ \text{primary performance metric, while equal error rate (EER) is} \]

\[ \text{used as a secondary metric.} \]

\[ \text{5.2. Evaluation on the LA task} \]

\[ \text{The components of our submitted system and their performance on development set is depicted in Table 3. Our single Context} \]

\[ \text{Gate CNN system with phase + log-CQT feature reaches 0.034} \]

\[ \text{and 1.09 in min-tDCF and EER, respectively. By fusing all sub-

\[ \text{systems together, better performance can be achieved, resulting} \]

\[ \text{in 0.027 and 0.90 in min-tDCF and EER, respectively. The fu-

\[ \text{sion system is submitted as our primary system.} \]

\[ \text{5.3. Evaluation on the PA task} \]

\[ \text{OpenSLR26, a simulated room impulse response database, is} \]

\[ \text{used for data augmentation for the PA task. Specifically, for} \]

\[ \text{each genuine speech in the training set, 20 randomly-chosen} \]

\[ \text{room impulse response are added. Thus a total number of} \]

\[ \text{108,000 RIR replicas are obtained.} \]

\[ \text{In order to avoid potential over-fitting, 2 different settings of} \]

\[ \text{hyper parameters } C_i (i = 1, 2, 3, 4, 5) \text{ are adopted for the} \]

\[ \text{1Refer to http://www.asvspoof.org/asvspoof2019/} \]

\[ \text{asvspoof2019_evaluation_plan.pdf for details.} \]

\[ \text{2Refer to http://www.openslr.org/26/ for details.} \]
Table 4: Primary submission results on the evaluation set for LA subtask in ASVspoof 2019 Challenge. The result indicated in bold is our submission.

<table>
<thead>
<tr>
<th>ranking</th>
<th>team</th>
<th>min-tDCF</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T05</td>
<td>0.0069</td>
<td>0.22</td>
</tr>
<tr>
<td>2</td>
<td>T45</td>
<td>0.0510</td>
<td>1.86</td>
</tr>
<tr>
<td>3</td>
<td>T60</td>
<td>0.0755</td>
<td>2.64</td>
</tr>
<tr>
<td>4</td>
<td>T24</td>
<td>0.0953</td>
<td>3.45</td>
</tr>
<tr>
<td>5</td>
<td>T50</td>
<td>0.1118</td>
<td>3.56</td>
</tr>
</tbody>
</table>

multi-task LightCNN (LightCNN-MT) models. The larger one (LightCNN-MT-L) uses (48,96,192,128,128), while the smaller one (LightCNN-MT-S) uses (16,32,64,48,48). Furthermore, both mean pooling (denoted as $\mu$) and mean+std (denoted as $\mu\sigma$) pooling are used, leading to 4 different models totally. LMS feature is used as input to our primary system, which is the score fusion of those 4 sub-models shown in Table 5.

Table 5: Performance of the 4 sub-models, primary as well as the constrastive submission on the development set for the PA subtask. $\mu$ indicates the mean pooling while $\sigma$ refers to the pooled standard deviation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Feature</th>
<th>min-tDCF</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LightCNN-MT-L-$\mu$</td>
<td>LMS</td>
<td>0.0180</td>
<td>0.59</td>
</tr>
<tr>
<td>LightCNN-MT-L-$\mu\sigma$</td>
<td>LMS</td>
<td>0.0189</td>
<td>0.71</td>
</tr>
<tr>
<td>LightCNN-MT-S-$\mu$</td>
<td>LMS</td>
<td>0.0235</td>
<td>0.88</td>
</tr>
<tr>
<td>LightCNN-MT-S-$\mu\sigma$</td>
<td>LMS</td>
<td>0.0221</td>
<td>0.79</td>
</tr>
<tr>
<td>Fusion (above 4)</td>
<td></td>
<td>0.0108</td>
<td>0.38</td>
</tr>
<tr>
<td>CGCNN</td>
<td>log-CQT</td>
<td>0.0092</td>
<td>0.35</td>
</tr>
<tr>
<td>CGCNN (RIR)</td>
<td>log-CQT</td>
<td>0.0078</td>
<td>0.31</td>
</tr>
<tr>
<td>Fusion (above 2)</td>
<td></td>
<td>0.0049</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Interestingly, our contrastive submission outperformed our primary submission on the evaluation set. Both of which significantly outperformed the baseline CQCC-GMM model in every replay configuration, shown in Figure 4. The contrastive model is a two way CGCNN fusion using the log-CQT feature - one being trained on the standard PA train set, while the other was trained on the augmented RIR data.

6. Conclusion

In this paper, we investigated multiple CNN based approaches, namely ResNet, LightCNN and most notably CGCNN for the ASVspoof 2019 challenge. Standard LMS as well as log-CQT features were used in conjunction with a newly uncertainty driven VAE model in order to ascertain robustness on development as well as evaluation subsets. Our results show that context-gated CNN networks are viable for both, logical and physical, scenarios. The proposed CGCNN model is shown to be reliable for both tasks. Our submitted system on the LA task, composed of a ResNet and CGCNN fusion, achieves a t-DCF of 0.027 on the development set and the 5th position on the evaluation set. On the other hand, our submission to the PA task, a LightCNN fusion, resulted in a t-DCF of 0.0108 on the development set and the 8th position on the evaluation set. Furthermore, our contrastive submission, a two way CGCNN fusion, outperformed our primary submission, achieving a comparable performance to the 3rd position.

7. Acknowledgement

This work has been supported by the Major Program of National Social Science Foundation of China (No.18ZDA293). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.
8. References


