

Convolutional Recurrent Neural Network and Multitask Learning for Manipulation Region Location*

Kang Li¹, Xiao-Min Zeng¹, Jian-Tao Zhang¹ and Yan Song¹

¹National Engineering Research Centre of Speech and Language Information Processing, University of Science and Technology of China, Hefei, China

Abstract

In this paper, we present our proposed system on Audio Deepfake Detection Challenge (ADD) 2023 Track 2- Manipulation region location (RL). Speech synthesis and voice conversion technologies have been developed in the past few years. However, synthetic audio is harmful when used by criminals and it may attack the security detection system and bring security risks, therefore, fake audio detection technology is in urgent need. Audio Deepfake Detection Challenge extends the attack scenarios into more aspects including manipulation region location where an audio is manipulated by other fake or true audio. To make localization more accurate, we apply the Convolutional Recurrent Neural Network where the CNN extracts high temporal resolution features and RNN models the context information. Besides, we apply linear-softmax pooling method to get the utterance-level true-fake determination which is a weighted sum of frame-level detection scores. To train a noise-robust model, we add MUSAN noise and reverberation to the raw audio. Our system ranked third in ADD 2023 Track 2 with achieving 54.49% segment-based F1-score and 79.50% utterance-level accuracy.

Keywords

Audio Deepfake Detection, Manipulation region location, CRNN, linear-softmax

1. Introduction

Audio Deepfake Detection (ADD) is the task to detect the fake audio which is synthesized. Manipulation region location (RL) is a subtask of ADD, the fake audio is manipulated partially with other true or fake audio, it is the multitask with utterance-level detection and segment-based localization. Due to the complexity of audios in realistic scenarios, it is still very hard to distinguish fake audios from real ones, and the localization is more challenged as the manipulated region hides in true audio [1]. It is crucial to detect the fake audio, as it can undermine the robustness of broadly implemented biometric identification systems and can be harnessed by in-the-wild attackers for criminal usage [2, 3, 4]. Some methods have been explored to detect the fake audios [5, 6, 7, 8, 9].


Audio Deepfake Detection challenge, start from 2022, has greatly promotes the development of ADD [10]. In this year, it has three tracks: the track1 includes two subtasks with fake audio generation and detection which aims to generate more realistic fake audios and design more discriminative models to detect the generated fake audio respectively; the track2 is Manipulation region location (RL) where participants should not only determine the audio type in utterance-level but also localize the manipulated region; the track3 aims to recognize the algorithms of deepfake audios.

This paper presents our work for Track2-Manipulation

IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023), August 19, 2023, Macao, S.A.R

✉ likang0311@mail.ustc.edu.cn (K. Li)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

region location (RL) of the ADD 2023 challenge. We study how to design neural network structure and how to train the model in both clip-level and frame-level to increase both the detection and localization capability of the model. Specifically we apply the convolution recurrent neural network [11] (CRNN) as our backbone and we train the CRNN model in multitask learning manner.

In audio detection filed, such as sound event detection, CRNN is usually used to model local and global information. It has three parts: CNN, RNN and localization module. The CNN extracts temporal representation with limited receptive region to increase the discrimination among frames, but the global information is insufficient. Directly enlarging the receptive region of CNN with pooling or large kernel will decrease the discrimination among frames, which is harmful to frame-level detection task. In CRNN, after extracting the frame-level output embeddings with CNN, RNN is used to further model the context information which compensate for the global information loss of CNN. After RNN, the localization module (a classifier) is used to get frame-level predictions. Considering the strengths of CRNN, we introduce it for RL task instead of only using CNN. Besides, as this track evaluates the model performance in both frame-level (localization) and clip-level (detection), the CRNN model is trained in multitask learning manner, and to transfer the clip-level classification ability to frame-level and aggregate the frame-level predictions to clip-level prediction, linear-softmax pooling [12] is used. Finally, to train a noise-robust model, MUSAN noise [13] and RIR [14] are used as data augmentation.

The model is trained on development dataset, and eval-

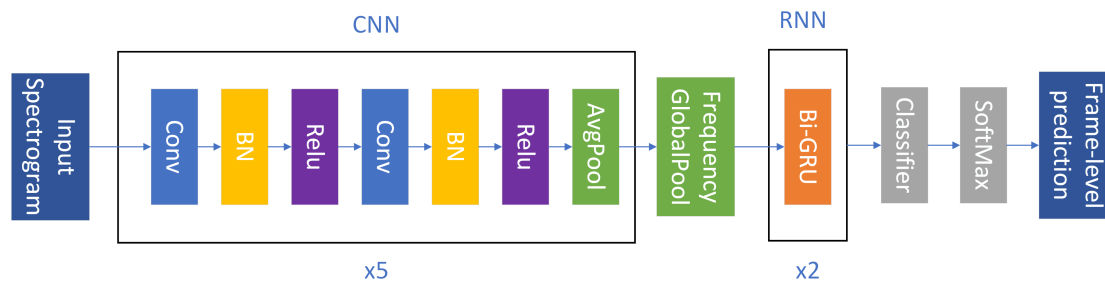


Figure 1: Convolutional Recurrent Neural Network.

uated on (1) official evaluation dataset ; (2) our synthetic partial fake audio dataset with the fully true and fake audio from track1.2 to analyze the models in more detail. Our system achieves 54.49% segment-based F1-score and 79.5% utterance-level accuracy on the official evaluation dataset, and also achieves 11.43% frame-level EER and 11.10% clip-level EER on our synthetic dataset.

The following of this paper is organized: in section 2, we introduce the CRNN model structure and the multitask learning method, in section 3, we introduce the dataset, data augmentation, feature extraction, experiment configurations and evaluation metric, in section 4 we evaluated the performance of different models and learning methods, and finally we conclude this paper in section 5.

2. Methods

2.1. Convolutional Recurrent Neural Network

The model structure of CRNN is shown in Figure 1, the CRNN model has three part: CNN, RNN and classifier+softmax. The input is log-mel spectrogram, it is feed to CNN block firstly, the CNN block has 5 sub-blocks, each sub-block is a VGG-style block which contains convolution block, batch normalization and ReLU activation function, they are stacked by Conv-BN-Conv-BN-ReLU and followed by average pooling. To maintain a high temporal resolution (as the detection is applied at every 0.01s), the average pooling is only applied along frequency-dimension. After the CNN block, frequency-wise global average pooling is used to get frame-level representations, then the output is feed to Bi-GRU which has 2 layers, finally a classifier with softmax is used to get frame-level prediction. The configuration of CRNN model is shown in Table 1.

Table 1

Configuration of CRNN. The config for CNN is denoted by (kernel_T,kernal_F)-(in_channels,out_channels)-(pool_stried_T,pool_stried_F). The config for Bi-GRU is denoted by (hidden dimension), the configuration for linear classifier is denoted by (in_channels,out_channels)

block	config	output shape
input	-	(400,41)
CNN1	(3,3)-(1,32)-(1,2)	(400,20,32)
CNN2	(3,3)-(32,64)-(1,2)	(400,10,64)
CNN3	(3,3)-(64,128)-(1,2)	(400,5,128)
CNN4	(3,3)-(128,128)-(1,2)	(400,2,128)
CNN5	(3,3)-(128,128)-(1,2)	(400,1,128)
Global pool	-	(400,128)
Bi-GRU1	(128)	(400,256)
Bi-GRU2	(128)	(400,256)
classifier	(256,2)	(400,2)
softmax	-	(400,2)
output	-	(400,2)

2.2. Multitask learning method

The relation of utterance fake and frame fake is bag-instance, and often obey the standard multiple instance (SMI) assumption: the bag label is positive if and only if the bag contains at least one positive instance. In clip-level, the model is trained in a multi-instance learning manner, while in frame-level, the model is trained in a standard supervised learning manner. We term the whole training method as multitask learning

Specifically given frame-level prediction $p_{frame} \in \mathbb{R}^{T \times 2}$, the 2 class denotes fake and true respectively. We select the frame-level fake prediction $p_{frame, fake} \in \mathbb{R}^{T \times 1}$, the clip-level fake prediction $p_{clip, fake} \in \mathbb{R}^1$ is a weighted average of $p_{frame, fake}$, it is implemented with linear-softmax pooling [12] method:

$$p_{clip, fake} = \frac{\sum_{i=0}^T p_{frame, fake, i}^2}{\sum_{i=0}^T p_{frame, fake, i}} \quad (1)$$

where i denotes the i^{th} frame, the $p_{clip, true} \in \mathbb{R}^1$ and

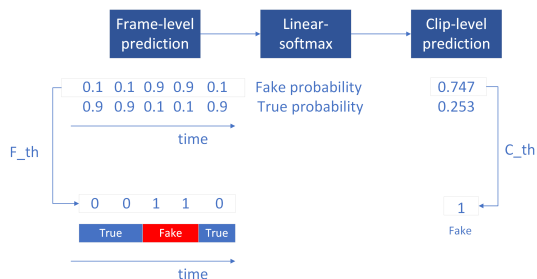


Figure 2: Manipulate region localization.

$p_{clip} \in \mathbb{R}^2$ are defined as:

$$p_{clip,true} = 1 - p_{clip,fake} \quad (2)$$

$$p_{clip} = \text{Concatenate}(p_{clip,fake}, p_{clip,true}) \quad (3)$$

Given clip-level prediction p_{clip} and frame-level prediction p_{frame} , the loss function for the data with batch size B is a sum of frame-level and clip-level Cross-Entropy loss in which is defined as follows,

$$L = -\frac{1}{B} \sum_{i=0}^B \sum_{k=0}^2 y_{clip,i,k} \log(p_{clip,i,k}) - \frac{1}{BT} \sum_{i=0}^B \sum_{j=0}^T \sum_{k=0}^2 y_{frame,i,j,k} \log(p_{frame,i,j,k}) \quad (4)$$

where the $y_{clip,i} \in \mathbb{R}^2$ and $y_{frame,i,j} \in \mathbb{R}^2$ denotes the clip-level and frame-level one-hot labels respectively, the i denotes the i^{th} sample, the j denotes the j^{th} frame.

With the linear-softmax pooling to aggregate the frame-level predictions to clip-level prediction, the gradient of clip-level loss also affects the frame-level learning, and the frame-level predictions are driven to the extremes 0 and 1, resulting in well-localized detections of sound events, which has been explored in [12]. Linear-softmax pooling is better than mean or max pooling, because the max pooling is affected by false positive predictions and the mean pooling shows worse performance when detect short duration manipulated region.

2.3. Manipulate region localization

As the fake region is very realistic, the fake probability is usually small when the model is evaluated on the evaluation dataset, therefore, we tune the threshold to a small value to get hard output from soft fake predictions, the manipulate region is further determined as shown in figure 2. And the utterance-level and frame-level predictions share different thresholds.

Table 2
Dataset description

DataSet	True	Full-fake	Partial-fake
Train	26554	1185	25354
Dev	8913	430	8480
Syn-Dev	2307	26017	8000
Evaluation	50000		

3. Experimental Setups

3.1. Datasets

The development dataset is ADD 2023 track2 dataset provided by organizers, the audio samples of train and dev set are shown in Table 2. In our exploration, we find the dev set are too easy to detect and localize fake audios, therefore, we synthesis another dev set with cut-paste the audio from track1.2 dev set, also shown in Table 2, and we mainly validate our model on the synthetic-dev dataset.

3.2. Input representations

The input sound clip is first cut or pad to 4s, and log Mel-spectrogram is extracted, which is based on short-time Fourier transform (STFT), the window size of fast Fourier transform (FFT) is 25ms, the hop size is 10ms, the FFT number 512, the mel filter number is 41. As a result, each 4-second sound clip is transformed into a 2D time-frequency representation with a size of (400×41) as the model input after instance mean-std normalization.

3.3. Data augmentation

We perform on-the-fly data augmentation by adding noise from MUSAN dataset and perform room impulse response (RIR) simulation.

3.4. Train settings

The model is trained for 10 epochs with learning rate of 0.01, SGD is used as optimizer with momentum of 0.9 and weight decay of 1e-4. We set balanced weight for CE loss based on the duration of true and fake region.

3.5. Evaluation metric

Our models are firstly evaluated on syn-dev dataset with frame-level and clip-level EER metric. Then the model is evaluated on the official rank metrics including utterance-level accuracy (ACC) and segment-based F1-score. Specifically, given the number of frame-level true positive (TP),

Table 3

Performance on the Syn-Dev dataset

Model	Pooling	DataAug	Frame-level EER, %	Clip-level EER, %	SB-F1, %
CNN	Linear-softmax	Yes	13.41	11.16	22.25
CNN	Linear-softmax	No	14.96	12.21	16.67
CNN	Max	Yes	13.41	12.16	22.25
CNN	Mean	Yes	13.41	11.28	22.25
CRNN	Linear-softmax	Yes	11.43	11.10	87.46
CRNN	Linear-softmax	No	17.21	11.78	42.23
CRNN	Max	Yes	11.43	12.36	87.46
CRNN	Mean	Yes	11.43	12.10	87.46

false positive (FP), false negative (FN) samples, the precision, recall, segment-based F1-score (SB-F1) is calculated,

$$P = \frac{TP}{TP + FP} \quad (5)$$

$$R = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = \frac{2PR}{P + R} \quad (7)$$

given the number of utterance-level positive (TP), false positive (FP), true negative (TN), false negative samples (FN), the ACC is calculated,

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

the ranking score is a weighted sum of SB-F1 and ACC,

$$Score = 0.3 \times ACC + 0.7 \times F1 \quad (9)$$

4. Results

In this section, we firstly evaluate our models on the syn-dev dataset to show the performance of CRNN model, linear-softmax pooling, and data augmentations, in this case, we use a fixed threshold of 0.5 to determine the fake detection. Then we report the results on the leaderboard of our models with tuning a best threshold to balance the precision and recall rate.

4.1. Evaluation on syn-dev dataset

As shown in Table3, the CRNN model with linear-softmax pooling and data augmentation achieves best results of 11.43% frame-level EER, 87.46%segment-based F1 and 11.10% clip-level EER. With out data augmentation, the performance decreases by a large margin, which shows that the MUSAN noise and RIR help train a robust model. Without GRU, the performance also decreased, especially the SB-F1, which show that GRU helps model the context information which is beneficial for detection task. Replacing linear-softmax with max pooling or mean pooling

Table 4

Performance on the evaluation dataset with different model and augmentation

Model	ACC, %	SB-F1, %	Score, %
CRNN+Aug+linear	79.50	54.49	62.02
CRNN+linear	53.6	33.6	39.2
CNN+Aug+linear	70.20	36.65	46.71
CNN+linear	67.2	28.9	40.42

Table 5

Performance on the evaluation dataset with different pooling function and thresholding. linear denotes linear-softmax pooling, fix denotes used fixed threshold of 0.5

Model	ACC, %	SB-F1, %	Score, %
CRNN+Aug+linear	79.50	54.49	62.02
CRNN+Aug+linear+fix	75.61	53.47	60.11
CRNN+Aug+mean	66.05	54.49	57.90
CRNN+Aug+max	78.70	54.49	61.70

achieves worse results which shows the linear-softmax pooling is a better aggregation method to pool the frame-level predictions to clip-level prediction (we only change the pooling function in the test stage, as simply use mean or max pooling in the training stage achieve worse results and they do not obey the standard multiple instance (SMI) assumption [12]).

4.2. Leaderboard results on evaluation dataset

As shown in Table 4, Our best model CRNN-Aug-linear achieves best results among our submissions with 54.49% SB-F1 and 79.50% ACC, which ranked third in the challenge. Without GRU or augmentaion, the performance decreased by a large margin, the evaluation may share different data distribution with the development, therefore, data augmentation will reduce the overfitting to the training set. As there are more parameters in CRNN and the feature space becomes larger after using RNN to modeling the context information among frames, the CRNN

model is easier to overfit to training data compared with CNN, therefore, without data augmentation, the performance of CRNN decreased a lot and is even worse than CNN. As shown in Table 5, without thresholding (i.e., use fixed threshold of 0.5), the performance decreases, which shows that the fake audio in the evaluation dataset may be more realistic, resulting in low prediction probability, and threshold independent metric may be more appropriate for evaluation. Finally, the linear-softmax pooling also achieves best performance on the evaluation dataset compared with mean and max pooling.

5. Conclusion

In this paper, we present our submitted systems on the ADD 2023 challenge. To be specific, we use the CRNN model as backbone and train the model in a multitask learning manner. The RNN models the context information which is beneficial to detection task, and in the multitask-learning, linear-softmax pooling helps get more precise clip-level fake detection results. MUSAN and RIR are used as data augmentation to make the model generalize to unseen data. Our final model achieves the 54.49% SB-F1 score and 79.5% ACC which ranked third on the leaderboard. In the future, we are aiming to design better models and study how to synthesis more realistic fake audios for training models.

References

- [1] J. Yi, Y. Bai, J. Tao, Z. Tian, C. Wang, T. Wang, R. Fu, Half-truth: A partially fake audio detection dataset, arXiv preprint arXiv:2104.03617 (2021).
- [2] H. Wu, Y. Zhang, Z. Wu, D. Wang, H.-y. Lee, Voting for the right answer: Adversarial defense for speaker verification, arXiv preprint arXiv:2106.07868 (2021).
- [3] Z. Wu, S. Gao, E. S. Cling, H. Li, A study on replay attack and anti-spoofing for text-dependent speaker verification, in: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, IEEE, 2014, pp. 1–5.
- [4] S. Liu, H. Wu, H.-y. Lee, H. Meng, Adversarial attacks on spoofing countermeasures of automatic speaker verification, in: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2019, pp. 312–319.
- [5] H. Wu, X. Li, A. T. Liu, Z. Wu, H. Meng, H.-y. Lee, Adversarial defense for automatic speaker verification by cascaded self-supervised learning models, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 6718–6722.
- [6] Z. Peng, X. Li, T. Lee, Pairing weak with strong: Twin models for defending against adversarial attack on speaker verification., in: Interspeech, 2021, pp. 4284–4288.
- [7] H. Wu, H.-C. Kuo, N. Zheng, K.-H. Hung, H.-Y. Lee, Y. Tsao, H.-M. Wang, H. Meng, Partially fake audio detection by self-attention-based fake span discovery, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 9236–9240.
- [8] Z. Lv, S. Zhang, K. Tang, P. Hu, Fake audio detection based on unsupervised pretraining models, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 9231–9235.
- [9] J. M. Martín-Doñas, A. Álvarez, The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 add challenge, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 9241–9245.
- [10] R. F. X. Y. C. W. T. W. C. Y. Z. X. Z. Y. Z. Y. R. L. X. J. Z. H. G. Z. W. S. L. Z. L. H. L. Jiangyan Yi, Jianhua Tao, Add 2023: the second audio deepfake detection challenge, IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023) (2023).
- [11] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, Convolutional recurrent neural networks for polyphonic sound event detection, IEEE Trans. Audio, Speech, and Language Proc. 25 (2017) 1291–1303.
- [12] Y. Wang, J. Li, F. Metze, A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling, in: IEEE ICASSP, 2019, pp. 31–35.
- [13] D. Snyder, G. Chen, D. Povey, Musan: A music, speech, and noise corpus, arXiv preprint arXiv:1510.08484 (2015).
- [14] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, S. Khudanpur, A study on data augmentation of reverberant speech for robust speech recognition, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 5220–5224.