Synthetic Speech Attribution Classification with EfficientNet and Cascade Approach at SPCup ICASSP 2022

Tri-Nhan Do*
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
dotrinhan99@gmail.com

Minh-Khanh Nguyen-Ngoc*
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
nguyenngocminhkhanh4999@gmail.com

Bang-Tam Nguyen-Ngoc*
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
bangtamnguyenn@gmail.com

Tuan Phan
Posts & Telecommunications Institute of Technology
Ho Chi Minh city, Vietnam
tuanpv.phan@gmail.com

Huy Nguyen
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
ntienhuy@fit.hcmus.edu.vn

***Abstract*—The synthetic speech attribution task in IEEE Signal Processing Cup[1] aims to determine if a computer-generated utterance is synthesized from a set of known speech generation algorithms and assign the utterance to its corresponding algorithm. This is an open-set multi-class classification problem with 6 labels (5 known algorithms, other than the known is regarded as unknown). Classification and clustering methods based on the efficientNet model are experimented. The best results for the public test of the methods were 93.2% for part 1 and 94.3% for part 2 including noise audios.**

***Index Terms*—synthetic speech, openset, efficientNet**

I. Introduction

Synthetic speech refers to any computer-generated utterance. Thanks to advances in deep learning, synthetic speech is getting closer to sounding like a human voice, which, however, might be exploited to impersonate and defame one's identity. Beyond a real/ fake audio classification, the task of synthetic speech attribution seeks to associate a synthetic utterance with its generation algorithm. Moreover, in the real world, classification is usually open-set, which means that unknown classes should be rejected or detected during testing. As a result, the system should adapt to unseen synthesizing approaches.

II. Method

Two major approaches are employed to address this task: classification and clustering.

A. Classification with thresholded softmax for unseen class

We employ efficientNet [3] to train a classifier on 5 known classes, with softmax activation in the last layer to compute probabilities for each label. We then use 500 audios to estimate the appropriate threshold, and a sample is classified as unknown if the probability of its most probable class is below this threshold.

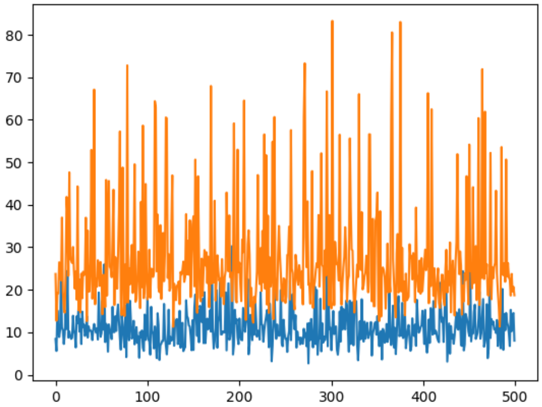

Fig. 1. Confident score for 500 test samples, orange part for known-class, blue part for unknown-class.

As illustrated in 1, the known and unknown classes are still not separated if using this method, there are still some known-class samples that have low probability and are mis-classified as unknown. The decision boundary becomes more unclear when data augmentation is applied in part 2 scenario.

B. Classification with Unseen as class

Selecting threshold for classification depends heavily on manual tuning at inference step. In this approach, we consider the unseen class provided by the organizer as a regular class, and build a 6-label classifier using efficientNet b3.

C. Cascade classification

The common training of all 6 classes containing both unknowns causes many outlier unknowns and known-classes,

*Equal contributions

[1] https://2022.ieeeicassp.org/signal_processing_cup.php

interfering with the inference process. We therefore separate into two models with two steps.

To begin, efficientNet b0 is used to solve a binary classification problem between seen and unseen algorithms. In the second stage, we use efficientNet b7 or efficientNet V2 to determine which of the five known classes was used to generate the given utterance.

It is worth noting that the binary classification sub-problem suffers from class imbalance, with seen data being five times more than unseen data. [1].

D. TuPlet for Clustering

In this challenge, the authors also introduced a variant of triplet loss to improve the classification process when unknown class appears.

The melspectrograms are passed through a backbone model to generate a vector of 1000 dimensions. This vector is considered a feature of the input sample. Based on the idea of triplet loss, we construct a new loss function with 4 components.

$$\begin{aligned} loss = D(anchor, positive) - D(anchor, negative) \\ -D(anchor, unknown) - D(negative, unknown) \end{aligned}$$

Where anchor is the feature vector of the input sample, positive denotes samples of the same label as the anchor, negative denotes the sample of different labels from the anchor, and unknown is the feature of the unseen class.

Intuitively, the loss aims to pull feature vectors of the same label to a cluster in space, and push samples with different labels and unknowns further away. D is a distance function, which can be cosine similarity or arc loss.

In the inference process, based on the vectors of the known classes, find the centroid for each class. The predicting samples will be calculated the distance to each centroid to determine which class belongs to.

E. Class Anchor Clustering

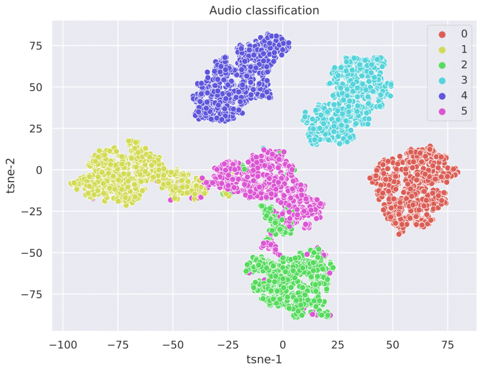

Fig. 2. Illustration of t-SNE on testset for 5 known algorithms and unknown class.

This method uses an entirely distance-based loss the explicitly around training data to form tight clusters class-dependent anchor points in the logit space [4]. In addition, we also experiment with variations of CAC loss, which is with the addition of margin $\mathcal{N}$ in the formula (article), similar to margin in triplet loss or SVM. This loss encourages the positive centroid distance to be smaller than the minimum negative centroid distances among which are at least greater than the positive distance plus the margin constant.

$$\mathcal{L}_T = log(1 + \sum_{j \neq y}^{\mathcal{N}} e^{d_y - d_j + \mathcal{N}})$$

F. Meta learning - few short classification

The experiment using meta learning is Prototypical Network [5] and the base learner is using EfficientNet B0 with 5-short, 5-query and 10 sample in support set. In addition, we also have an experiment, instead of Each prototype is the mean vector of the embedded support points belonging to its class. We use Kmedoids algorithm to choose the sample with minimum distance with other sample in the same class. This can help model more robust to outlier in support set.

$$C_i = \text{argmin}_i \sum_{j=1}^{\mathcal{N}} d(x_i, x_j), 1 \leq i \leq \mathcal{N}$$

III. Experiment

A. Data analysis

The utterances' duration span from 1.35s to 14.76s with an audio sample rate of 16000, and each audio is cut into segments of 24000 samples and fed into the detector.

The audios are then feature-extracted to form a mel-spectrogram of 80 mel channels with a 1024-window size and a 256 hop-length.

B. Data augmentation

In order for a deep learning model with many parameters to be able to learn different aspects of the problem, as well as be more robust with unseen data, a large enough amount of data is required, one of the ways is to generate data from available set.

Authors use two prominent augmentation methods in audio: noise addition and SpecAugment.

1) Audio Augmentation: With noise augmentation, the data is processed before training. The Matlab Audio toolkit and ffmpeg library are used to add noise and reverberation to the audio.

- Noise addition SNR in dB, randomized from 5 to 15 dB, with a Probability of 0.8.
- The volume of audio is randomly scaled between -2 and 2, then normalized to the domain [-1,1] to avoid audio clipping.
- We do not use pitch control or time stretch because it can change the voice which affect the attribute of each algorithm.
- To simulate voice in real environment, reverberator is used. The decay factor, which is time takes for reflections

TABLE I
BENCHMARKING ACCURACY OF EACH METHOD ON PUBLIC TEST SET

Method	Part 1	Part 2
Threshold	89.4	67.8
6-class	93.2	92.4
Cascade	91.1	94.3
CAC	87.4	N/A
Meta learning	85	N/A

to run out of energy, gets random uniform in range [0.2. . . 0.8].

From 6000 original audios of 5 seen-algorithm and unseen audios, 30k audios data was generated based on the above methods.

2) SpecAugmentation: With specaugmentation [2], the augment process is random on the fly during training.

Each sample for training is a 125-frame segment random from mel spectrogram of an audio sample. These samples will be applied time masking with maximum possible length of the mask is 35. With frequency masking, maximum possible length of the mask is 12.

C. Training

The backbone model we use throughout the experiment is efficientNet with variant versions and efficientNet V2 model.

Most classification models use Cross Entropy and clustering methods use CAC loss or triplet loss.

For each method we tune the data with different random thresholds, training configurations with a learning rate of 0.001 and a learning rate decay 0.99 using exponential schedule.

Training results are exported to onnx format, then we use matlab deep learning toolbox for visualize model at each layer and infer for final result

IV. RESULTS

V. CONCLUSION

Synthetic Speech Attribute of open-set scenarios helps detect and classify artificial voices. Data augmentation methods are used to increase the amount of data and simulate speech in a real environment. The two main approaches are classification and clustering. With experiments based on the public test set, we find that the cascade classification method gives the highest and most robust results for part 2 with noise data. The SOTA methods of open-set clustering or the proposed method of Tulet yield low results on this challenge and need further experimentation.

REFERENCES

[1] Lin, Tsung-Yi, et al. ”Focal loss for dense object detection.” Proceedings of the IEEE international conference on computer vision. 2017.

[2] Park, Daniel S., et al. ”Specaugment: A simple data augmentation method for automatic speech recognition.” arXiv preprint arXiv:1904.08779 (2019).

[3] Tan, Mingxing, and Quoc Le. ”Efficientnet: Rethinking model scaling for convolutional neural networks.” International conference on machine learning. PMLR, 2019.

[4] Miller, Dimity, et al. ”Class anchor clustering: A loss for distance-based open set recognition.” Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021.

[5] Snell, Jake, Kevin Swersky, and Richard Zemel. ”Prototypical networks for few-shot learning.” Advances in neural information processing systems 30 (2017).