Hannah Babe and Alessandra Vennema
AIT Budapest, Spring 2022

Deep Learning: Audio Spoof Detection

Introduction

In the following documentation we provide a brief report of our final project, an application to identify computer generated audio. Such programs, notably ASV or Automatic Speech Verification, are becoming more common as alternative ways to protect secure information or accounts from scammers.

Previous Solutions

There are a number of advanced techniques and implementations of Audio Spoof detection. A yearly ASVSpoof content is held each year. Best methods include a combination of front end feature extraction and back end classifiers.

Dataset

Our database is the same one used for the Third Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2019) (http://www.asvspoof.org). The database is split into two sections, one for logical access (LA) scenarios, and one for physical access (PA). We chose to use data from the logical access. Both LA and PA databases are the further split into three datasets - training, development, and evaluation, which are composed of speech from 20 (8 male, 12 female), 10 (4 male, 6 female) and 48 (21 male, 27 female) speakers respectively. The three partitions are disjoint in terms of speakers, but are identical in recording conditions, and there is no significant channel or background noise effects. The fake/spoofed audio was then generated from the real data using a number of different spoofing algorithms, including several voice conversion and speech synthesis algorithms. The voice conversion systems included neural-network-based and transfer-function-based methods, while the speech synthesis systems included waveform concatenation, neural-network-based parametric speech synthesis using source-filter vocovers, and neural-network-based parametric speech synthesis using Wavenet. Spoofed audio was also created using publicly available toolkits including Merlin3, CURRENT4, and MaryTTS5.

While the training and development datasets contain spoofing attacks generated with the same algorithms or conditions (known attacks), the evaluation dataset contains spoof audio generated with different algorithms/conditions (unknown attacks). Therefore reliable detection models must generalize to previously unknown attacks.

In the LA subset, in the training dataset there are 2,580 ‘bonafide’ audio samples, and 22,800 spoofed samples. In the development dataset, there are 2,548 ‘bonafide’ samples and 22,296 spoofed samples. In the evaluation dataset there are upwards of 80k trials including genuine and spoofed speech meaning that the evaluation dataset is approximately 4 GB in size.

Proposed Method

A standard machine learning pipeline was implemented (Figure 1).

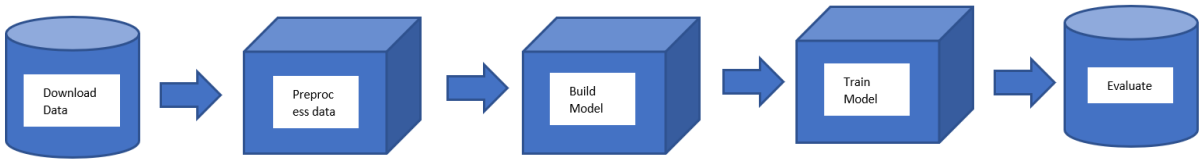

Figure 1: Machine Learning Pipeline

We began by downloading the data set. As seen below, our set had about 10x more real audio files compared to computer generated files (Figure 2). Hence, we implemented class weights to balance the data set.

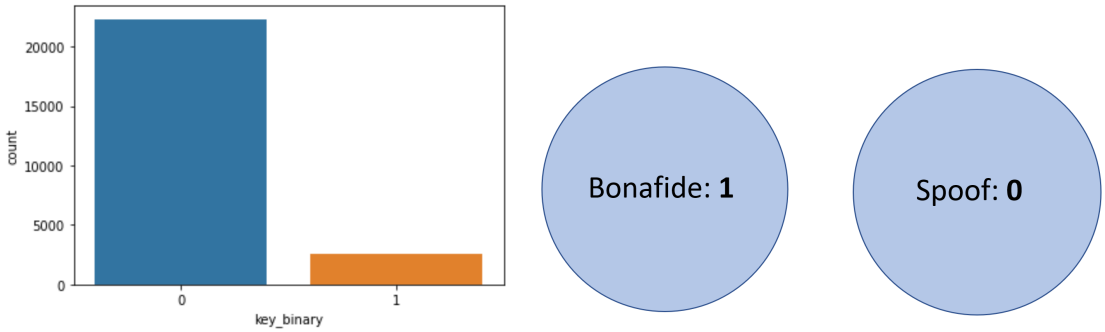

Figure 2: df_eval Elements

We then visualized the sound into spectrograms and then mel-spectrograms (Figure 3).

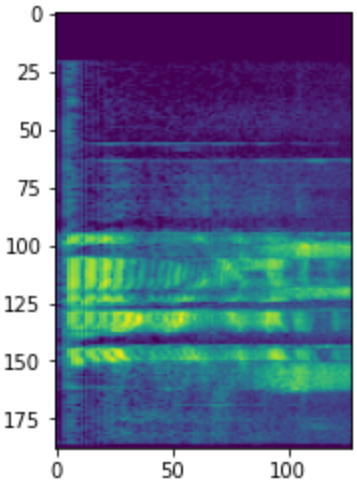

Figure 3: Sample Spectrogram

We defined a model using the following layers (Figure 4). We specified an optimizer and the loss function as categorical_crossentropy). We then shuffled the data and trained the model for 3 epochs.

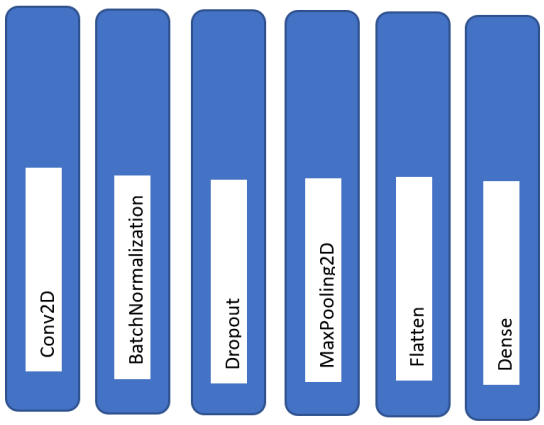

Figure 4: Model Layers

Evaluation Method

As we struggled to increase the accuracy of our model on the evaluation dataset, we only used the basic methods of model evaluation - running model.predict and model.evaluate. If we had achieved higher accuracy, we would have looked at the misclassification cases to see if there were any themes in misclassification to try and improve our model.

Results

Our model achieved an accuracy of 0.99 and loss of 0.0348. However when evaluated with the evaluation generator, the model only reaches a 54% accuracy. When running model predict we

ran into a problem where the model predicted all audio files from the evaluation dataset were spoofed. We were unable to determine the precise cause of this error.

```
[ ]  import seaborn as sns
     ax = sns.heatmap(conf, annot=True, fmt='d', cmap='Blues')
     ax.set(xlabel='Predicted Label',
            ylabel='True label');
```

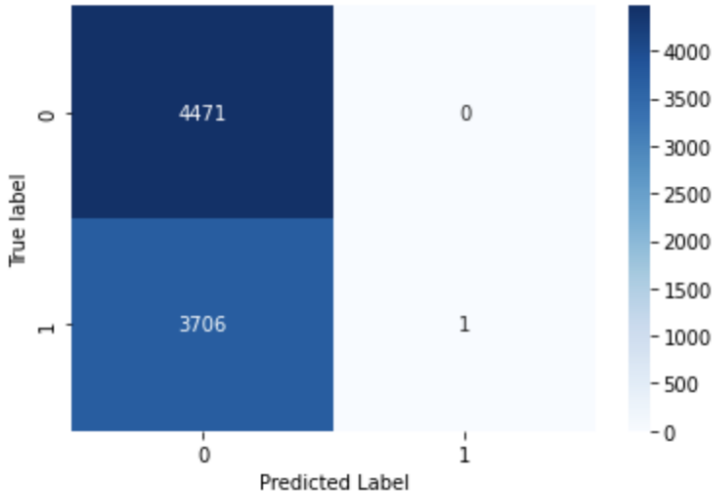

Figure 5: Confusion Matrix

```
[18] print(reconstructed_model.evaluate(evaluation_generator))

     127/127 [==============================] - 18s 48ms/step - loss: nan - accuracy: 0.5465
     [nan, 0.5465059280395508]
```

Figure 6: Model.evalue with Evaluation Generator

We successfully were able to understand, process, and use the data from the dataset we found, and were also able to create a functioning data pipeline.

Discussion

In a more detailed approach we would attempt to improve how our model generalizes so that it is able to accurately predict spoofed audio files. We could consider in the future using a pretrained model to accomplish this. We would like to figure out why our model is not able to differentiate between real and spoofed audio in the evaluation dataset, especially since there are spoofed audio files present in the evaluation dataset that would be previously unseen by the model - meaning it should be misclassifying spoofed audio files as real if it were misclassifying anything,

not real audio files as spoofed. We would also be interested in further looking into the practical applications of our project.