A No-Reference model for Detecting Audio Artifacts using Pretrained Audio Neural Networks

David Higham
Amazon Prime Video
highamdh@amazon.com

Ayush Bagla
Amazon Prime Video
ayushbgl@amazon.com

Veneta Haralampieva
Amazon Prime Video
venetah@amazon.co.uk

Abstract

This work presents a No-Reference model to detect audio artifacts in video. The model, based upon a Pretrained Audio Neural Network, classifies a 1 second audio segment as either: No Defect, Audio Hum, Audio Hiss, Audio Distortion or Audio Clicks. The model achieves a balanced accuracy of 0.986 on our proprietary simulated dataset.

1. Introduction

Audio artifacts can have a negative impact on the Quality of Experience for a watcher of a video streaming service. In this work we present a No-Reference model to detect 4 different manifestations of audio artifacts:

- Audio Hum - the addition of a constant frequency to the original waveform.
- Audio Hiss - the addition of white noise to the original waveform.
- Audio Distortion - the clipping of the original waveform.
- Audio Clicks - abrupt large transitions in the amplitude of the original waveform.

The model, based upon a Pretrained Audio Neural Network (PANN) [13], classifies a 1 second audio waveform as one of the 4 defective classes or a 5th 'No Defect' class.

The main contributions of this paper are:

1. An audio artifact dataset, created by simulating audio artifacts on a diverse dataset of defect free videos.
2. A model for detecting audio artifacts. The model achieves a balanced accuracy of 0.986 on the audio artifacts dataset.

2. Related work

2.1. Audio defect detectors

The majority of work in audio defect detection has been performed using classical signal processing techniques. [2, 3] identify Audio Hum using the Power Spectral Density over a 10-30 second window. This large analysis window means predictions are temporally coarse. Our model is applied to only 1 second of audio allowing a much higher temporal granularity of prediction.

[1, 17, 18] identify Audio Hiss and Audio Clicks by modeling the audio signal using autoregressive techniques. They assume that a future audio sample can be approximated using a linear combination of past samples. They predict a sample as defective if the prediction error, the difference between the actual and predicted sample, is greater than a threshold.

Recently, [19] detected degradations in audio signals by applying a Convolutional Neural Network to the log-mel spectrogram of the audio waveform. The authors demonstrated that a shallow model, trained using simulated data, could accurately identify noise, distortion and reverberation.

2.2. Neural networks for audio classification

Inspired by the success of Deep Learning in other domains such as Computer Vision [10, 15] and Natural Language Processing [5, 21], Convolutional Neural Networks (CNN) have become widely applied for audio classification [6, 11]. [11] utilize Youtube-100M to show that using the log-mel spectrogram as input to a CNN model can outperform fully connected baseline models. [6] explored using the raw waveforms directly with a ResNet [10] model. [14] investigate adapting a variety of different models for audio tagging. They demonstrate that their CNN14, a VGG [20] inspired network, can outperform previous systems on AudioSet [7]. Furthermore, they show that extending this network to utilize both log-mel spectrograms and wavegrams, a learnt representation of the original waveform, can lead to further improvement.

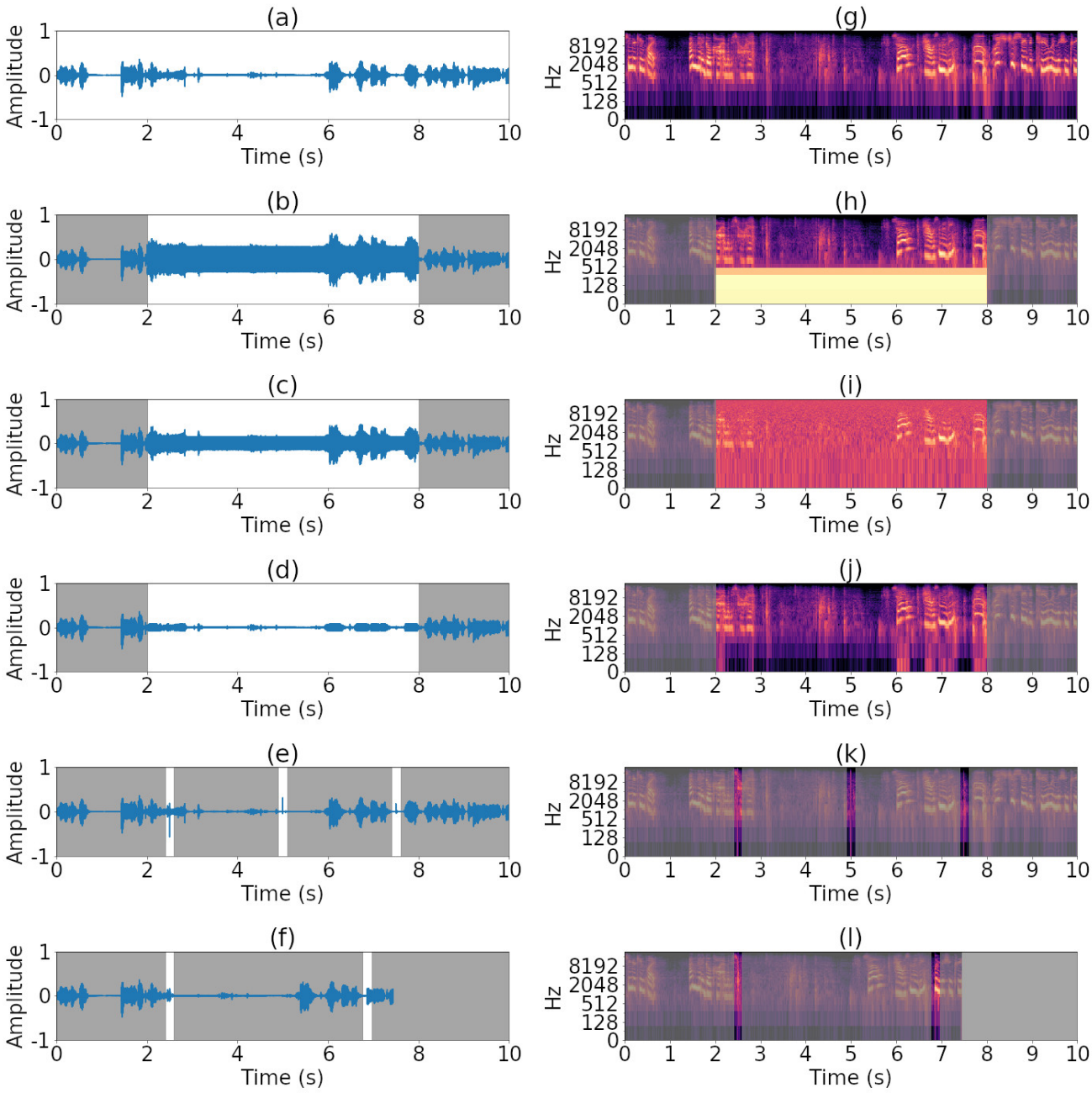

Figure 1. Waveforms and Spectrograms for each of the defects: (a) No Defect (b) Audio Hum. Defective interval between 2 and 8 seconds indicated by non-grayed area. Frequency = 50 Hz. Amplitude = 10000. (c) Audio Hiss. Defective interval between 2 and 8 seconds indicated by non-grayed area. Amplitude = 10000. (d) Audio Distortion. Defective interval between 2 and 8 seconds indicated by non-grayed area. Clipping Value = 0.2. (e) Audio Clicks - Additive. Clicks simulated at 2.5, 5 and 7.5 seconds indicated by non-grayed area. (f) Audio Clicks - Reductive. Clicks simulated at 2.5 and 7.5 seconds indicated by non-grayed area. Note that the waveform has a shortened duration because of the removal of samples. (g-l) The mel-spectrogram for each of the corresponding waveforms. The non-grayed area interval indicates the defective periods.

Recently, [9] explore various training techniques such as ImageNet pretraining, checkpoint averaging and improving the quality of AudioSet labels. [14, 16] investigated techniques to combine a Transformer model with a CNN, while [8] propose a convolution-free Transformer model, which achieves state-of-the-art results on AudioSet.

CNN6
Log-mel spectrogram 94 frames x 128 mel bins
5 x 5 @ 64 BN, ReLU
Pooling 2 x 2
5 x 5 @ 128 BN, ReLU
Pooling 2 x 2
5 x 5 @ 256 BN, ReLU
Pooling 2 x 2
5 x 5 @ 512 BN, ReLU
Global pooling
FC 512, ReLU
FC 5, Sigmoid

Figure 2. Network architecture used in this work. The architecture is a modified version of PANN-CNN6 first presented in [13]

3. Model

We formulate defect prediction as a multi-class classification:

$$\hat{y} = \arg\max_y p(y|x;\theta) = \arg\max_y f(x) \qquad (1)$$

where there are 5 classes: No Defect, Audio Hum, Audio Hiss, Audio Distortion and Audio Click. $f(x)$ is a Pretrained Audio Neural Network (PANN) [13], a framework of neural networks designed for multi-label audio classification [7]. Instances of the framework ingest an audio clip's waveform and output the probability that the clip exhibits each of the classes. Internally the model is a Convolution Neural Network applied to a log-mel spectrogram. A log-mel spectrogram is a time frequency representation of the clip's waveform.

We use transfer learning to fine-tune a PANN-CNN6 instantiation (Figure 2). The PANN framework includes a selection of high performing models. We chose PANN-CNN6, a light weight model, to meet our latency targets. We replace the final fully connected layer with a fully connected layer with a 5 element output. The output of this layer is passed to a Softmax function.

4. Audio artifacts dataset

Due to the low prevalence of audio defects within streaming video, we do not have a dataset of real samples to develop the model. Instead, we create a dataset by simulating the four defects of interest on defect free videos. The defect free content comprised of 127 hours from 162 videos. The videos were selected to be representative of typical streaming content. The selection criteria considered the genre and language of the videos. The videos were partitioned into train (103 videos), validation (28 videos) and test (31 videos) sets; ensuring that genre and languages were represented across partitions. The audio was demuxed from the videos, downmixed to stereo, resampled to 48 kHz and converted to 16-bit integer PCM. These clips provided the base audio and are used for the No Defect class.

To create the defective content we simulated defective intervals on the base audio. For each base audio we simulated each of the defects following Algorithm 1. There were two different simulation methods for the Audio Click defect, therefore we ended up with 6 versions (1 No Defect and 5 defective) of each base audio. The following subsections present the method of simulation for each of the defective classes, however first we discuss the post-processing performed once the defects have been simulated.

Algorithm 1: Simulation of defective intervals

```
Input: audio - defect free audio;
Input: defectParams - parameters required to apply
       the defect;
Input: minIntervalDuration - the minimum
       defective interval duration. Default: 0.5;
Output: defectiveAudio - audio with simulated
        defective intervals
defectiveAudio = copy(audio);
clean = random bool;
intervalStart = 0;
do
    intervalDuration = minIntervalDuration +
     Gamma(1, 10);
    intervalEnd = max(intervalStart +
     intervalDuration, len(audio));
    if not clean then
        for channel in channels do
            defectiveAudio =
             applyDefect(defectiveAudio,
             defectParams, intervalStart,
             intervalEnd);
        end
    end
    intervalStart = intervalEnd;
    clean = not clean
while intervalStart != len(audio);
```

We split each of the No Defect audio and defective audio into individual channels and then 1 second chunks. All No Defect chunks were added to the dataset. A defective chunk was added to the dataset if it's waveform was different to the corresponding No Defect chunk. The class label was set to that of the parent audio. Table 1 outlines the number of samples in each dataset.

Table 1. The number of samples in each class in each dataset.

Name	No Defect	Hum	Hiss	Distortion	Click	Total
Train	447168	241990	241870	174711	147742	1253481
Validation	146282	78976	80160	56423	49300	411141
Test	288702	146301	147672	104359	89210	776244

4.1. Audio Hum

Audio Hum is the addition of a single frequency tone to the original waveform. We simulated Audio Hum by adding a sine wave to the original waveform. The frequency of the sine wave was uniformly sampled from $\{x \in \mathbb{Z} | 20 > x > 1220\}$. The amplitude of the sine wave was uniformly sampled from $\{x \in \mathbb{Z} | 60 > x > 32768 - \max(originalWaveform)\}$.

4.2. Audio Hiss

Audio Hiss is the addition of white noise to the original waveform. We simulated Audio Hiss by adding random noise to the original audio waveform. The amplitude of the noise was uniformly sampled from $\{x \in \mathbb{Z} | 60 > x > 32768 - \max(originalWaveform)\}$.

4.3. Audio Distortion

Audio Distortion is the clipping of the original waveform. We simulated Audio Distortion by bounding the original waveform between $-value$ and $value$. Where $value = ratio \times \max(abs(originalWaveform))$ and ratio was uniformly sampled from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$.

4.4. Audio Click

Audio Clicks are abrupt large transitions in the amplitude of the waveform. We applied two different simulation methods for Audio Clicks: Additive and Reductive.

4.4.1 Additive

An additive Audio Click was simulated by adding a Butterworth filtered unit impulse to the signal [4]. Prior to filtering the amplitude of the impulse was 32768. We used a 3rd order Butterworth filter with a cut-off frequency uniformly sampled from $\{x \in \mathbb{R} | 0.05 > x > 0.45)\}$.

4.5. Audio Distortion

4.5.1 Reductive

A reductive Audio Click was simulated by removing a segment of the waveform. To ensure the removal results in an Audio Click we check that: 1) the segment's waveform has a zero crossing; and 2) the removal of the segment would result in a sample to sample amplitude difference of at least 3277.

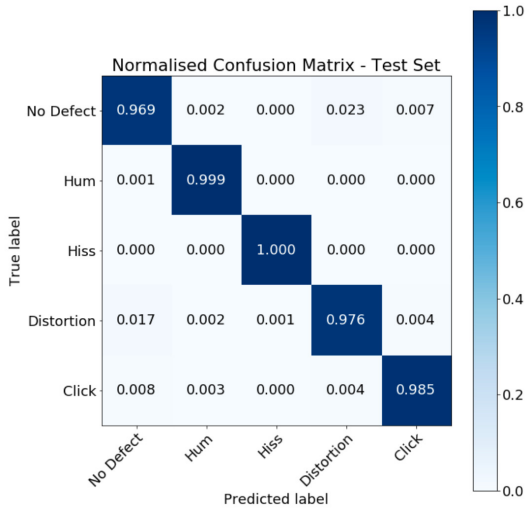

Figure 3. Normalised Confusion Matrix for the evaluation over the test set.

5. Training

The model was optimised over the train set using a negative log-likelihood loss. The loss was optimsied using Adam [12], with a learning rate of 0.0001, β_1 of 0.9 and β_2 of 0.999. The batch-size was 512 and the model was trained for 98000 iterations on a AWS EC2 p3.2xlarge instance.

Every 100 iterations the model was evaluated using the balanced accuracy over a 1% subset of the validation set. At completion of training, the parameters that gave the largest validation balanced accuracy were returned.

6. Results

The model achieves a balanced accuracy of 0.986 over the test set. Figure 3 displays the confusion matrix for the test set. We observe that the Audio Hum and Audio Hiss classes achieve a recall of over 0.99. Of the defective classes, recall is lowest for the Audio Distortion class; 1.7% of the time Audio Distortion is predicted as No Defect. Distortion has a sparse effect on the waveform; it only effects a fraction of the samples in the chunk. We hypothesize that the chunks that are incorrectly predicted as No Defect may have a very small fraction of distorted samples.

We have identified the following future work:

- While our defect simulation methods are derived from

literature, we have not validated they are aligned with a watcher's perception of a defect. Through a perception study, we will refine our simulation algorithms to more accurately reflect common audio defects observed in streaming video.

- The model was developed using simulated data due to insufficient real defective data being available. The simulation algorithm makes a set of assumptions that may not hold in the production environment. We will create a dataset of real audio defects so we can validate the classifier's performance on real streaming video.

7. Discussion

This work presented a No-Reference model for the identification of audio artifacts in video. The model based on a Pretrained Audio Neural Network identifies 4 common audio defects. To train, validate and test the model we created an audio artifacts dataset by simulating defects on streaming videos. Experimentation suggests that our method, which requires only 1 second of audio, achieves a balanced accuracy of 0.986.

References

[1] Pablo Alonso-Jiménez, Luis Joglar-Ongay, Xavier Serra, and Dmitry Bogdanov. Automatic detection of audio problems for quality control in digital music distribution. In *AES 146th International Convention*, 2019.

[2] Matthias Brandt. *Automatic Restoration of Audio Signals in Media Archives*. PhD thesis, 2018.

[3] Matthias Brandt and Joerg Bitzer. Automatic detection of hum in audio signals. *AES: Journal of the Audio Engineering Society*, 62(9), 2014.

[4] Matthias Brandt, Simon Doclo, Timo Gerkmann, and Joerg Bitzer. Impulsive disturbances in audio archives: Signal classification for automatic restoration. *AES: Journal of the Audio Engineering Society*, 65(10), 2017.

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, 2019.

[6] Logan Ford, Hao Tang, François Grondin, and James Glass. A deep residual network for large-scale acoustic scene analysis. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019.

[7] Jort F. Gemmeke, Daniel P.W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017.

[8] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. 2021.

[9] Yuan Gong, Yu-An Chung, and James Glass. PSLA: Improving Audio Event Classification with Pretraining, Sampling, Labeling, and Aggregation. 2021.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Shawn Hershey, Sourish Chaudhuri, Daniel P.W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017.

[12] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[13] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28(1), 2020.

[14] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D. Plumbley. Sound Event Detection of Weakly Labelled Data with CNN-Transformer and Automatic Threshold Optimization. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28, 2020.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 2017.

[16] Koichi Miyazaki, Tatsuya Komatsu, Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, and Kazuya Takeda. Convolution-Augmented Transformer for Semi-Supervised Sound Event Detection. Technical report, 2020.

[17] Rudolf Mühlbauer. *Automatic Audio Defect Detection*. PhD thesis, 2010.

[18] Laurent Oudre. Automatic Detection and Removal of Impulsive Noise in Audio Signals. *Image Processing On Line*, 5, 2015.

[19] Yuki Saishu, Amir Hossein Poorjam, and Mads Græsbøll Christensen. A CNN-based approach to identification of degradations in speech signals. *Eurasip Journal on Audio, Speech, and Music Processing*, 2021(1), 2021.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.