Deep Neural Network Classification of Reverberant Environments with AudioSet

Carl Moore
American University
4400 Massachusetts Ave NW,
Washington DC
carlmoore256@gmail.com

Abstract

This paper presents an implementation of a simple CNN for acoustic scene classification (ASC). The presented model was trained on 3,000 labeled audio samples from AudioSet, [2] a publicly available data-set of 2 million YouTube videos. Audio samples are split into sub-frames and fed into the network as mel-frequency cepstral coefficients (MFCCs), providing roughly 13,000 images for training and validation. The aim is to experiment with widely available computer vision models, and determine the validity of re-purposing a generalized vision system as a sonic classifier. Results demonstrate the need for more specific model architectures for audio classification of smaller data-sets. The relatively small subset used in this implementation suffered issues with over-fitting and under-fitting, and ultimately lacked enough properly labeled samples that shared similar timbral properties of an acoustic environment.

1. Introduction

The natural reflections within an acoustic field can provide useful context about a scene, such as whether it is an enclosed or open space, where objects are located and how they're moving, and possibly even the types of materials distributed throughout the space, among other things [3]. On a more basic level, acoustic scene classification (ASC) attempts to classify a scene's acoustic environment, and has been an active area of research in machine learning since as early as 1997 [8].

The high availability of microphone transducers in portable devices, in addition to their ability to resolve omni-directional features in a scene, makes them ideal for many scene classification tasks. Recent research has demonstrated improvements in testing accuracy using a multi-modal approach, by combining audio with image classification [5] [6].

Due to the effectiveness of CNN image classification models, a common practice for audio-related tasks has been to package audio signals into spectrograms, two-dimensional vectors of a signal's time and frequency components, and pass them through convolutional networks. This practice has proven to be fairly effective [9]. An even more effective two dimensional input for CNNs are MFCCs, which represent higher-level audio features than spectrograms [4]. MFCCs are obtained from the inverse-Fourier-transform of a log magnitude spectrum of a signal, perceptually weighted in the spectral domain using the Mel-filterbank.

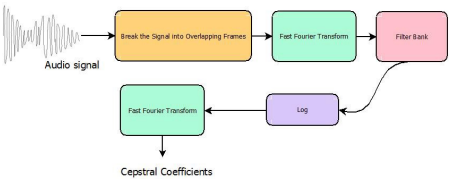

Figure 1. Process of generating cepstral coefficients

This paper describes an implementation of an ASC using the publicly available AudioSet as training and validation data for the classification of reverberant scenes. The aim is to classify several different reverberant conditions, such as a small room, a large hall, or an outdoor area. The experiments demonstrate different methods for tweaking the network for the input of raw cepstograms.

1.1. Related Work

Audio scene classification has been implemented in several experiments using CNNs. Hershey et al. provides a framework for general audio classification using CNNs, and presents a new data-set of YouTube videos called YouTube 100M. The data-set consists of around 5.24 million hours of video, each with 30,871 labels. The main goal of the experiment is to predict video labels using audio features. The experiment also includes a comparison with AudioSet, yielding lower accuracy scores due to the smaller size of AudioSet. The CNN model intakes mel-spectrograms of size 96x64, and compares various network architectures includ-

ing AlexNet, VGG, and ResNet-50 for spectrogram cassification. Their results show 93 percent accuracy on the testing set using this methond [9].

Acoustic Scene Classification is detailed by Valenti et al., demonstrates a 79 percent accuracy on the DCASE dataset, designed specifically for acoustic scene classification. Their implementation uses log-mel spectrograms and a convolutional neural network [4].

Pham et. al significalty improves on these results using a more complex architecture. Their experiment uses CNN pre-training to provide features to a DNN and a second CNN in parralel for classification. They achieve 90 percent accuracy on the DCASE dataset [7].

1.2. Data

AudioSet is a publicly available human-labeled data-set of over 2 million YouTube video clips at 10 seconds long, containing audio features and organized into an ontology [2]. The ontology's class "acoustic environment" formed the basis for the raw audio gathered for this experiment. This class is described as a set of videos containing the "spatiality of the recording," and is listed as containing 183,680 labeled samples in seven categories: "Inside, small room," "Inside, large room or hall," "Inside, public space," "Outside, urban or manmade," "Outside, rural or natural," "Reverberation," and "Echo." [1]. These seven categories are this experiment's classes.

AudioSet provides three sets: a balanced evaluation set, a balanced training set, and an unbalanced training set. For this experiment, training data was taken from the balanced training set, while validation data was taken from the evaluation set.

AudioSet contains features and links to the original videos, but does not provide the actual video and audio file, which is needed to generate the MFCC cepstograms. To further complicate matters, many videos in the dataset have since been removed or changed their privacy status, and are no longer accessible.

In order to obtain and process the data, a custom downloading and extraction program was implemented in Python. A parser identifies videos from the data-set that contain the appropriate class tags, downloads and strips the audio using youtube-dl, and extracts the 10 second segment using ffmpeg. To generate the MFCC images, a custom Python implementation was created to batch process the audio data and perform the conversion. The audio is first normalized, then processed using the librosa features module to generate an MFCC array. For the filterbank, 128 Mel-frequency bands were specified for this experiment, which is represented on the Y-axis of a cepstogram. The number of bands determines the resolution of the data - this implementation follows Pham et. al in using 128 bands. Other implementations choose fewer bands, such as Valenti et. al, who use 60 Mel bands. Our experiment did not cover a comparison of filterbank resolutions, which may have led to different results.

The mel-cepstogram images are subdivided into seven sub-frames sized 128-128 pixels, in order to optimize the input into the CNN. These frames represent roughly 1.4 seconds of audio each. It's worth noting that this process does separate individual training samples into several smaller components, providing the network with a greater number of images for training. However, future considerations need to be made for this experiment regarding the effect of this procedure. Because reverberant sections of audio are inherently temporal, this division might lead to mis-classification in the network if the training data is shuffled.

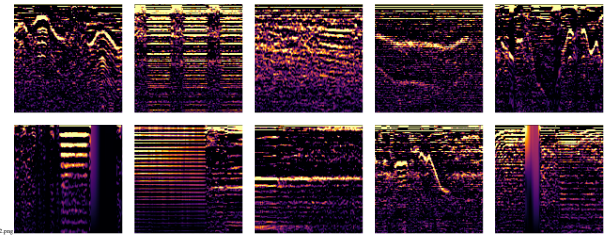

Figure 2. A series of MFCC cepstograms. The content of each frame represents 1.4 seconds of audio

1.3. Methods and Experiments

In this experiment, the task of ASC was handled by implementing a generic CNN template provided by Keras for use on the CIFAR 10 image data-set. The network itself is ideal for processing and classifying large volumes of low-resolution 32x32 images, which made it good candidate for implementing in the context of mel-cepstograms.

It is worth noting that much of the implementation of this model, including the tweaking of network design and hyperperameters, remains in the process of refinement. After identifying a large issue with over-fitting, the first two convolutional layers were changed from small inputs of 32x32 with a 3x3 kernel, to more closely match the model presented by Pham et al., with a 128x128 size fully connected convolution layer with a 5x5 kernel. My model still contains aspects of the CIFAR 10 Keras implementation, with ReLu activation following both convolution layers. Pham et al. implements max pooling layers in those locations, which is a future consideration when refining this model.

Identifying the issue of overfitting began at the early stages of building my model, which was built on a basic MNIST classifier, and was trained on thousands of instrument samples. This was before the experiment took a different direction, but the model demonstrated overfitting of specific instruments, reporting a 95 percent accuracy rate during the training stage, and a 40 percent accuracy score on the test data. I determined this was overfitting due to

the limited size of the data-set. To improve this, I manually gathered more samples, but ultimately decided to change to a larger data-set to create a better train-validation split.

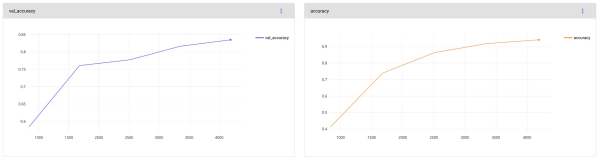

Figure 3. Early experiment classifying 19 different instruments. Data shows train and validation accuracy increase, but testing yielded low scores

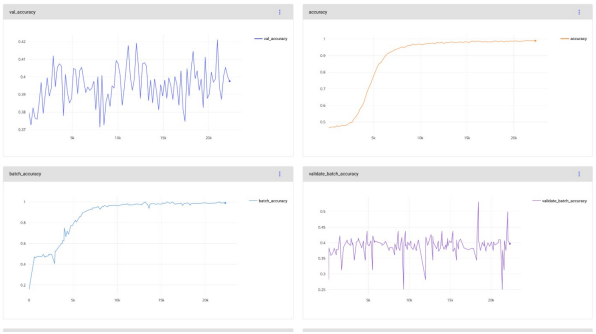

Figure 4. Initial tests with the reverberation data show validation score never passing 50 percent, while training accuracy quickly increases. Testing yielded low scores of around 15 percent

My final models reveal a low accuracy rate of around 15 percent during the testing stage. This could be due to many reasons, which will be discussed. The final model was trained with a much larger fully connected layer, and modifications to its design to make it closer to existing spectrogram classifiers. During training sessions, many hyperparameters were tweaked. Batch size ultimately was optimized at 32. The final model ran for 40 epochs, and took over six hours to complete on a GPU cluster.

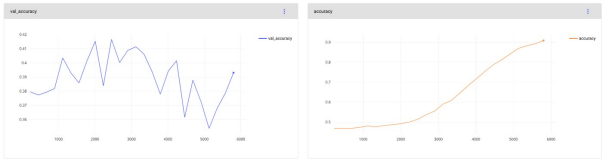

Figure 5. Current implementation after tweaking the model architecture still shows overfitting

1.4. Conclusion

There are many improvements that can be made to the model presented. One method to improve accuracy is to implement transfer learning, using a pre-trained CNN used to classify spectrograms. The final layers could be tweaked to fit the task of ASC. Another method to improve results would be to change the underlying desing to an RNN, or more ideally, an LSTM. Given the time-domain nature of reverberation, it would be beneficial to train a network taking into account temporal characteristics. Improvements in the experiment design from the input data perspective could be made as well. On average, it takes humans around 14 seconds to identify an acoustic environment [4]. This means sub-dividing the frames into equal height and length interrupts any continuous reverberation properties, and contributes to inaccuracy and over-fitting of unrelated information.

In terms of the data-set, the model could be significantly improved by re-arranging the training and validation splits. AudioSet provides an unbalanced training set with 2,042,985 total video segments, whereas this experiment used the balanced version with only 22,176 segments in total. The unbalanced set could be split between training and valdation, while the evaluation subset could be used for testing.

There are many possible uses cases for AudioSet, and the tools developed in this experiment contribute to the accessibility of implementing raw audio classification using it. With future experiments, I aim to improve the model in the methods discussed, and accurately perform acoustic scene analysis with a focus on reverberation characteristics of an environment.

References

[1] Audioset.

[2] D. F. A. J. W. L. R. C. M. M. P. M. R. Jort F. Gemmeke, Daniel P. W. Ellis. Audio set: An ontology and human-labeled dataset for audio events. *Google AI*, Jan 1970.

[3] P. McKerrow. Acoustic flow. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep 2008.

[4] A. D. G. P. T. V. Michele Valenti, Stefano Squartini. A convolutional neural network approach for acoustic scene classification. *2017 International Joint Conference on Neural Networks (IJCNN)*.

[5] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, page 639–658, 2018.

[6] A. Owens, P. Isola, J. Mcdermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] L. Pham, I. Mcloughlin, H. Phan, and R. Palaniappan. A robust framework for acoustic scene classification. *Interspeech 2019*, 2019.

[8] N. Sawhney. Situational awareness from environmental sounds. Jun 1997.

[9] D. P. W. E. J. F. G. A. J. R. C. M. M. P. D. P. R. A. S. B. S. M. S. R. J. W. K. W. Shawn Hershey, Sourish Chaudhuri. Cnn architectures for large-scale audio classification. *arXiv.org*, Jan 2017.