Classification of Acoustic Scenes Using Convolutional Neural Networks

Jinbin Bai
Shanghai Jiao Tong University, China
jinbin5bai@gmail.com

Abstract

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network, inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. In recent years convolutional neural networks are growing exponentially in the fields of computer vision, natural language processing and speech recognition. However, there is a dearth of research on detection and classification of acoustic scenes and events. In combination with IoT and wireless sensor networks, convolutional neural networks could help to characterize acoustic scenes and therefore better address noise issues present in urban environments.

This paper investigates the theory and construction of convolutional neural networks for classification of acoustic scenes like streets, supermarkets and cafes. A convolutional neural network architecture is proposed for our task. The inputs to the networks are time-frequency patches extracted from the computed mel-spectrogram of the signals. Dropout[1] and weight decay regularization mothods are applied and the cross-entropy loss is optimized using Adam algorithm.

1. Introduction

In the recent years, Cloud computing, the Internet of Things (IoT) or big data analytics are commonly used in people's daily lives. Additionally, the advent of low-cost and low-power transceivers has made the deployment of wireless sensor networks possible, allowing cities to monitor all kinds of acoustic scenes and events in real time.

This paper is originated from the idea of extracting valuable information from audio recordings. Neural networks are computing systems that have been widely used for computer vision problems and recently research [2, 3, 4, 5] has proved that convolutional neural networks are also suitable for audio recognition and classification problems.

The propose of this paper is to exploring how to use the convolutional neural networks to identify the specifical acoustic scenes present in recorded sounds.

2. Related Work

2.1. Classification of Acoustic Scenes

The topic of classification of acoustic scenes using convolutional neural networks has received incresing interest over the last few years, Justin *et al*. [3] and Karol[4] used a convolutional neural networks for classification of acoustic scenes, making use of the public UrbanSound8K dataset, which is also one of the dataset used for study in the present paper. Using different architectures they achieved average test accuracy results of 73% and 79% respectively, the latter using different data augmentation techniques. Venkatesh *et al*. [5] used another two convolutional neural networks: AlexNet and GoogLeNet, with the spectrogram of the audio signals as input features to the networks. Huy *et al*. [6] used recurrent neural networks for the task of acoustic scenes classification and achieved state-of-the-art accuracy results in the LITIS Rouen dataset, confirming the validity of this type of architecture for the acoustic scenes classification task.

2.2. Convolutional Neural Networks

Convolutional neural network (CNN, or ConvNet) is a class of deep neural network, they are made up of neurons that have adjustable weights and biases. Each neuron receives some inputs, performs a elementwise product, adds a bias term and performs a non-linear activation. A loss function is computed and the parameters are updated using gradient descent optimization on this function. However, CNNs present some new characteristics that make them more suitable architectures when working with images as inputs.

A typical ConvNet architecture is shown in Figure 1, which shows the use of two new types of layers: convolutional layers and pooling layers. The depth of the 3D vol-

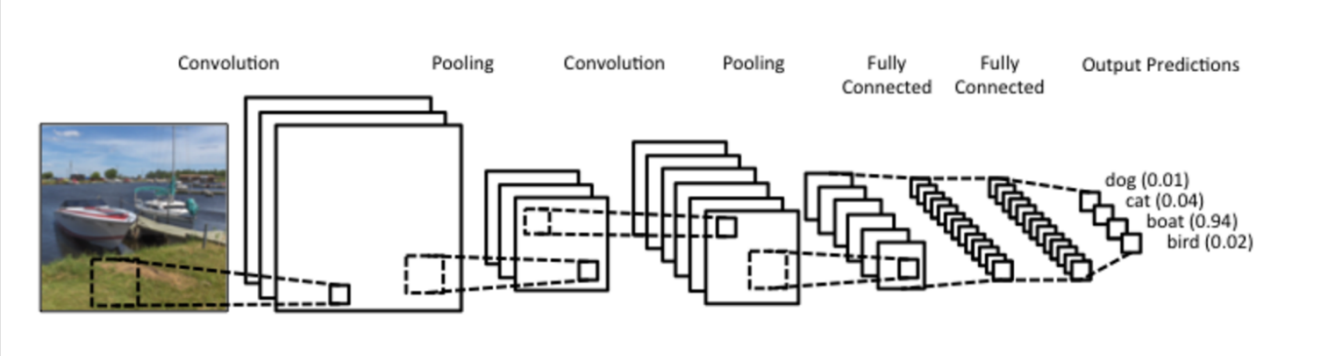

Figure 1. Architecture of a ConvNet having convolutional, pooling and fully connected layers

umes corresponds in this image to the dimension going into the plane.

Convolutional neural networks are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the sparse connectivity and shared-weight[7, 8] architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. Counter intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommendation systems, image classification, image segmentation, medical image analysis, natural language processing, human-computer interfaces, and financial time series, etc.

2.3. Feature Generation

Feature generation is the process of taking raw, unstructured data and defining features, i.e. variables, for potential use in the statistical analysis. This is a really important step in order to provide the machine learning algorithm with the best possible data to perform in the best possible way. An example showing the importance of this step is shown in Figure 2.

Fourier transform. The Fourier transform allows the conversion of signals between the time and the frequency domains. The continuous Fourier transform of a signal $s(t)$ is defined as

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt$$

The discrete counterpart, i.e. the Discrete Fourier Transform (DFT) is defined as

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-j\frac{2\pi}{N}kn}$$

where k refers to the Fourier component number, or frequency bin, and n to the sample number. In general, $S(k)$

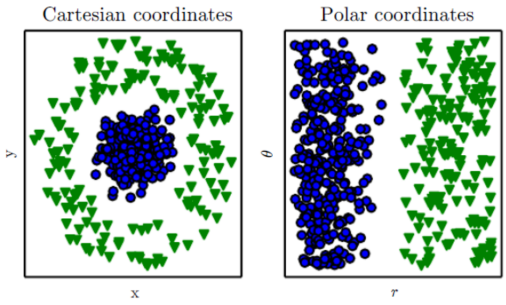

Figure 2. Example of how different representations may turn an impossible task (separating the two categories drawing a straight line) into an easy task for a computer to solve. Picture taken from [7]

is complex valued, and usually the only interesting part is the magnitude of it. This is called a magnitude spectrum $|S(\omega)|$. A power spectrum is calculated by squaring the magnitude spectrum, i.e. $|S(\omega)|^2$. The frequency step Δf of a DFT is calculated as the sampling frequency f_s divided by the number of frequency components N_f in the spectrum.

$$\Delta f = \frac{f_s}{N_f} = \frac{1}{\Delta t \cdot N_f}$$

The number of frequency components N_f is identical to the number of samples of the signal used in the transform. However, the FT returns a double sided spectrum containing negative frequencies which are usually discarded. This means that the single sided spectrum contains half the number of frequency bins than the number of samples in the time domain. The product $\Delta t \cdot N_f$ is equal to the period time T, and hence: An important conclusion from this last formula is that the sampling rate of the signal has no direct influence in the frequency resolution of the corresponding FT. However, the sampling rate does have an influence in the frequency coverage of the FT. Higher sampling frequency allows for higher frequency coverage as established by the

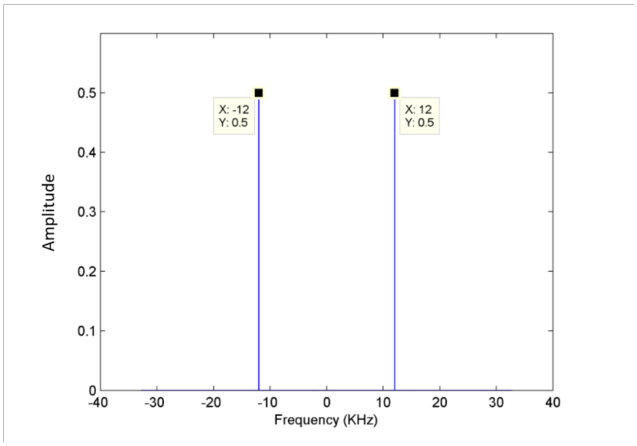

Figure 3. Fourier transform of a sine wave of frequency 12kHz, showing both negative and positive frequencies.

Nyquist-Shannon sampling theorem. In Figure 3 a picture showing the fourier transform of a sine wave can be seen.

Spectrogram and STFT. The DFT assumes the signal to be transformed as being periodic. That is, the end of the signal is connected to the beginning of the signal. This can cause discontinuities at the signal edges, that would show up in the spectrum as non-zero amplitudes at other frequencies than the ones present in the signal. This spread of the amplitude caused by the assumed periodicity of the signal is commonly referred to as $leakage$.

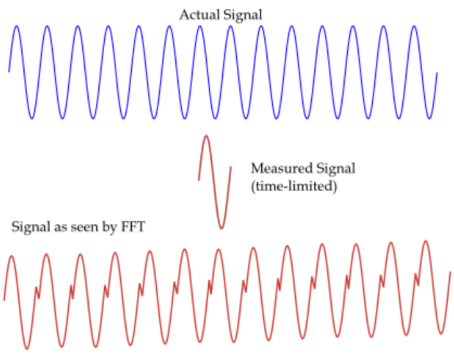

Figure 4. An example of how the DFT periodicity assumption might lead to spectral leakage. Picture taken from [9].

One solution to avoid this leakage is to window the signal before taking the DFT. One commonly used window is the Hanning window, which is defined as:

$$h(n) = \frac{1 - \cos\left(\frac{2\pi n}{N+1}\right)}{2} \quad n = 1, 2, \ldots, N$$

where n is the sample number and N the total number of samples. A Hanning window function is depicted in figure 5.

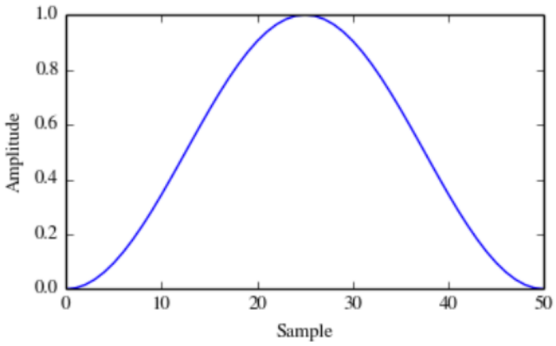

Figure 5. Hanning window function, N = 50

The spectrogram is nothing but several DFT stacked together, which allows to visualize how frequencies change over time. In order to do so, a time window needs to be chosen for which to compute the DFT. This time window is usually chosen to overlap with the following time frame. This is done to preserve all the information of the signal that would otherwise be lost due to two consecutive windowing operations.

Mel-frequency spectrogram. The mel scale is a logarithmic frequency scale that tries to better adapt to human hearing. It was developed by experimenting with the human interpretation of pitch in 1940 's with the sole purpose of describing the human auditory system on a linear scale. The experiment showed that the pitch is linearly perceived in the frequency range $0 - 1000$ Hz. Above 1000 Hz, the scale becomes logarithmic. An approximated formula widely used for mel-scale is shown below:

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f_{Hz}}{700}\right)$$

where f_{mel} is the resulting frequency on the mel-scale measured in mels and f_{Hz} is the frequency measured in Hz. The plot of this function can be seen in Figure 6.

Basically the mel-frequency spectrogram is the regular spectrogram of a signal, mapped onto the mel-frequency scale. In addition to that, the mel-spectrogram is usually grouped into frequency bands. This grouping is obtained by multiplying the discrete spectrogram with a mel-scaled filterbank made up of several overlapping triangular windows. The discrete frequency bins are therefore mapped into a pre- defined number of mel-frequency bands.

Since the frequency range of the signal, determined by its sampling rate, is distributed in uniformly-spaced bands along the mel-scale, this has the consequence that low frequencies are emphasized over high frequencies, for which a more coarse frequency resolution is obtained.

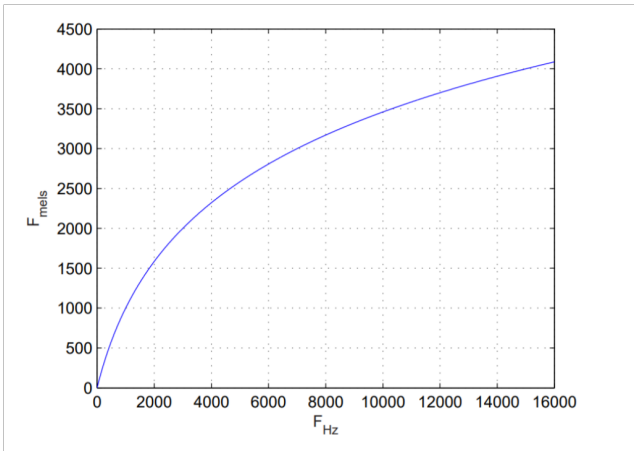

Figure 6. Mel frequency scale vs linear frequency scale

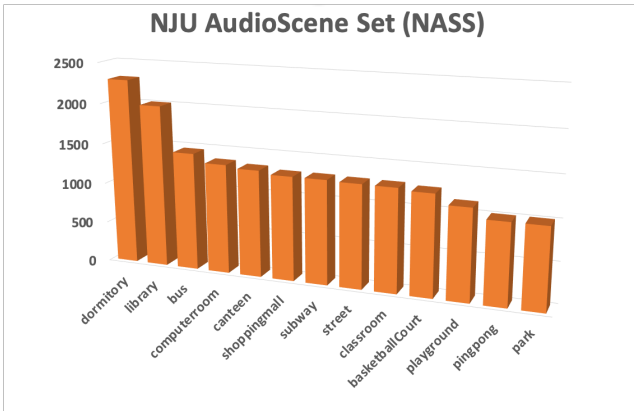

Figure 7. Campus AudioScene Set

3. Acoustic Scenes Classification

3.1. Software

Besides all the general libraries used for data processing and analysis in Python such as Numpy or Matplotlib, four specific libraries were used in this project: The library SoundFile was used for reading and writting audio files. The audio analysis library Librosa [10] was used for the resampling of the audio files as well as for generating the mel-frequency spectrograms that were fed as training data to the network. For the neural network programming part, Facebook's library PyTorch was used. The library Scikit-learn was used for calculating the confusion matrixes.

3.2. Dataset

Both TAU Urban Acoustic Scenes 2019 [11] and Our own dataset [7] are used in this project.

TAU Urban Acoustic Scenes 2019 datasets contain only data from the 10 known acoustic scene classes. The development set contain 40 hours of data, with 14400 segments (144 per city per acoustic scene class). The training/test setup includes segments from Milan only to the test subset. There are 9185 segments in the training set, 4185 in the test set, and additional 1030 segments from Milan.

Our own Campus Acoustic Scenes Sets contain 13 campus acoustic scene classes, with more than 10000 segments (10 seconds per segments). Figure 7 shows the number of segments for each category.

All the recording are in WAV format, and is 24-bit audio in 2 channels, with recording is 10 seconds long.

3.3. Input features to the networks

We take 13 mel-frequency cepstral coefficients over windows of 0.025 second. We augment the feature with first and second order differences, resulting in a 39-dimensional vector.

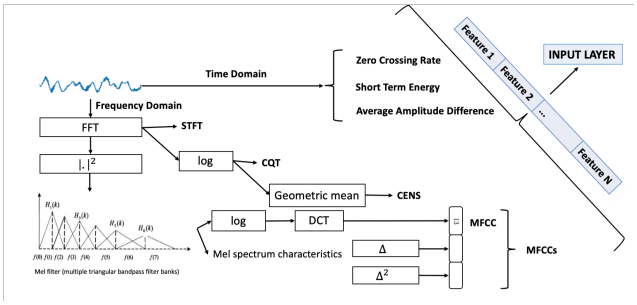

Figure 8. Feature Extraction Process

The mel-spectrogram was chosen since it provides a finer resolution at lower frequencies, where most of the relevant acoustic information is present.

3.4. Network architecture

In Figure 3.4 a sketch of our network architecture used for the classification task can be seen. In the figure, $Conv$ stands for convolutional layer, $Max - Pooling$ stands for maximum pooling layer, BN stands for Batch Normalization, $1 * 1$, $3 * 3$ and 32 , 64 , 128 stand for the filter size and numbers. And $ReLU$ and $Softmax$ are the activation functions of the specified layer.

4. Experiments

4.1. Training procedure and hyperparameters

In Table 1, the final hyperparameter choice used for training of the baseline model is shown.

4.2. Results

In Figure 10 the normalized confusion matrices for the test sets on DCASE can be seen. And in figure 11 the

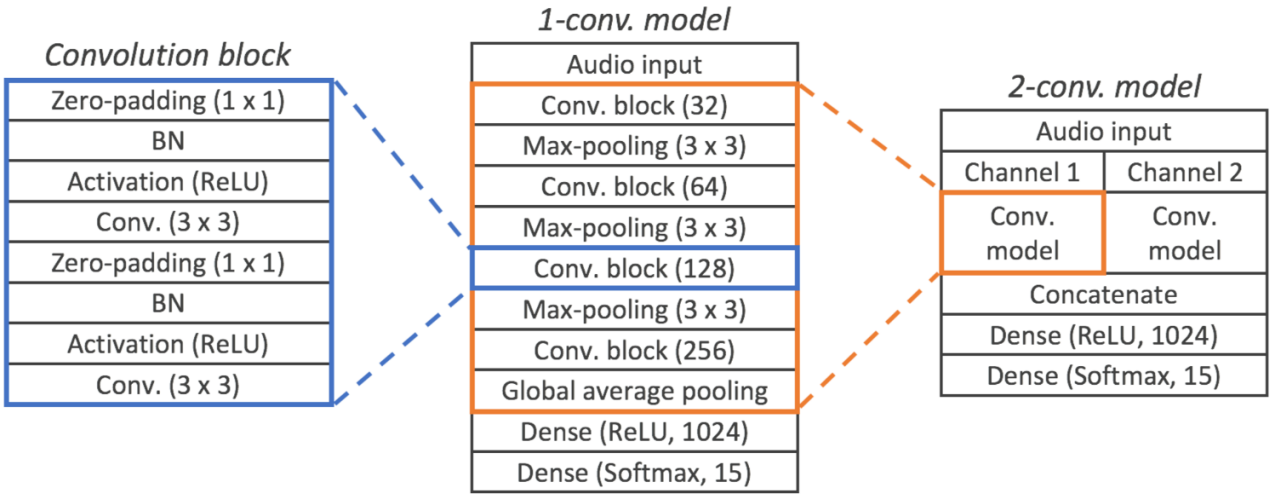

Figure 9. Network Architecture

Hyperparameter	Value
Learning Rate	0.001
Batch Size	64
Epochs	50
Weight decay factor	0.005
Dropout probability	0.5

Table 1. Training hyperparameters

normalized confusion matrices for the test sets on our own NASS data sets can be seen. This gives a clearer picture about the underfitting and overfitting distribution over the different classes.

5. Conclusion

A convolutional neural network architecture was developed for the single-label multi-class classification task using the DCASE and our own acoustic scenes data sets, achieving an overall prediction accuracy of 86%.

In this research, we learned how to use python to exact classical features, e.g, MFCCs, from wav files and we try to build a convolutional neural network using both tensorflow and pytorch frameworks. Although the finally accuracy is not better than the state-of-the-art, we still contribute to a campus acoustic scenes data sets (NASS) and a patent (ID: 201911152466.2) which shows a way to support automatic equalizer settings for music player based on our model.

Finally, our research was regarded as a national-level project and completed as an excellent-level project (ID: G201910284072).

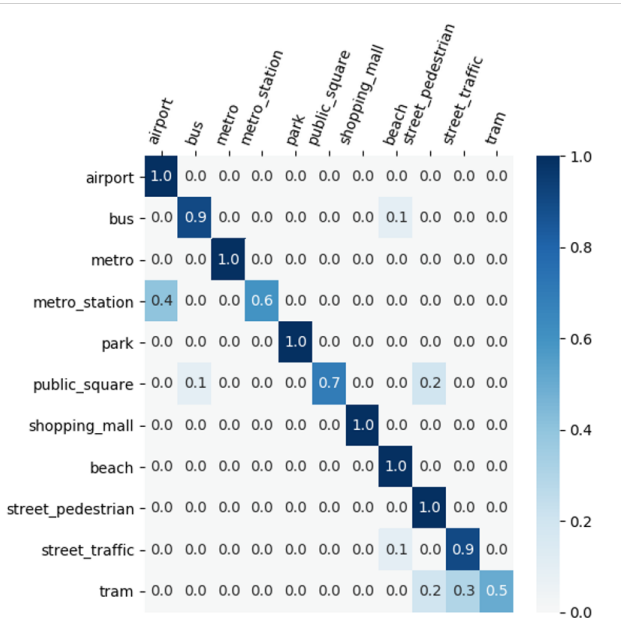

Figure 10. Results on DCASE data sets

6. Acknowledgement

This work is conducted in collaboration with the Team of some of my friends. We would like to thank Andrew Ng and Hung-yi Lee for teaching us knowledge of deep learning. We are also grateful for the help from Prof. Li Zhang, who brings us to the sea of python and artificial intelligence programming.

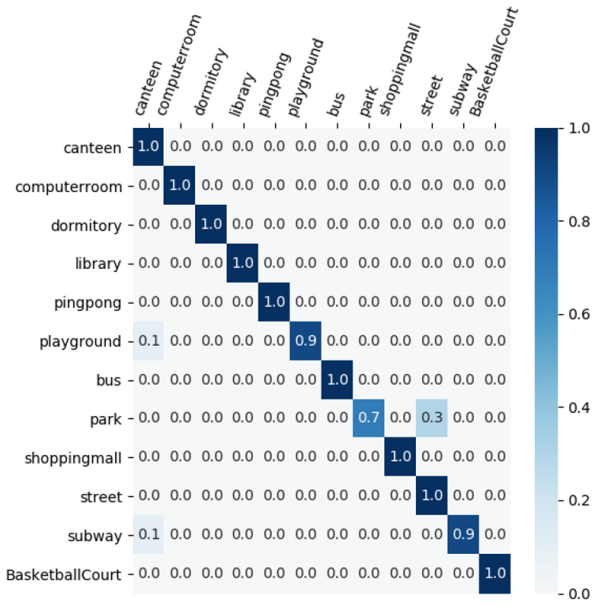

Figure 11. Results on NASS data sets

References

[1] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 1

[2] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 1041–1044, New York, NY, USA, 2014. Association for Computing Machinery. 1

[3] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, Mar 2017. 1

[4] Karol J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2015. 1

[5] Venkatesh Boddapati, Andrej Petef, Jim Rasmusson, and Lars Lundberg. Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, 112:2048–2056, 12 2017. 1

[6] Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur, and Alfred Mertins. Audio scene classification with deep recurrent neural networks, 2017. 1

[7] Aaron Courville Ian Goodfellow, Yoshua Bengio. Deep learning. www.deeplearningbook.org. MIT Press 2016. 2

[8] Convolutional neural networks for visual recognition. http://cs231n.github.io/convolutional-networks/. Accessed: 2018-06-01. 2

[9] John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing (4th Edition)*. Prentice-Hall, Inc., USA, 2006. 3

[10] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt Mcvicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python, 01 2015. 4

[11] Dcase 2019: Acoustic scene classification. http://dcase.community/challenge2019/task-acoustic-scene-classification. 2019. 4