

Urban Sound Classification

Christian Gunther, Kevin Le, Mike Ranis, Derar Durubeh

Abstract—Machine learning applications for image classification and recognition is becoming the benchmark in artificial intelligence. Popular methods and tools developed for image related applications are soaring in both quantity and quality. In this paper we seek to examine the feasibility of several machine learning tools that were developed initially for image related applications and seek to adapt them to classify an image representation of sound. We represent sound as an image of their spectrograms, the merits of this representation is that we have both a temporal dimension of the data and a spatial dimension of our data that of the frequency band. In this way we can examine several methods that are best adapted to sequential and spatial data. The resulting image files produced from the sound files are thus seen to be smooth and characteristic of the environment that produced them, thereby constituting a good candidate to apply image classification algorithms on them to learn relevant features and to classify them.

I. INTRODUCTION

The potential benefits of robust sound classification are immense, from surveillance to house monitoring, it is necessary to be able to discern sounds like a baby crying, air-conditioning or glass breaking for smart home monitoring systems that could soon be essential to have. Every day sounds that are characteristic of the background of everyday living is the focus in this paper, those sounds are more prone to noise and are more random in both the periodic and aperiodic sense than specialized sound like speech and music for which there already are state of the art techniques for recognition and classification. Environmental sounds are thereby any sound that is not speech or music and that is produced from environments common in day to day life. The prime focus of our effort in this paper is to represent those sounds and to classify them. During the past few years, many attempts to recognize environmental sounds have been made. Presently, there is an increasing focus on classifying environmental sounds using deep learning techniques [9] the improvements in the field of image classification in recent years are leading researchers to start using images when classifying sounds. Sound data representation in the machine learning literature that produced the best results has usually been through either auditory images [6], spectrogram image features [8] and spectrogram-derived subband power distribution [9]. For the representation of our sound data, the best way to do this per the state of the art results is to represent the sound files using their power spectrum [5], more commonly known as the mel-frequency cepstrum, this transformation provides a heat map like representation of the signal breaking down the signal based on how much of its total power is in each frequency as time progresses, with darker colors (higher values) indicating more relative power

at that frequency. For all our proposed solutions we will thus be using the mel spectrograms as the input to the machine learning algorithm. Having defined the inputs, the objective is to classify the testing dataset into the appropriate classes, and so the output is simply the class label of the data input. The methods we will be contrasting in this paper will be the unsupervised learning algorithm of nearest neighbors and the supervised learning algorithms of artificial neural networks and convolutional neural networks.

II. RELATED WORKS

Machine learning literature on sound classification generally and environmental sound classification specifically range in scope and methods from data representation, feature extraction, and algorithms used. Attempts at this problem using classical machine learning algorithms like support vector machines(SVM) for the classification were performed in Wang et al [10] where they used for their data representation a gabor based scale frequency map and used principal component analysis and linear discriminant analysis to extract features. Comparison of different classical machine learning methods were outlined in Lu et al [11] where they concluded that SVM's outperformed both K-nearest neighbors and Gaussian mixture models. While in both those studies high accuracies were reported, we suggest that using such feature extraction routines such as principal component analysis and other dimensionality reducing procedures would reduce the robustness of the classification step to noise. This is specially important in environmental sound files since they typically arise in very noise environments where there will be other countless sources of sound producing agents, thus it becomes paramount to ensure the procedure of classification we are using is not rigid or inflexible when noise is introduced to the data. This is also true for the kind of machine learning models we intend to use, using such algorithms as SVM's where essentially a hyperplane boundary is demarcating the classes, a slight modification via noise to the classifier features could dislocate the data file to a different class. Other noteworthy methods of feature extraction are presented by Bisot [13] in which they learn features from time-frequency images in an unsupervised manner. The images are decomposed using matrix factorization methods to build a dictionary and the projection coefficients are used as features for classification. Modern feature extraction methods audio processing comprises of using 1D CNNs that learn acoustic models directly from audio waveforms are popular due to the ability of these networks to take advantage of the signals fine time structure[14]. In Piczak [11] they worked on the same dataset and problem as we do in this paper, the proposed

solution in that paper consisted of spectrogram image features in conjunction with convolutional neural networks, it was hypothesized in that paper that due to the fact that general sounds are not precisely localized in the time-frequency spectrogram, but may preserve strong local relationships, means that the global convolution and subsampling approach inherent to the CNN has advantages. The results presented in that paper convey that this is indeed the best way to handle these kind of features for this task. An improvement on the results in the paper by Piczak was achieved in [15], in which the proposed solution consisted of an end-to-end 1D CNN for environmental sound classification that learns the representation directly from the audio waveforms instead of from 2D representations, after that several convolutional layers are used to learn low-level and high-level representations. The highest level of representation is then used for classifying the input signal by means of three fully connected layers, this implementation in [15] provided the best results thus far, outperforming previous implementations of sound classification by CNN's. A noteworthy study to mention concludes that for environmental sound classification, the ratio (classification performance)/(computational cost) is more favourable for deep neural networks compared to both Gaussian classical machine learning algorithms such as SVM's and GMM's. Since this ratio is particularly important on mobile devices with limited processing and battery capacity, evaluating deep neural networks for sound classification on mobile devices is very relevant [12].

III. DATASET AND FEATURES

A. Dataset

The dataset is called UrbanSound8K [1]. This data set contains 8732 labeled sound excerpts ($t=4s$). The sounds are from 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, street music. The audio excerpts are .wav files. The data set is already shuffled and separated into 10 folds for a 10 fold cross validation and the authors of the data recommend not reshuffling the data during training due to the distribution of the sound classes within each fold.

B. Features

We used the Librosa audio package in python to extract features from the audio files in our dataset. The following are the features that we extracted: (1) Mel-Frequency Cepstral Coefficients (MFCC): Coefficients derived from a cepstral representation of the audio clip, (2) Chromagram: Pitch class profiles. They capture harmonic and melodic characteristics within the music, (3) Mel-scaled spectrogram: Psychoacoustic scales that capture the distances from low to high scale frequency, (4) Contrast: Difference between parts of a sound or different instrument sounds, (5) Spectral Contrast: Represents the strength of spectral peaks and valleys in each a sub-band as contrast distribution, (6) Tonnetz: Representation of tonal space. Visualizations of the .wav file formats and the mel-scaled spectrograms for each class in the data set can be seen in figures 1 and 2.

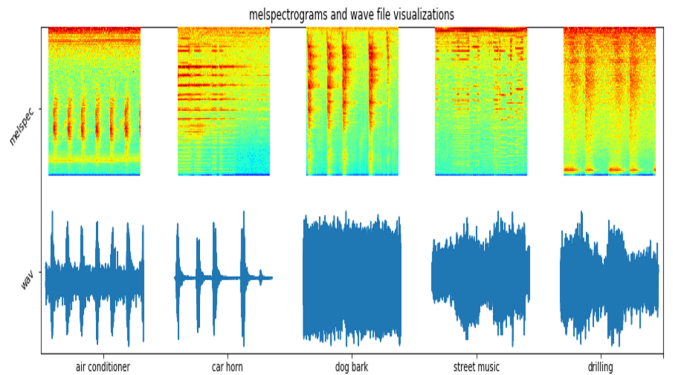


Fig. 1. Mel-Spectrograms and .wav file plots of 5 of the urban sound classes

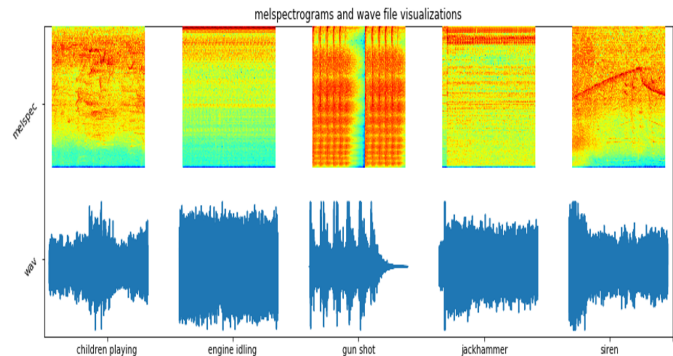


Fig. 2. Mel-Spectrograms and .wav file plots of the other 5 classes of the urban sounds

IV. METHODOLOGY

To solve the urban sound classification problem, we use several recognition approaches from the literature. The major consideration in their selection was their ability to exploit the characteristics of this problem. We considered models that range from a simple interpretation to more complexity.

A. K-Nearest Neighbor

The K-Nearest Neighbor algorithm (KNN) is a non-parametric method that is the simplest of machine learning algorithms. The algorithm classifies the input by assigning a class membership of the most common among its k -nearest neighbors, where k is a positive integer. KNN is also used for regression, but for this task, we will omit it. In a statistical setting, the KNN can be best understood that when given a dataset, containing pairs (X_i, Y_i) of data points X_i and class labels Y_i , this is reordered by the norm and point x , such that $\|X_{(1)} - x\| \leq \dots \leq \|X_{(n)} - x\|$. An example of the class membership assignment can be followed in Figure ???. Of the many distance metrics used for KNN, the Manhattan distance was used based off results of a grid search cross validation of the model. The distance can be computed by the following:

$$d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^n |x_i - y_i| \quad (1)$$

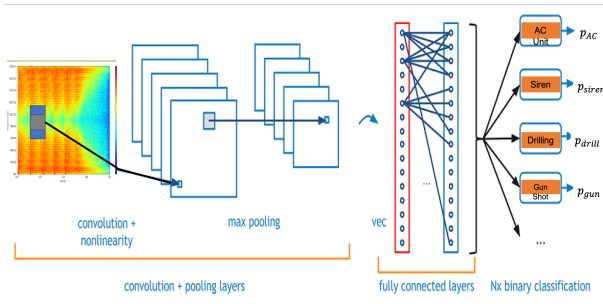


Fig. 3. Feature extracted audio wave file being forward propagated through a standard convolutional neural network to output a posterior probability

Along with this grid search, our model used 11 nearest neighbors for the k-nearest neighbor classification.

B. Deep Neural Network

A deep neural network, also known as a multi-layer perceptron, is a hierarchy of weighted connections between artificial neurons. Generally the architecture can be best understood by three levels: the input, the hidden layers, and output. Learning representations of data automatically is possible due to these multiple levels of abstractions. The main benefactors of this learning accomplishment are due to two things. First, the neurons in the hidden layer act as classifiers, that are trained to filter and detect specific features by receiving weighted inputs, transforming it with the activation function and passing it to the outgoing connection. Our network consisted of 4 hidden layers with ReLU activation functions and applied dropout. For layers 1 through 3, there were 400, 500, and 400 neurons for each respective layer. The loss taken after the output of this network was cross-entropy loss for the reason of the task being classification. Following loss computation, the network backpropagates the measured error to learn the features that solve the problem. Using the chain rule of calculus, the algorithm calculates the derivative of the error with respect to each weight in each layer.

$$\begin{aligned} \frac{\delta E^n}{\delta w_{jk}} &= -\delta_k z_j \\ \delta_k &= \frac{\delta E}{\delta a_k}, z_i = \frac{\delta E^n}{\delta w_{ij}} = \delta_j x_i \\ \delta_j &= g'(a_j) \sum_k w_{jk} \delta_k \end{aligned}$$

where, w_{jk} is the weight of the output layer and $g'(a_j)$ is the activation layer of the hidden layer. Given the network topology and learning procedure, we are able to learn the internal representations in the service of the task of classifying sound files.

C. Convolutional Neural Network

This is a standard convolutional neural network (CNN), which models the class posterior probability distribution $P(y|x)$ with a feature extractor composed of multiple layers of convolutions, ReLU nonlinearity, and pooling, and a

softmax layer, composed of a linear layer and the softmax function. An example of such an architecture can be visualized in Figure 3. This architecture is similar to the deep neural network in that it is a hierarchy of interconnected neurons, but differs mainly by its usage of convolutional layers for processing imaged-related inputs. For this task, our topology consisted of having 3 layers in total, where our convolutional layer had a filter size of 5 and depth of 32, followed by a ReLU nonlinearity and maxpooling with same zero-padding. This is followed by a fully connected layer and softmax layer. Similar to the deep neural network, a cross-entropy loss is used to measure the CNN's error, which can be understood as follows:

$$E(w) = - \sum_{n=1}^N \{t^n \ln(y^n) + (1-t^n) \ln(1-y^n)\} \quad (2)$$

where t is the class label and y is the predicted label. In conjunction, an Adam optimizer is used when tuning the weights during backpropagation.

V. EXPERIMENTAL SETTING

For our experiments, we apply the above methods to the UrbanSound8K dataset. The training parameters and hardware used are also discussed here.

a) *Training Procedure:* When training KNN, the simpler models to train, we performed a grid search cross validation across the number of nearest neighbors, weighted assignments, and distance metrics, as mentioned previously in the methodology section. This consists of our training procedure for the KNN. As for the DNN and CNN, the procedure consists of tuning the mini-batch size, learning rate, regularization value, and dropout probability. To make a valid comparison of the two architectures' performances, our hyperparameters are consistent for both networks and was determined through several combinations of hyperparameter tuning. From experimentation, we designed our training procedure to use a mini-batch size of 97, learning rate of 0.0001, a regularization of 0.01 and dropout probability of 0.2.

b) *Evaluation Metrics:* Our primary evaluation metric for our classification models was accuracy. Accuracy is defined as the number of correct predictions over the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where

$$TP = \# \text{ of True Positives}$$

$$FP = \# \text{ of False Positives}$$

$$TN = \# \text{ of True Negatives}$$

$$FN = \# \text{ of False Negatives}$$

We also looked at the AUC (Area Under the Curve) of the ROC (Receiver Operating Characteristics) curves for

the different models. The ROC curve plots the TPR (True Positive Rate) against the FPR (False Positive Rate).

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{TN + FP}$$

Finally, we created confusion matrices to visualize which classes were being correctly and incorrectly classified. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class

c) Hardware Utilized: For all of our reported experiments, we utilized the Nvidia GTX Titan XP GPU units. Based off a Pascal GPU architecture, the Titan XP is able to provide a reported 11.4 Gbps memory speed, 1582 MHz boost clock, and 12 GB G5X frame buffer. Models were trained using a single GPU unit, rather than multiple GPUs, due to resource availability.

VI. RESULTS

Below we discuss the results of our experiments.

Model	Accuracy	AUC
CNN	73.4%	93.1%
DNN	68.3%	90.6%
K-NN	55.6%	84.4%

TABLE I

MODEL PERFORMANCE ON URBANSOUND8K DATASET



Fig. 4. Loss

Intended to investigate the performance of sound recognition within urban settings, our experimental procedure is focused on comparing several simple and complex model performances, based off their accuracy and AUC. This helps us understand the difficulty level of the task, given the model complexity, implying how realistic the application of sound recognition could be when the model is deployed. These results were conducted on folds 8 through 10 of the cross validation folds of the dataset, after being trained on folds 1 through 7. Each model performance is fairly compared

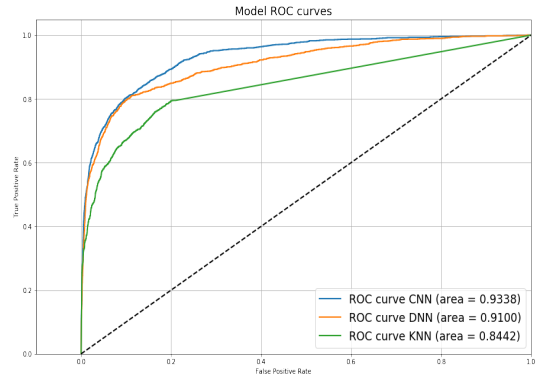


Fig. 5. ROC Curves

with one another, as the training parameters and hardware are consistent for each experiment.

To begin with, from Table VI, it is easily noticeable that the CNN outperformed the DNN and KNN in both accuracy and AUC, achieving a 73.4% and 93.1% respectively. This might imply that the feature extraction of the dataset effectively corresponds with the use cases of the CNN, as we would expect, since the concatenation of the features represent the visual form of an image. Clearly, patterns within the audio file can be interpretative from a vision perspective, since the CNN is able to perform well. In retrospect to these neural networks, it is not surprising that the KNN would underperform the other models by approximately 11% in accuracy. This only separates the difference in model complexity by a farther degree. However, it is worth noting from Figure 5 that the KNN was able to achieve a decent AUC of 84.4%.

The confusion matrices compare our models' predictions with the actual label of the sound. Both the Convolutional Neural Network and the Deep Neural Network had similar confusion matrices. One can analyze the confusion matrix and see which classes get classified incorrectly as which other classes. For example, the CNN incorrectly classifies air conditioners as children playing, jackhammers, and drilling. These incorrect classifications can provide insight into the model as well as the data. These different sounds probably have similar structures. Another explanation could be that air conditioners are a background sound, so other sounds may be present in the clip. Looking at the K-Nearest Neighbor confusion matrix, it has a lot more incorrectly classified classes. Something interesting to note is that it predicted almost all classes as children playing at least some of the time.

VII. CONCLUSION AND FUTURE WORKS

Of the methods implemented in our project, the CNN performed the best, the DNN was the second best, and the KNN came in last place. The methods that we tested performed in a manner that we expected; we did not expect the KNN to outperform either the DNN or the CNN. However, we did see higher accuracy results in some of the research that we came across for Urban Sound classifications [4].

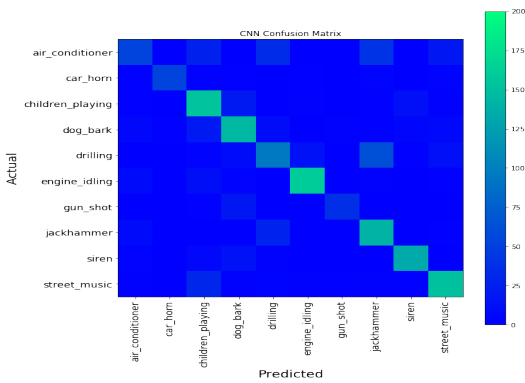


Fig. 6. CNN Confusion Matrix

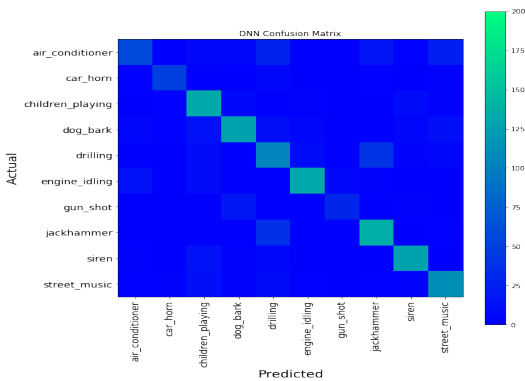


Fig. 7. DNN Confusion Matrix

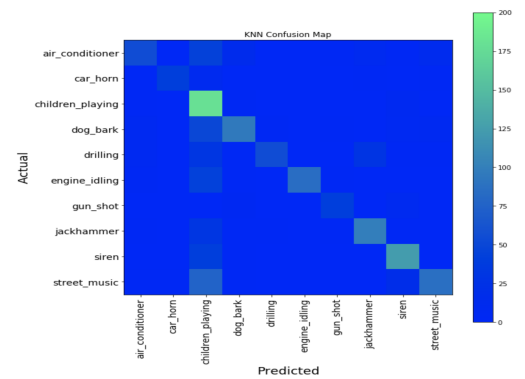


Fig. 8. KNN Confusion Matrix

Overall, one of the biggest bottlenecks of our project was the time it took to extract the features for use with our neural networks, and if we had more time, perhaps a faster method of feature extraction could have been discovered. Additionally, we would have spent more time figuring out how to get our extracted features properly into a well-known CNN, such as VGG-16 or AlexNet and accessed the performance of our model using those CNNs (We came close to achieving this, but the training was super slow due to the nature of the feature extractions and the server pods kept timing out). Furthermore, we would have also liked to explore implementing an LSTM network for the sound

classification tasks on this dataset.

CONTRIBUTIONS

a) Kevin: My contributions to this project consists of organizing and refactoring the codebase. I produced results for the KNN model and assisted with developing the DNN and CNN. Prior to these accomplishments, I worked with experimentation of several feature extractions, loading the data, and writing the codebase in PyTorch (before switching to Tensorflow). As for writing, I helped with writing the methodology section and experiments.

b) Christian: I worked on loading the dataset, exploring the data, and extracting features from it in TensorFlow. Additionally, I helped debug the DNN and CNN as well as plotted the results. For the report, I wrote parts of the Experiments and Results sections.

c) Mike: I worked on a pytorch version of feature extraction, dataloading, and VGG-16 CNN implementation, but it failed to function fully due to timing constraints and slow feature extraction (the pod would time out). I created the graphs for the melspectrogram and .wav file plots. For the report I wrote the dataset and features section, as well as the conclusion.

d) Derar: Meetings, progress reports and experimentation with different methods and procedures. Mel-Spectrogram feature extraction code and implementation. CRNN (convolutional + LSTM) implementation in pytorch. Report first three parts. Final report proof reading.

Attached is our repository for reference. Click [here](#)

REFERENCES

- [1] Justin Salamon, Christopher Jacoby, Juan Bello, "UrbanSound8K dataset." [Online]. Available: <https://urbansounddataset.weebly.com/urbansound8k.html>10foldCV.
- [2] Benjamin Doran, "Urban-Sound-Classification." [Online]. Available: <https://github.com/BenjaminDoran/Urban-Sound-Classification>.
- [3] Aaqib Saeed, "Urban Sound Classification, Part 1." [Online]. Available: <http://aqibsaeed.github.io/2016-09-03-urban-sound-classification-part-1/?fbclid=IwAR013w3zKLA5qL0F-jUztliqP2aW9yyvIS2-MIHLMP3S1fX21PERAlw>.
- [4] Venkatesh Boddapatia, Andrej Petefb, Jim Rasmussonb, Lars Lundberg, "Classifying environmental sounds using image recognition networks." [Procedia Computer Science]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050917316599>.
- [5] Ian McLoughlin, Zhang H.-M., Xie Z.-P., Song Y, and Xiao W, Robust sound event classification using deep neural networks, Audio, Speech, and Language Processing, IEEE Transactions on, vol. PP, no. 99, 2015.
- [6] Thomas C Walters, Auditory-based processing of communication sounds, Ph.D. thesis, University of Cambridge, 2011.
- [7] Jonathan Dennis, Huy Dat Tran, and Eng Siong Chng, Image feature representation of the subband power distribution for robust sound event classification, Audio, Speech, and Language Processing, IEEE Transactions on, vol. 21, no. 2, pp. 367377, 2013.
- [8] Jonathan Dennis, Huy Dat Tran, and Haizhou Li, Spectrogram image feature for sound event classification in mismatched conditions, Signal Processing Letters, IEEE, vol. 18, no. 2, pp. 130133, 2011.
- [9] Classifying environmental sounds using image recognition networks Venkatesh Boddapatia, Andrej Petefb, Jim Rasmussonb, Lars Lundberga, OF* aDepartment of Computer Science and Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden Sony Mobile Communications AB, Mobilvgen, 221 88 Lund, Sweden.
- [10] J.-C. Wang, C.-H. Lin, B.-W. Chen, and M.-K. Tsai, Gabor-Based Nonuniform Scale-Frequency Map for Environmental Sound Classification in Home Automation, in IEEE Transactions on Automation Science and Engineering, vol. 11, no. 2, pp. 607613, Apr. 2014

- [11] L. Lu, H.-J. Zhang, and S. Z. Li, Content-based audio classification and segmentation by using support vector machines, in *Multimedia Systems*, vol. 8, no. 6, pp. 482-492, Apr. 2003
- [12] S. Sigita, A. M. Stark, S. Krstulovic, M. D. Plumbley, Automatic environmental sound recognition: performance versus computational cost, in *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 11/2016, Vol. 24, No. 11
- [13] Bisot, V., Serizel, R., Essid, S., Richard, G. (2016). Acoustic scene classification with matrix factorization for unsupervised feature learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6445-6449).
- [14] Hoshen, Y., Weiss, R. J., Wilson, K. W. (2015). Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4624-4628).
- [15] End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network Sajjad Abdoli1 , Patrick Cardinal, Alessandro Lameiras Koerich Department of Software and IT Engineering, Ecole de Technologie Supérieure, Université du Québec, H3C 1K, Montreal, QC, Canada