Audio Recognition using Mel Spectrograms and Convolution Neural Networks

Boyang Zhang   Jared Leitner   Sam Thornton
Dept. of Electrical and Computer Engineering, University of California, San Diego
Email: {boz083, jjleitne, sjthornt}@ucsd.edu

Abstract — Automatic sound recognition has received heightened research interest in recent years due to its many potential applications. These include automatic labeling of video/audio content and real-time sound detection for robotics. While image classification is a heavily researched topic, sound identification is less mature. In this study, we take advantage of the robust machine learning techniques developed for image classification and apply them on the sound recognition problem. Raw audio data from the Freesound Dataset (FSD) provided by Kaggle is first converted to a spectrogram representation in order to apply these image classification techniques. We test and compare two approaches using deep convolutional neural networks (CNNs): 1.) Our own CNN architecture 2.) Transfer learning using the pre-trained VVG19 network. Using our self-developed architecture, we achieve a label-weighted label-ranking average precision (LWLARP) score and top-5 accuracy of 0.813 and 88.9%, respectively, when predicting 80 sound classes.

***Index Terms*— Machine Learning; Audio Recognition; Mel Spectrogram; CNNs**

I. Introduction

Sound is one of the five primary senses humans use to understand the world around them. Many of today's autonomous systems are predominantly vision based [1], and do not take advantage of the additional environment information provided from audio. Developing intelligence that is able to process both image and audio data concurrently would provide autonomous systems a deeper understanding of their environment and allow them to interact with their surroundings in a more meaningful way.

Sound is defined as a vibration that propagates through a medium (air, water, etc.) as an audible compression wave [2]. These physical vibrations can be converted to an electrical signal using a transducer such as a microphone. This signal is then digitized in order to carry out various preprocessing and machine learning techniques. Sound waves are described by physical quantities including frequency, amplitude, and direction. We look to extract these properties and utilize them for recognizing certain sounds.

Audio signals are inherently one-dimensional (amplitude over time). We look to transform these signals into a more descriptive representation that better illustrates the previously mentioned quantities. In this study, we propose the use of the Mel spectrogram, a transformation that details the frequency composition of the signal over time [3]. Since this results in an image representation of the audio signal, the Mel spectrogram is the input to our machine learning models. This allows us to make use of well-researched image classification techniques. The convolution neural network (CNN) is a powerful deep learning model that can learn a feature hierarchy for images. Since we are interested in predicting 80 different categories of sound, our model must be able to learn a high number of features in order to recognize specific sounds. In this study, we focus on two approaches to sound recognition using CNNs. In the first, we construct our own CNN architecture and fully train all the layers using our dataset. Second, we make use of transfer learning by utilizing the pre-trained VGG19 network, and only train the last few layers using our data.

The rest of the paper is organized as follows. In section II, we compare our proposed approach to previous work conducted on this topic. Section III details the dataset and our preprocessing methods. In section IV, our models used for sound recognition are described in detail. Section V shows the results for both of the two approaches. Finally, we conclude the paper in section VI.

II. Related Work

Previous work focused on audio recognition is described in [4-8]. The study conducted in [4] uses more traditional methods like k-NN and naïve Bayes to perform audio classification, while [5] shows how modern CNNs can be used to classify audio. The paper focused on CNNs obtained higher accuracy for classification. The biggest weakness from [4] and [5] is their lack of preprocessing, as their methods primarily work directly with the raw audio data. Based on these results, we determined additional preprocessing was necessary to accurately classify audio using CNNs.

It is well-known that CNNs have high performance for image classification. In [6], the authors showed how popular CNN architectures, such as AlexNet, VGG, Inception, and ResNet, performed when used for audio classification. Their approach involved decomposing the audio time series with a short-time Fourier transform to create a spectrogram which was used as an input to the CNN. The problem we faced with many of these models is that they are large and consist of many trainable parameters. This approach worked for [6] since their training dataset had 70,000,000 samples. Since our dataset is on the order of thousands of samples, it seemed unlikely that these large networks could be trained with our limited data. One approach to get around this is transfer learning. This involves using a pre-trained network, freezing most of the layer weights, and only retraining the last few layers on our audio training data. This is one of the approaches we pursued for this project. If we wanted to fully train a network with our data, a network such as the one discussed in [7] would be more appropriate. This network was designed for a dataset with 8732 entries and 10 classes, as opposed to the 70,000,000 entries and 30,871 classes in [6]. This inspired our second approach to classifying our dataset, which involved constructing a smaller architecture.

[8] tries to interpret and explain how CNNs can classify audio signals. Using both spectrogram and raw audio inputs, CNNs are trained and a layer-wise relevance propagation (LRP) is used to see how the models select features and make decisions. They showed the unique regions on the input signal that correspond most strongly to that particular output label. The results of the paper also confirm that spectrogram inputs lead to higher accuracy over raw audio inputs.

The results of these previous works influenced our proposed method to audio classification. For preprocessing, the Mel spectrogram is used to represent the audio signal in a more descriptive manner. Transfer learning and a smaller CNN architecture are implemented to accurately classify our audio data. The following section details our raw audio preprocessing and the Mel spectrogram.

III. Dataset & Features

The popular machine learning and data science website Kaggle has provided two datasets containing labeled audio clips as part of an ongoing competition. The first is the Freesound Dataset (FSD) [9], which is a collection of crowdsourced annotations of 297,144 audio clips. A subset (4,970) of these audio clips comprise the competition's curated dataset, which have been cleaned and validated to remove label noise. The second dataset is the Yahoo Flickr Creative Commons 100M dataset (YFCC) [10]. The YFCC dataset contains 99,206,564 photos and 793,436 videos. The soundtracks of a subset (19,800) of YFCC videos comprise the competition's noisy dataset. All audio data were sampled at 44.1kHz and range from 0.2 - 30 s in length.

In this study, we chose to focus on the curated dataset in order to determine whether our proposed method could work with clean data. This resulted in 4,970 labeled audio clips in which 80% were reserved for training and the remaining 20% for testing. There is a total of 80 sound categories, corresponding to everyday sounds such as applause, a dog bark, a motorcycle, and a raindrop. Using this subset, each raw audio waveform was first processed by trimming the silent sections of the clip and then either further trimmed or zero-padded to equal a length of 2 seconds. This is necessary because our models require a static input dimension, and 2 seconds is the average length of the audio data. Next, each processed clip was transformed into its Mel spectrogram representation. A spectrogram is a visual depiction of a signal's frequency composition over time. The Mel scale provides a linear scale for the human auditory system, and is related to Hertz by the following formula, where m represents Mels and f represents Hertz:

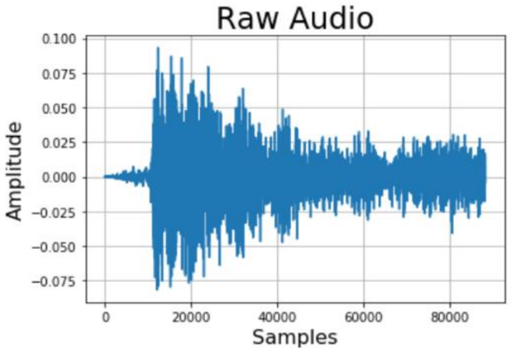

Figure 1: Raw audio signal

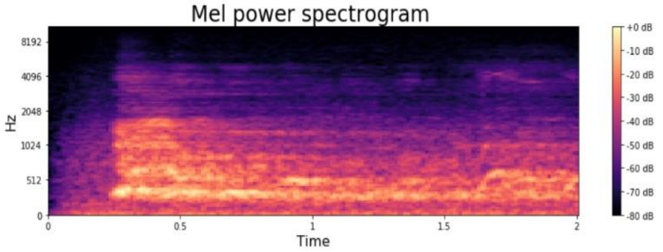

Figure 2: Corresponding Mel spectrogram

$$m = 2595\, log_{10}\left(1 + \frac{f}{700}\right)$$

The Mel spectrogram is used to provide our models with sound information similar to what a human would perceive. The raw audio waveforms are passed through filter banks to obtain the Mel spectrogram. After this process, each sample has a shape of 128 x 128, indicating 128 filter banks used and 128 time steps per clip. Figures 1 and 2 display a raw audio clip and the corresponding Mel frequency representation, respectively. Our models look to learn features from this representation, and their architectures are described next.

IV. Proposed Method

In this section we present our two approaches to the audio classification problem. The different model architectures and components are discussed in detail.

A. Self-Developed CNN

CNN has been very successful in various tasks due to its unique layers. It is usually composed of convolution layers and pooling layers. A brief description of these layers is shown below. We choose CNN mainly because of its ability to analyze spatial invariant features and using a relatively small number of parameters.

The convolution layer uses filters to translate over the input and then takes the inner product before adding the bias. Each filter has its own set of weights and bias. The weights and bias are the only parameters to train. Each layer can have multiple filters to learn different features. This gives CNN the benefits of relatively small number of parameters to learn and being able to learn spatial invariant features.

The pooling layer is used to reduce the dimensionality of the subsequent layers. The commonly used pooling techniques are maxpooling and average pooling, where maxpooling takes the maximum value of the pooling window and average pooling takes the average value.

For hidden layers in the model, we use Relu activation function. It has the form $Relu(x) = max(0, x)$. The non-linearity of the expression eliminates the gradient vanishing problem. For the output layer, we use softmax activation function for classification. The function squashes a vector into range (0,1) that all the element adds up to 1. This can be interpreted as the probabilities for each element.

$$Softmax(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

Since it is a multi-label classification task, we use binary cross-entropy loss for training the model. Binary cross-entropy loss calculates the loss for each class for a given sample independently. For each label, it evaluates the loss using the current class vs the rest. The expression is shown below:

$$BCE = \begin{cases} -log(f(s_1)) & if\ it\ belongs\ to\ current\ class \\ -log(1 - f(s_1)) & else \end{cases}$$

where s1 is the current score.

We use Adam optimizer for stochastic gradient descent. Adam optimizer uses variable learning rate so that the step size is invariant to the magnitude of the gradient, which is a typical problem encountered in traditional stochastic gradient descent.

A visualization of the model is shown below:

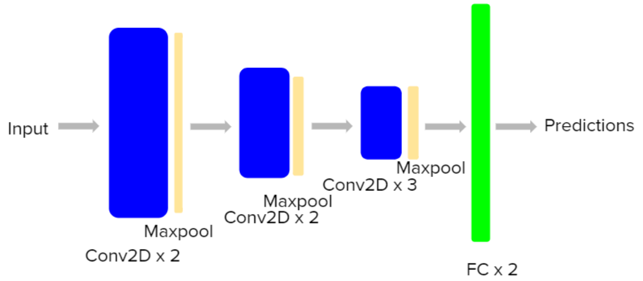

Figure 2: Self-developed CNN architecture

B. Transfer Learning

Transfer learning is a machine learning technique where a model is constructed and trained with a set of data then repurposed for a different task. It has the benefit of shortening training time and improve performance.

VGG19 is a large-scale CNN based model trained with Imagenet data. It is very successful for image classification. applying VGG19 model for transfer learning in other image classification tasks, it is common to only retrain the fully connected layers. However, for our project, we are repurposing the structure for audio classification. We expect the high-level features learned in the last few convolution layers will be different between audio and image data. Thus, we will be retraining the last convolution block (containing 4 convolution layers) and the fully connected layers.

V. Results

For this project, the main metric we use to evaluate our models' performance is label-weighted label ranking average precision (LWLRAP). This measures the average precision of retrieving a ranked list of relevant labels for each test clip (i.e., the system ranks all the available labels, then the precisions of the ranked lists down to each true label are averaged). The novel "label-weighted" part means that the overall score is the average over all the labels in the test set, where each label receives equal weight (by contrast, plain lrap gives each test item equal weight, thereby discounting the contribution of individual labels when they appear on the same item as multiple other labels). [11]

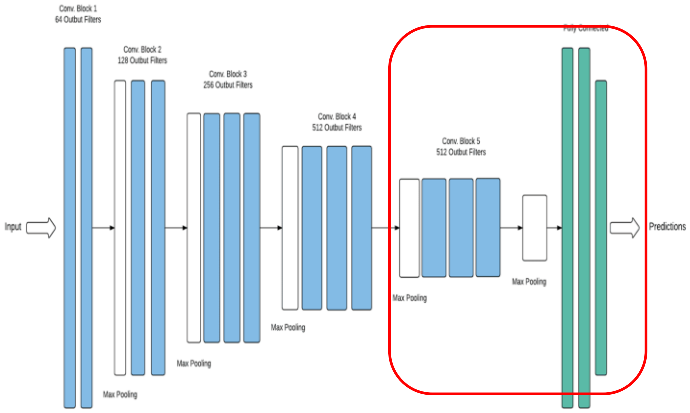

Figure 2: VGG19 network with last layers (red boxes) retrained

We also use top-5 categorical accuracy as a reference. This is not an accurate measure of the performance, since it is a multi-label classification problem. However, it will offer a more direct and easier to interpret measurement. From all 4970 samples we used for training and testing, only 1 sample has more than 5 labels. Thus, ideally the first ground-truth label(when there are multiple labels, the metric selects the first label as the ground-truth) should be in the top 5 of the model's predictions.

The classification performance on the test set for different models is shown below.

Models	Epochs	LWLRAP	Top 5 Categorical Accuracy
Deep CNN	400	0.813	88.9%
Transfer Learning	100	0.797	88.5%
VGG19 (no weight)	400	0.737	82.9%

Table 1. Classification Results for Models

The Deep CNN has the best performance in terms of classification accuracy. It is a result of iterating through hyperparameters to find the balance between complexity and overtraining. As seen above in the model introduction section, our deep CNN model is relatively small compared to popular structures that have been implemented successfully in various tasks (e.g. Resnet, Inception…) This is because of the main difficulty for this project, getting the best performing model with limited number of data.

The transfer learning performs similarly to the deep CNN model in terms of accuracy. This confirmed our expectation that the lower level features learned by the convolution layers from image data can also be applied to the audio data. Although the accuracy of the transfer learning model is slightly worse than that of the CNN model, the training time is drastically reduced, 100 epochs compared to 400. We also train the VGG19 model without pretrained weights for comparison. VGG19 is a large-scale network, and as expected, the performance is the worst of the three. There is not enough data for the model to perform optimally.

We can also analyze how the models perform for each category. Tables below shows the top-5 best and worst performance for each model.

Class	Precision	Recall	F-1 score
Bass drum	1.00	0.94	0.97
Gurgling	1.00	0.89	0.94
Finger snapping	1.00	0.88	0.94
Harmonica	1.00	0.86	0.92
Hi-hat	1.00	0.84	0.91

Table 2. Top 5 best classification performance for deep CNN

Class	Precision	Recall	F-1 score
Chirp and tweet	0.17	0.17	0.17
Walk and footsteps	0.5	0.1	0.17
Squeak	0.5	0.12	0.2
Sink (filling/washing)	0.5	0.17	0.25
Water tap and faucet	0.5	0.2	0.29

Table 3. Top 5 worst classification performance for deep CNN

From table 2, we can observe that the model performs very well at classifying instruments' sound. This is the result of well-recorded instruments' sounds have distinct frequency features. After we preprocessed the raw audio using mel-spectrogram, the model can learn these features easily.

In table 2, we can observe the classes that have bad performances are sounds with wide range of variability and easy to be confused with other sounds, even for human ears. For example, without context it is easy to confuse 'sink filling' and 'water tap'. The low recall rate indicates that a lot of the samples are unlabeled, likely caused by labeled as another similar label. It is also difficult to check confusion between classes because of the nature of multilabel classification.

Class	Precision	Recall	F-1 score
Bass drum	0.94	0.94	0.94
Kids speaking	0.95	0.9	0.92
Harmonica	1.00	0.86	0.92
Hi-hat	0.94	0.89	0.92
Finger snapping	1.00	0.84	0.91

Table 4. Top 5 best classification performance for VGG19 transfer learning

Class	Precision	Recall	F-1 score
Walk and footstep	0.0	0.0	0.0
Squeak	0.33	0.12	0.18
Fill with liquid	0.5	0.18	0.27
Scissors	0.44	0.24	0.31
Sink (filling/washing)	0.4	0.33	0.36

Table 4. Top 5 worst classification performance for VGG19 transfer learning

From table 4 and 5, we can observe that the transfer learning model performs similarly to the deep CNN model. They share most of the classes for both best and worst classification performance. There are a few classes that perform differently. It also shows good classification performance with instruments' sound and bad performance with easily confused everyday sounds.

VI. Conclusion

Our proposed method details a robust machine learning approach to classify audio clips. Using our self-developed CNN architecture, we achieve a LWLRAP score of 0.813 and a top-5 accuracy of 88.9%, when predicting 80 sound classes on the validation set. Additionally, we achieve similar performance with the VGG19 network using transfer learning with a LWLRAP score of .797 and a top-5 accuracy of 88.5%.

There is future work that can be done to further improve our sound recognition system. In order to develop a more powerful system, larger/deeper neural networks could be implemented. This would require a larger amount of training data, meaning we would need to utilize the noisy dataset provided by YFCC. While this dataset is harder to work with and its labels are less reliable, using this data would likely help generalize the system to work in noisier conditions. Techniques such as data augmentation could also be used in order to artificially construct more training data. A future application of interest is to use this system in conjunction with a computer vision system for automatically extracting information from video clips.

VII. Contributions

All team members contributed to each area of this project. These areas included data preprocessing, model construction and testing, and poster/paper writing.

VIII. References

[1] S. Chu, S. Narayanan and C. J. Kuo, "Environmental Sound Recognition with Time–Frequency Audio Features," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142-1158, Aug. 2009.

[2] “The Nature of Sound.” The Physics Hypertextbook.

[3] Kilshore Prahallad, “Spectrogram, Cepstrum and Mel-Frequency Analysis,” Carnegie Mellon University.

[4] Homburg, Helge, et al. "A Benchmark Dataset for Audio Classification and Clustering." *ISMIR*. Vol. 2005. 2005.

[5] Piczak, Karol J. "Environmental sound classification with convolutional neural networks." *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015.

[6] Hershey, Shawn, et al. "CNN architectures for large-scale audio classification." *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017.

[7] Salamon, Justin, and Juan Pablo Bello. "Deep convolutional neural networks and data augmentation for environmental sound classification." *IEEE Signal Processing Letters* 24.3 (2017): 279-283.

[8] Becker, Sören, et al. "Interpreting and explaining deep neural networks for classification of audio signals." *arXiv preprint arXiv:1807.03418* (2018).

[9] Eduardo Fonseca, Jordi Pons, Xavier Favory, Frederic Font, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra. "Freesound Datasets: A Platform for the Creation of Open Audio Datasets." In Proceedings of the International Conference on Music Information Retrieval, 2017.

[10] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li, YFCC100M: The New Data in Multimedia Research, Commun. ACM, 59(2):64–73, January 2016

[11] Kaggle Freeaudio Tagging 2019, https://www.kaggle.com/c/freesound-audio-tagging-2019/overview/evaluation