Audio Classification: Environmental sounds classification

Baljinder Kaur, Jaskirat Singh

▶ **To cite this version:**

Baljinder Kaur, Jaskirat Singh. Audio Classification: Environmental sounds classification. 2021. hal-03501143

HAL Id: hal-03501143

https://hal.archives-ouvertes.fr/hal-03501143

Preprint submitted on 23 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Audio Classification: Environmental sounds classification

Baljinder Kaur
Dept. of Computing Science (Multimedia)
University of Alberta
Edmonton, Canada
bkaur2@ualberta.ca

Jaskirat Singh
Dept. of Computing Science (Multimedia)
University of Alberta
Edmonton, Canada
jaskira2@ualberta.ca

***Abstract*—Recent advancements in the field of machine learning have led to a growing interest in many classification problems, especially involving data in the form of images, video, and audio files. One of the prominent classification problems is to classify sounds and to predict the category of that sound. Some of the applications where such a classification model can be applied in the real world are security systems, classifying music clips to identify the genre of the music, classifying different environmental sounds, speaker detection, and verification. Audio classification is the task of analyzing different audio signals. In this paper, we provide a brief overview of the area of audio classification, describing its system, various modules of feature extraction and modeling, applications, underlying techniques, and some indications of performance. Following this overview, we will discuss some of the strengths and weaknesses of current classification technologies and outline some potential future trends in research, development, and applications. We paid close attention to the inputs, network structures, temporal pooling strategies, and objective functions as these are the fundamental components of many audio classification subtasks. The paper concludes with discussions on future trends and research opportunities in this area.**

***Index Terms*—Audio Classification, Environmental sounds classification, Urban sounds Dataset, Speaker Diarization, CNN, SincNet, Deep Learning.**

I. Introduction

Audio classification is the process of analyzing audio recordings and categorizing them. Audio classification has numerous applications in the field of AI such as chatbots, automated voice translators, virtual assistants, music genre identification, and text to speech applications. Audio classifications can be many types such as — Acoustic Data Classification, Music classification, Natural Language Classification, and Environmental Sound Classification. Although current advancements in the field of Audio classification are undeniable, still making a machine learn a sound and classify it under different categories is a tedious process. The first step in Audio classification problems is to start with annotated audio data. The model requires annotated data to learn how to hear and what to listen for. The model should be smart enough to grasp the input audio and categorize it into different types. For this process, the model should be capable to learn different audios and the variations they possess. In this paper, a high-accuracy algorithm of audio classification is presented. We plan to discriminate different environment sound in a one-to-four-second window. To classify audios in these ten audio classes more accurately, we have compared multiple methods and discussed them in detail. We trained our model on available audio archives to classify particular audio and try to make it as capable as human beings to figure out the sounds in the audio sample. We applied Convolutional Neural Network (CNN) along with MFCC (Neural Networks) to improve the work efficiency of our model and tried to make comparisons with other models such SincNet.

II. Related Work

In [5] authors have used some traditional approaches to perform audio classification; they have few pitfalls as the major part is related to error cases. It can directly learn from speaker utterances to a hypersphere. In the end, SoftMax pretraining was used for improving the performance level of the system. The main advantage of this model is that it works well on both text-independent and text-dependent tasks. We will focus on our approach to reduce and get a better understanding of those. Also, their model size is too large so it's consuming a large space on CPU, we will try to tackle this in our model and reduce CPU requirements as well. Many other works have been done to improve the audio classification model. In [4], audios are classified into speech, laughter, silence, and non-speech sounds from the recordings of discussions in meetings.

i-vectors used in [3] have been one of the most popular methods for recognizing speakers and have been a state-of-the-art method that is classified using techniques such as heavy-tailed PLDA [7]. The issue with this method was over-dependence on handcrafted feature engineering as specified in [2].

The raw waveform used in [8] can help the neural network(NN) to identify and interpret complex representations of raw audio samples easily than using standard hand-crafted features such as MFCCs or FBANK. The authors proposed SINCNET architecture to develop a neural architecture that efficiently processes speech from audio waveforms and convolves the waveform with a set of parametrized Sinc functions to implement band-pass filters(low and high cut-off frequency). In SincNet all operations are fully differentiable and the cut-off frequencies of the filters are optimized with

other CNN parameters using gradient-based optimization routines. A standard CNN pipeline is applied after the first sinc-based convolution. Multiple standards convolutional, fully connected, or recurrent layers are stacked together to finally perform classification with a softmax classifier. SincNet is designed to implement rectangular bandpass filters, leading to more meaningful CNN filters. Some main model properties are Interpretability, Computational Efficiency, Fast Convergence, Few Parameters.

In this paper [1], authors have presented a method of combining multi-head attention mechanism with SVM in a recurrent framework which is most commonly used in sound recognition models. UrbanSound8K dataset was used to validate the applicability of the model to distinguish into classes. There are some advantages of SVM as a classifier, as they also compared SVM with the other two classifiers, LR and KNN. The experiment results of this method demonstrated that the proposed method could bring outstanding improvements in the accuracy of the classification. However, their main focus was on the generality of the model, and they did not perform any feature fusion as well as model parameter setting according to the characteristics of each dataset. In the future, they are going to explore the characteristics of different audio datasets and also work on the designing part for the task-specific feature representation and model parameters to further calculate the performance of the transformer.

This paper [6] describes the audio content analysis in the context of video browsing. They have discussed in detail a novel two-stage audio segmentation and classification method that segments and classifies an audio stream into speech, music, environment sound, and silence. The classes they have detailed out in the method are the basic dataset for video structure extraction. And a novel two-stage algorithm has been also developed and presented. The first stage of the classification is separating speech from non-speech, based on simple features such as high zero-crossing rate ratio, low short-time energy ratio, spectrum flux, and LSP distance. Further, it will segment non-speech class into music, environmental sounds, and silence with some rule-based classification methods in the second stage. In this process, they have introduced two new features: noise frame ratio and band periodic. After performing the experiments the result has shown that the proposed audio classification scheme is effective, and the overall accuracy rate is over 96%. In the future, they will work on the classification method to discriminate more audio classes. Along with this, they will also demonstrate developing an effective scheme to perform audio content analysis so they can improve the video structure parsing and indexing process.

III. Implemented Method

A. *Dataset*

We used UrbanSound 8K Dataset for demonstrations.This dataset contains ordinary sounds recorded from day-to-day city life. The dataset contains 8732 sound excerpts of urban sounds of length less than 4 seconds from 10 different classes such as car horns, street music, and sirens, etc. Every sound sample is labeled with its respective class. Dataset consists of two parts:

```
street_music        1000
air_conditioner     1000
dog_bark            1000
children_playing    1000
engine_idling       1000
drilling            1000
jackhammer          1000
siren                929
car_horn             429
gun_shot             374
Name: class, dtype: int64
```

Fig. 1. UrbanSound 8K Dataset Classes

- **Audio folder**: It has 10 sub-folders named ‘fold1’ through ‘fold10’. Each sub-folder contains several ‘.wav’ audio samples eg. ‘fold1/7061-6-0-0.wav’
- **Metadata folder**: It has a file ‘UrbanSound8K.csv’ that contains information about each audio sample in the dataset such as its filename, its class label, the ‘fold’ sub-folder location, etc. The class label is a numeric value ranging between 0–9 for all 10 classes. eg. the number 0 means air conditioner, 1 is a street music, and so on.

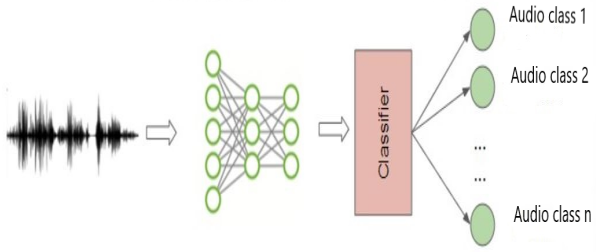

Fig. 2. The flow chart of the Audio classification model

B. *Libraries*

In this project we have used Librosa library to read the input audio signals. It has a fixed sample rate of 22050 Hz and converts different type of signals to 1D mono channel signal. Apart from this, it is normalizes audio data and set its value between range -1 to 1.

C. *Feature Extraction*

Feature extraction is a process of creating an independent vector that will represent input audio data into vector form. We have used Mel Frequency Cepstral Coefficients(MFCC) which is a hand-crafted method, to extract important information in form of features from the audio signals. MFCC extracts the

patterns based on the frequency and time characteristics of the signal. We utilized these extracted features to classify audio signals into the appropriate classes. We applied these MFCC features on UrbanSound8K.csv as it contains the mapping for all audio signals in the dataset. Then perform iteration through all audio files and extract the top 40 important features.

	feature	class
0	[-215.79301, 71.66612, -131.81377, -52.09133, ...	dog_bark
1	[-424.68677, 110.56227, -54.148235, 62.01074, ...	children_playing
2	[-459.56467, 122.800354, -47.92471, 53.265705,...	children_playing
3	[-414.55377, 102.896904, -36.66495, 54.18041, ...	children_playing
4	[-447.397, 115.0954, -53.809113, 61.60859, 1.6...	children_playing

Fig. 3. Top Feature Extraction

D. Dataset Split

We split the dataset into 2 parts: Training dataset and validation dataset in ratio 80:20 and performed training on the training dataset and tested the model on the validation dataset.

E. CNN Architecture

Our main goal is to improve the audio classification process of different sound sets and improve the accuracy rate on UrbanSounds 8K Dataset. Deep learning has shown remarkable success in multiple sound-related works such as audio classification. To gain full advantages from deep learning we need to add audio datasets at first, then extract features with MFCC and train models with CNN. In this step, we created the CNN model.

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 100)	4100
activation_4 (Activation)	(None, 100)	0
dropout_3 (Dropout)	(None, 100)	0
dense_6 (Dense)	(None, 200)	20200
activation_5 (Activation)	(None, 200)	0
dropout_4 (Dropout)	(None, 200)	0
dense_7 (Dense)	(None, 100)	20100
activation_6 (Activation)	(None, 100)	0
dropout_5 (Dropout)	(None, 100)	0
dense_8 (Dense)	(None, 10)	1010
activation_7 (Activation)	(None, 10)	0

Total params: 45,410
Trainable params: 45,410
Non-trainable params: 0

Fig. 4. CNN Model Architecture

We stacked together multiple standard CNN, fully connected, or recurrent layers and then performed classification with a softmax classifier. The softmax function transforms a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. Mostly softmax function in neural network model as an activation function. The network is configured in a way that output has N values, one value for each class in the classification task, and the softmax function normalizes the outputs and converts them into probabilities whose sum is equal to one from weighted sum values. Every value in the output of the softmax function is referred to as the probability of membership for each class.

F. SincNet Architecture

In the second part, we added the SincNet architecture. The primary reason behind using SincNet as it is one of the most suitable to process raw speech samples with band-pass filters. The operations involved in SincNet are fully differentiable and cut-off frequencies of the filters can be jointly optimized with other parameters of the CNN by using Stochastic Gradient Descent(SGD) or other gradient-based optimizations. We have selected this architecture because it is overall faster and performs better than standard CNN methods on raw audio samples. A standard CNN pipeline can be done after the first sinc-based convolution.

After this, we will use the Adversarial Feature mapping method for speech enhancement, which advances the feature mapping approach with adversarial learning as it directly transforms noisy features into enhanced ones. It will also reduce the feature mapping loss and mini-maximize the discrimination loss. In this way [8], [9] multiple standard convolutional, fully connected, or recurrent layers can then be stacked together to finally perform classification with a softmax classifier.

IV. Milestones

1) To achieve better accuracy with the publicly available UrbanSounds 8K Datasets.
2) To classify audio in specific categories.
3) Building an efficient and robust model to classify audios.

V. Tools and Resources

- GitHub Repository Link: Source Code
- Version Control: Github
- Programming Language: Python
- Tools used: Google Colab

VI. Results & Discussion

We are using CNN as it typically makes good classifiers and performs particularly well with image classification tasks due to their feature extraction and classification parts. We believe that this will be very effective at discovering patterns within the MFCC's much like they are effective at finding patterns within images. We used a sequential model, starting with a straightforward model architecture, consisting of three Conv2D convolution layers, with our final dense output layer.

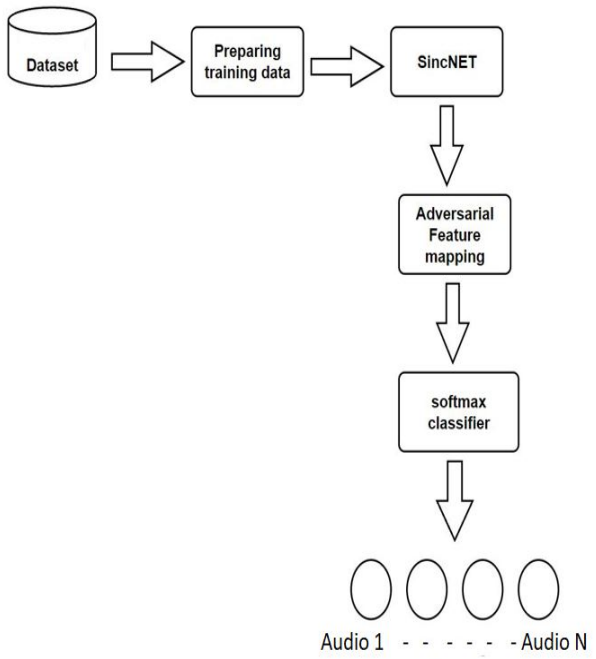

Fig. 5. Overview of Sincnet model

Our output layer has 10 nodes that match the number of possible classifications. In the training part, a CNN model needs to take a significant amount of time, so we started with a low number of epochs and a low batch size. Once we figured out that the output of the model is converging, then we increased both of them. In this way, we increased training after running a total of 100 epochs as the training accuracy level was almost 68% for CNN.

```
test_accuracy=model.evaluate(X_test, y_test, verbose=0)
print(f"Loss:{test_accuracy[0]} -- Accuracy: {test_accuracy[1]*100}%")
```

```
Loss:0.7293620705604553 -- Accuracy: 76.35947465896606%
```

Fig. 6. Validation Loss and Accuracy

VII. Conclusion

Audio Classification is a very challenging and intriguing experiment nowadays. Our CNN-based model had 68.15% accuracy on the training dataset and 76.36% accuracy on the validation set and loss observed on the training dataset is 0.9385 and loss observed on the validation dataset are 0.7294.

VIII. Future Work

- Extend models to classify different speakers.
- Focus to improve the level of accuracy.
- Explore n-shot learning approach for classification.

	dog_bark	children_playing	car_horn	air_conditioner	street_music	gun_shot	siren	engine_idling	jackhammer	drilling
dog_bark	195	0	0	0	0	0	0	0	0	0
children_playing	0	83	2	1	0	0	0	2	2	1
car_horn	3	1	187	4	1	0	4	0	1	4
air_conditioner	0	0	8	158	0	4	3	0	5	4
street_music	0	0	2	2	189	1	0	6	0	2
gun_shot	0	1	2	0	0	211	0	0	0	2
siren	0	1	1	2	0	0	80	0	1	2
engine_idling	1	0	1	0	3	0	0	180	0	2
jackhammer	0	0	2	2	0	0	1	0	193	1
drilling	0	1	12	3	1	2	1	2	2	159

Fig. 7. Confusion Matrix

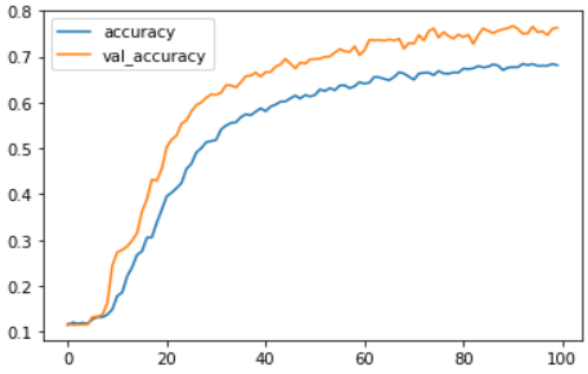

Fig. 8. Training-Validation Accuracy Curves

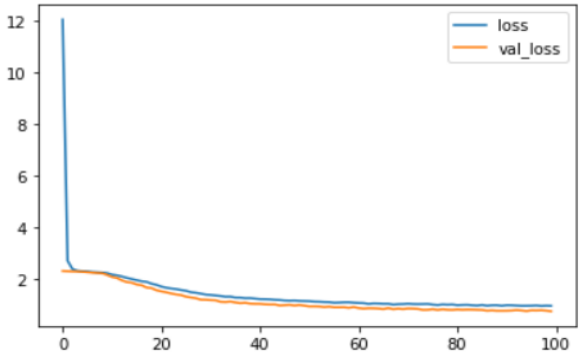

Fig. 9. Training-Validation Loss Curves

REFERENCES

[1] Ali Ahmadian, Lei Yang, and Hongdong Zhao. Sound classification based on multihead attention and support vector machine. *Mathematical Problems in Engineering*, 5 2021.

[2] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[3] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[4] Don Kimber and Lynn Wilcox. Acoustic segmentation for audio browsers. In ,, 1997.

[5] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system. *1*, 05 2017.

[6] Lie Lu, Hao Jiang, and Hongjiang Zhang. A robust audio classification and segmentation method. *the ninth ACM international conference on Multimedia*, 10 2001.

[7] Pavel Matejka, Ondrej Glembek, Fabio Castaldo, Md Jahangir Alam, Oldrich Plchot, Patrick Kenny, Lukas Burget, and Jan Cernocký. Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 4828–4831, 05 2011.

[8] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. Improving Speech Recognition by Revising Gated Recurrent Units. In *Proc. Interspeech 2017*, pages 1308–1312, 2017.

[9] Mirco Ravanelli, Dmitriy Serdyuk, and Yoshua Bengio. Twin Regularization for Online Speech Recognition. In *Proc. Interspeech 2018*, pages 3718–3722, 2018.