MSIA 432 Deep Learning
Spring 2020

Project Proposal Form

Project title	**Urban Sound Classification**
Team members	1. Srividya Ganapathi 2. Shreyashi Ganguly 3. Aditya Gudal 4. Bhavya Kaushik 5. Manish Kumar 6. Aakanksha Sah 7. Aditya Tyagi

Problem statement

The objective of this project is to use deep learning to classify urban sounds from the UrbanSound8k dataset. We have 10 classes of urban sounds and we will predict probabilities for a given audio clip belonging to each of the classes. The model performance evaluation metric will be the **classification accuracy**.

Some applications of this project are -

1. Assistive devices for hearing impaired individuals
2. Smart home security systems
3. Predictive maintenance with airborne sound analysis

So, hearing impaired individuals, security system companies and manufacturing companies can benefit from this solution.

Some relevant industrial work in this domain are -

1. The Apple Air Pods Pro are using Active Noise Cancellation and Transparency mode to filter external sounds. Several audio equipment companies are focusing on smart hearables that let the user customize and adjust the levels of external noise they perceive. This improves the overall audio experience for the user. Deep learning audio classifiers have helped in this research.
2. Smart wearable devices like Fitbit have been developed with user friendly designs and a rapid decision-making mechanism for important sounds for hearing impaired people. For instance, a person who is hard of hearing does not hear doorbell and telephone sound, displays and vibrations are assistive modes that are being used to make him/her aware of these sounds.
3. Companies like IBM, SAP and Siemens use predictive maintenance. IBM Watson is deployed for predictive maintenance of Kone's elevators and DC Water's Hydrants. The railway industry also uses track monitoring systems to identify voids underneath tracks.

Dataset

- We will be using UrbanSound8K dataset consisting of 8732 labeled sound excerpts (=<4s) of urban sounds. There are 10 classes :air conditioner, car_horn, children_playing, dog_bark, drilling, enginge_idling, gun_shot, jackhammer, siren, and street_music. Along with the individual files in .wav format corresponding to audio clips a metadata(.csv) is provided which tags the files to its class.
- The total size of the data in compressed format is 5.7GB. While 7 of the 10 classes have 1000 observations each, car_horn, gun_shot and siren have 429, 374 and 929 observations respectively.

- The dataset might not be a true representative of the actual population, example gunshot incidents are extremely rare, however the dataset provides a good number of samples. This should not be a problem though and instead help in better learning of different sounds.
- Data augmentation of a sound data set can be done and will be explored in our project. Augmentation techniques such as Noise Injection, change in pitch, change in speed and shifting time can be used in order to generate synthetic data.
- The data is open sourced and can be downloaded from the url : https://urbansounddataset.weebly.com/download-urbansound8k.html There are no restrictions upon use of data.

Proposed Technical Approach

We propose to train a deep convolutional neural network classifier on carefully extracted visual representations of each of the audio samples to address the task of environmental sound classification.

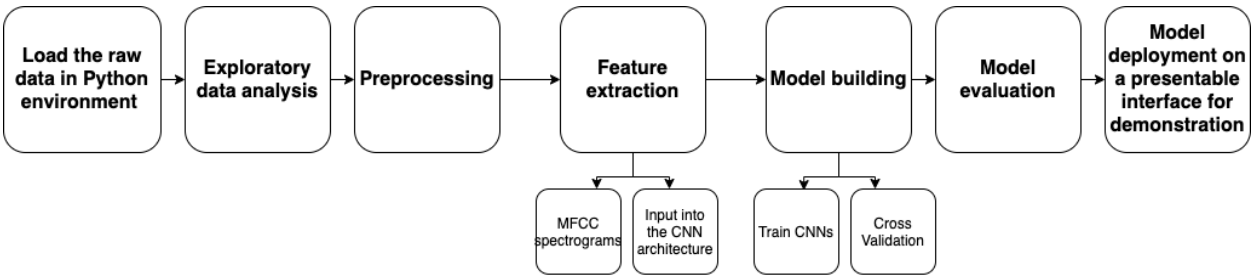
We plan to use the Librosa package in Python to analyze the audio files and extract attributes like the number of audio channels, sample rate and bit-depth. Thereafter we will preprocess all the sound clips using Librosa to ensure consistency across the whole dataset. In order to extract the features, we will need to train our deep learning model, we plan to create a visual representation of each of the audio samples. This will allow us to identify features for classification, using the same techniques used to classify images with high accuracy (CNNs). Spectrograms are one of the most popular techniques for visualizing the spectrum of frequencies of a sound and how they vary during a very short period. We will be using a similar technique known as Mel-Frequency Cepstral Coefficients (MFCC), which has recently become a highly recommended technique in the domain of audio analysis. We will then use cross validation to compare different CNN architectures and tune the hyperparameters to obtain the best classifier.

Humans can detect and identify countless varieties of environmental sounds instantly. At the end of the human ear canal there is the eardrum which vibrates and precisely transmits almost the exact information of the incoming sound wave (if its frequency is between 20Hz to 20,000 Hz). The inner parts of the human ear (cochlea) have sophisticated systems of sensitive receptors (hair cells) which stimulate the auditory nerve, which in turn conducts the signals to the brain for further processing. Once the brain has memorized a sound, its repetition triggers the auditory memory and the sound is classified.

To our knowledge, the way we are solving this problem of environmental sound classification doesn't align with the way the human brain deals with it. We are taking the sound wave and creating a visual representation out of it in order to leverage the advanced algorithm of CNNs to capture patterns in the images which will help in the task of audio classification. Humans don't encounter MFCC spectrograms in daily life to identify the category of a sound because we don't need the intermediate step of creating a visual representation. Nevertheless, the MFCC spectrograms have also been considered very efficient in capturing enough information about the sound clips to differentiate between the different categories and hence they are enough to undertake the task at hand.

MSIA 432 Deep Learning
Spring 2020

Flow chart:



Metrics

- As this is a classification problem, the success criteria for the project is to **achieve a Correct Classification Rate (CCR) > 80% (test dataset).** The reason for prioritizing CCR over other metrics such as F1 score, AUC is that the sound classes are mostly balanced (Each class has ~10% of the total records) and targeting for a high CCR value itself will ensure overall model accuracy and reliability.
- This project has huge applicability in areas such as aiding hearing impaired individuals and designing smart homes. For these applications, the need for correctly classifying only a specific set of classes over the others is not essential. Therefore, we do not prioritize the need of correctly classifying sounds belonging to only a selected set of classes, but all classes in general.
  To summarize, for this domain - high classification rate is more meaningful than false positive rates for a class.
- We examined some research papers, articles as well as GitHub projects [Links in the appendix] to learn of some additional ways of ensuring high model reliability.
- <u>Learnings:</u>
  - For such projects, CCR is generally preferred for measuring model accuracy
  - Ensure equal representation of all classes in the test dataset or else few classes could bias the accuracy metrics
  - It is valuable to additionally ensure a minimum of 90% accurate predictions at a class (or sound label) level
- Incorporating the above mentioned checks will help build a model that predicts the class of a sound excerpt reliably

Backup plan

The success of the project mostly depends on the ability of our model to distinguish patterns in the different categories of sounds. We have identified the following possible roadblocks in achieving the level of performance stated above,

- Difficulty to distinguish between overlapping waveforms of some of the classes of sounds, e.g.,
  - repetitive sounds of air conditioners, drilling, engine idling, jack hammer
  - sharp peaks of dog barking and gunshots

  Possible solution would be to try to incorporate minute differences via engineered features
- Some audio files may have varying degrees of background noise, causing the model to overfit to the noise itself. To solve for this, we would be exploring various de-noising packages such as pyaudio, noisereduce. We can also test different network architectures and hyperparameter tuning to protect against said overfitting
- Some of the sound categories such as children playing and street music might be a combination of various sounds and hence show a lot of variation between samples. The model might have a

low accuracy in predicting these classes. We will take special notice of the model performance in these categories of the test data. If needed, we will use different techniques to amplify these signals.

- By the time we reach our first checkpoint, we should have been done with Literature review, EDA, feature engineering and extraction and a baseline CNN model.
- We expect to build a deep learning model that will be able to classify a testing audio file as our minimum viable product.

Related work

- Automatic urban sound classification is a growing area of research with applications in multimedia retrieval and urban informatics. A lot of researchers have worked on this dataset before. This dataset is available on https://urbansounddataset.weebly.com/urbansound8k.html. A lot of blog posts, research papers and articles have been published. Many people have trained deep learning models such as CNNs, RNNs with LSTMS https://arxiv.org/pdf/1805.00237.pdf, http://noiselab.ucsd.edu/ECE228/Reports/Report15.pdf,https://www.preprints.org/manuscript/201811.0509/v1/download to classify sounds. Some people have used traditional signal processing techniques to extract features such as spectral shift, mel-spectrogram, chroma-stft, mel frequency cepstral coefficients that help distinguish the sounds.https://www.ijrte.org/wp-content/uploads/papers/v7i5s3/E11900275S19.pdf
- Hand crafted features extracted and CNN-based feature extractor gives a testing accuracy of 73.1 +/- 6.2(https://www.preprints.org/manuscript/201811.0509/v1/download), VGG with CNN feature vectors of length ≈ 3500 gave a testing accuracy of around 72.86% (https://arxiv.org/pdf/1805.00237.pdf). Using CNNs on handcrafted features gave a testing accuracy of 87.8% and using Boosting it gave a testing accuracy of 91.9%.(http://noiselab.ucsd.edu/ECE228/Reports/Report15.pdf).
- With our model which will be a CNN with MFCC Spectrograms we hope to achieve an accuracy of around 85%. We will revisit and try different CNN architectures to improve our accuracy over time.

References of code:

Music Feature Extraction in Python
Audio Classification Using CNN — An Experiment - AI Graduate
Udacity-ML-Capstone/4 Model Refinement.ipynb at master · mikesmales/Udacity-ML-Capstone