URBAN SOUND AUDIO CLASSIFICATION WITH DEEP LEARNING

Emma Cooper
eyc4xd@virginia.edu

Chris Lee
cl7zn@virginia.edu

Connie Cui
qqv3uu@virginia.edu

Alex Bass
ujb3bu@virginia.edu

December 14, 2022

ABSTRACT

Previous research has shown that noise pollution is linked to several mental and physical health conditions in humans, wildlife and ecosystem deterioration, and child development issues, as well as other concerns. Drawing upon Convolutional Neural Networks, Alexnet, and popular feature extraction frameworks such as spectograms, we aim to create a Deep Learning Algorithm to help classify noise pollution from a large audio dataset.

1 Motivation

With the ever-growing population in urban cities, the concern of noise pollution rises. New challenges arise in the policing of cities and in the increased detriment to both human and environmental health. Coined in the early 1970s, the term ‘noise pollution’ refers to “any unwanted or disturbing sound that affects the health and well-being of humans and other organisms” [1]. The World Health Organization (WHO) also classifies noise pollution as any noise that exceeds 65 decibels (dB). Noise pollution is commonly found in cities and more urban areas, with examples ranging from traffic, to construction, to even animals such as dogs when they bark or howl.

Studies find that noise pollution encourages lawbreaking and violent crime. For example, increasing background noise by 4.1 decibels causes a 6.6% increase in the violent crime rate [2]. In the area of policing, artificial intelligence introduces new ways to mitigate crime-related problems and help investigation after crimes have been committed [3]. From designing an audio-based surveillance system capable of automatically detecting, classifying, and registering a sound event to having the ability to discern a baby crying from air conditioning or glass breaking when monitoring homes, the potential benefits of robust sound classification in this area are immense [4].

Another downside to increased noise pollution is the negative effect it has on our health, both mentally and physically. Noise pollution can induce increased anxiety, stress, hysteria, and depression, decreased productivity, hearing loss, high blood pressure, heart disease, stroke, speech interference, sleep disturbance, and many other negative side effects [1]. In addition, children that experience heavy amounts of noise pollution growing up have found their memory, reading skills, and attention spans affected as well. Furthermore. noise pollution has an enormous environmental impact and does serious damage to wildlife - it interfere with breeding cycles, mating calls, basic survival skills (ex. finding food), and more, the result of which is hastening the extinction of some species and simply disrupting many animals’ lives [1].

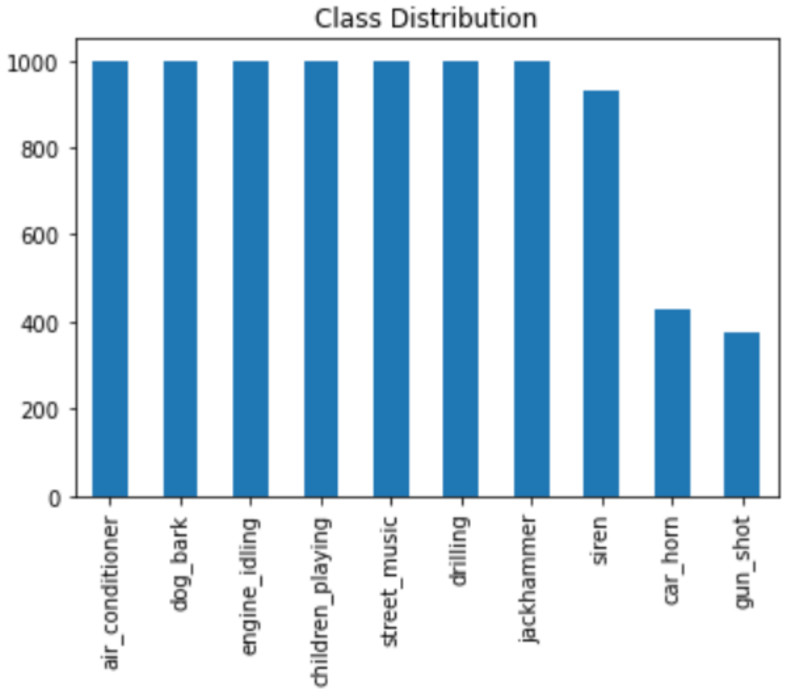
Our group seeks to explore models and machine learning algorithms that are able to identify sounds related to noise pollution in urban areas based on their sound excerpts and frequencies. By doing so, we hope our algorithms and findings can help reduce noise pollution and ensure correct noise management to establish safer cities and prevent further negative effects on ourselves and our environment.

2 Literature Survey

Through multiple series of classification experiments with various algorithms and features being tested in urban sound classification, it appears that there are promising results in the future of this field. Many related works found come from more recent years spanning from the late 2010s to 2021. Based on these reports and their findings, the current most optimal urban sound classifying machine learning algorithm has been found to be the convolutional neural network (CNN). More commonly known for its success in the image classification field, the convolution neural network uses its convoluting layers to share context within the small 'neighborhoods' in the network to allow related information from both images and audio to be more closely associated and understood within the algorithm. Currently, there is not a specific CNN architecture that has been determined to be the optimal architecture for urban sound classification as not enough research has been done to reach such a conclusion. One other method that may provide more feature options for urban sound classification is through audio feature extraction, which converts each sound file to their various audio characteristics. Examples of features that were found to be extracted in various reports include:

- Mel-Frequency Cepstral Coefficients (MFCC): Coefficients derived from a cepstral representation of the audio clip
- Mel-scaled spectrogram: Psychoacoustic scales that capture the distances from low to high scale frequency .
- Chromagram: Pitch class profiles that capture harmonic and melodic characteristics within the music
- Contrast: The difference between parts of a sound or different instrument sounds
- Spectral Contrast: Representation of the strength of spectral peaks and valleys in each a sub-band as contrast distribution
- Tonnetz: Representation of tonal space with tonal centroid features

In most related works, the most commonly used feature extraction method was the Mel-Frequency Cepstral Coefficients (MFCC), while several works tried all features or certain subsets of them. Overall, most models across the board using some form of Mel-Frequency Cepstral Coefficients (MFCC) as the primary feature and a convolutional neural network (CNN) algorithm had excellent performances, with accuracy scores ranging within the 90 percent to 99 percent accuracy range.



3 Methodology

For our baseline models, we have focused on using Convolutional Neural Networks (CNNs). We chose this machine learning technique specifically due to the observations we noted in our literature review. Based on several other urban sound classification studies, it seems that convolutional neural networks perform the best among many other machine learning techniques explored, so we decided to focus on different variations of convolutional neural networks that may help improve model performance.

For our preliminary experiments and baseline model, we have focused on an AlexNet architecture for our convolutional neural network. This is due to the Alexnet's widespread popularity and high performance in the CNN community, particularly in the image classification sphere. We will start our modeling with AlexNet to see if it is able to produce similarly successful results. The AlexNet architecture consists of a combination of five convolutional or max-pooling layers, three fully connected layers, and two dropout layers. All layers except for the final output layer use a Relu activation function and the final output layer uses a Softmax activation function [10].

We extended our use of AlexNet architecture based models into our later models as well, incorporating ten fold cross validation methods to ensure that our preliminary results were consistent across the dataset. Our later models include applying our 10 fold cross validation methods to an all features AlexNet architecture model, an all features AlexNet architecture with an additional dropout layer model, individual AlexNet architecture models for each single feature, and an all features GoogLeNet architecture model.

We explored a GoogLeNet architecture to see how it would compare to our AlexNet architecture when performing on our urban audio dataset. The GoogLeNet architecture consists of a combination of 22 layers (or 29 layers including the pooling layers) that combine into 9 inception modules. These inception modules are used to help resolve many issues that larger networks may face, which includes being prone to overfitting and suffering from either exploding or vanishing gradients. "The Inception module is a neural network architecture that leverages feature detection at different scales through convolutions with different filters and reduced the computational cost of training an extensive network through dimensional reduction" [12]. More specifics regarding the GoogLeNet architecture and its intricacies can be found at our source [12].

4 Experiments

For our preliminary experiments, we started with two baseline models after conducting our feature extraction on the data's audio files: an AlexNet CNN model with MFCC features as well as an AlexNet CNN model with all audio features. These two models were conducted with a simple training and test dataset. Following these preliminary experiments, we sought to expand to utilizing 10 fold cross validations methods on our next models, which included our AlexNet CNN with All Features

4.1 AlexNet CNN with Individual Features

Our first baseline models that we explored used a standard AlexNet architecture convolutional neural network using each of the extracted features using the Librosa package: the Mel-frequency cepstral coefficients (MFCC), the Mel-scaled spectrogram, Chromagram, Contrast, and Spectral Contrast features. We trained and tested the model using 10-fold cross validation on the predetermined folds of our dataset. Our model had 50 epochs and a batch size of 50.

To gauge model performance, we examined the accuracy metrics of a model trained on the first 9 folds of MFCC, while validating on the 10th fold, being able to reach a 99.1% training accuracy and 62.1% validation accuracy. The accuracy seems inconsistent across the labels: air conditioner predictions were almost as good as random guesses, while siren and car horn accuracies were over 90% each.

The MFCC model's validation accuracy dropped to 50.7% when averaging our 10 fold cv, likely because the 10th fold may have had the more accurately predicted classes than other folds. Other features had lower validation accuracies, as seen in the table given in Section 5. The Mel-Scaled Spectrogram had the highest average validation accuracy out of the individual feature models, but only reaching 52.1% accuracy. The other models did seem to perform poorer, with mixed results.

Individual Models' 10 Fold CV Model Description	Average Validation Accuracy
CNN AlexNet with Tonnetz	41.7%
CNN AlexNet with Mel-Scaled Spectogram	52.1%
CNN AlexNet with Chromogram	43.0%
CNN AlexNet with Constant-Q Chromagram	44.7%
CNN AlexNet with MFCC	50.7%
CNN AlexNet with Spectral Contrast	34.8%

4.2 AlexNet CNN with All Features

Our second baseline model we experimented with was the exact same in architecture, and the only difference from the first baseline model is in our feature set. Whereas our first baseline model had only the Mel-frequency cepstral coefficients as the feature variables, this second model has a feature set containing other audio features extracted from the Librosa Python library including mel-scaled spectogram, Chroma Energy-Normalized, constant Q-chromagram, and a power spectogram. These features each attempt at measuring different spectral attributes of the data for a more well-rounded training set.

In our training notebook [11], we extracted each of these features from our 8732 audio clips. Then we implemented the same AlexNet model we applied to our previous data, but this time with these additional features. Similarly to our first baseline model, we trained and tested the model using the first nine folds of the dataset as our training data and our tenth fold as our test data. After running with 50 epochs, we saw good evidence of overfitting in this model as well because our final training accuracy is 99.5% where as our final test accuracy is 63% though some epochs were up to 70%. While we think this performed well for a baseline model, it was only marginally better than our previous model which had a test accuracy of 62%. We speculate a few changes will bring a better performing model.

4.3 AlexNet CNN with All Features and 10 Fold Cross Validation

After creating the baseline models, we wanted to implement 10-fold cross validation to create the best possible model, and so that our model can see all of the data we have available. So, the first model we implemented was the AlexNet CNN with all the features. Unfortunately, from this experiment we found that the crossfold validation did not improve our previous validation accuracy and actually was lower at an average 61% validation accuracy instead of the previous 63%. All models discussed here are found on our GitHub[11].

4.4 Altered AlexNet CNN with All Features and 10 Fold Cross Validation

One of the biggest patterns we noted in our initial models was a lot of overfitting when they were being trained. We saw high training accuracies yet lower validation accuracies in our epochs indicating that our model was fitting too closely to the training data to the point where it affected the model's ability to predict our validation data accurately. In order to try to counter this in our AlexNet architecture, we tried adding an additional dropout layer to the original AlexNet architecture while maintaining all other features and parameters from the previous model. Dropout layers are typically used to help prevent overfitting in deep learning models so we thought this may be an appropriate way to address our overfitting problem while maintaining the overall AlexNet structure. In the end, while some of the validation accuracies for our various 10 fold showed some promise, with our 9th fold having a 69% validation accuracy, our overall average validation accuracy across the ten folds ended up being 60.37%, which was surprisingly less than the average 10 fold cross validation accuracy for our original AlexNet.

4.5 GoogLeNet CNN with All Features/10 Fold Cross Validation

In addition to experimenting with different features and methods, we also explored different architectures - one of which was GoogLeNet. Unlike the AlexNet architecture, GoogLeNet's implementation is 22 layers deep and utilizes inception modules and average pooling. Despite it's many layers, its model has less parameters to AlexNet yet is still known to both compute faster than and outperform AlexNet. However, when we applied the GoogLeNet architecture to a baseline model that uses the same features and methods described in "AlexNet CNN with All Features" our results showed that our model did not outperform AlexNet. The model had the following training accuracies [dense 1: 98% , dense 3: 99% , dense 4: 98%] and the following test accuracies of [dense 1: 54% , dense 3: 53% , dense 4: 52%] which is worse compared to AlexNet's performance.

Since layers tend to decrease accuracy, instead of continuing with the full GoogLeNet architecture we experimented with a partial/mini version that replicates the first path of GoogLeNet. This partial architecture was first applied to the same model as the full one. In the all features model it resulted in a final training accuracy of 97% and a final test accuracy of 66% which is slightly better than AlexNet and much better than the full GoogLeNet architecture when they were implemented. However, it also signifies over-fitting. Due to it's performance in the all features model, we also tried it in a second model with 10 fold cross-validation. In the second model, this architecture resulted in an average accuracy of 60% which is worse compared to AlexNet's performance with the same method (everything kept constant besides the architecture implemented).

4.6 Other Alternative Models

Besides AlexNet and GoogLeNet, we also explored LeNet-5 and VGG-16 - applying them to the all features model. Unsurprisingly, LeNe-5 had a similar performance that was only slightly worse compared to AlexNet, considering AlexNet improves upon LeNet-5. VGG-16 ended up performing much worse than AlexNet which was surprising since the VGG CNN builds upon AlexNet. Perhaps its because of using audio inputs rather than image inputs. However, with more time, it would be interesting to investigate these architectures even further to understand their results.

5 Results

The table below shows the average 10-fold validation accuracy of our various models to help summarize our different model experiments and their error analysis.

10 Fold CV Model Description	Average Validation Accuracy
AlexNet with All Features	61.9%
AlexNet with All Features and Additional Dropout Layer	60.4%
CNN GoogLeNet with All Features	59.1%
CNN AlexNet with Mel-Scaled Spectogram	52.1%
CNN AlexNet with MFCC	50.7%
CNN AlexNet with Contrast	44.7%
CNN AlexNet with Chromogram	43.0%
CNN AlexNet with Tonnetz	41.7%
CNN AlexNet with Spectral Contrast	34.8%

While all of our average 10-fold cross validation accuracies could be improved on, it's important to examine the average validation accuracies for each urban audio label. The below table shows the distribution of the validation accuracy from our baseline train/test AlexNet CNN model with all 6 features by the 10 audio labels. As we can see, there is quite a spread of how well the model was able to identify specific classes of our audio files, with a gun shot being the easiest to identify with a 93% accuracy and sirens being the most difficult for the model to identify with an accuracy of only 43%. This supported our reasoning for looking into the various individual feature models, as it seems that different models will identify different sounds better.

Label Name	Average Validation Accuracy
Gun Shot	93%
Car Horn	85%
Street Music	79%
Dog Bark	78%
Children Playing	63%
Drilling	60%
Air Conditioner	56%
Jackhammer	56%
Engine Idling	46%
Siren	43%

6 Conclusion

After experimenting with our various models, it appears that the AlexNet fit on all extracted features model has the best performance with an average validation accuracy of 61.9%. However, this is not a significant increase in accuracy compared to the others, as seen above. This means that while our results do in fact support our original hypothesis of the AlexNet CNN with all features performing the best, the evidence is not entirely conclusive or satisfactory considering how low the average validation accuracy was and how close it was in value to the other models' accuracies.

The validation accuracies per class also do not seem to be entirely consistent; it may be useful, given more time and resources, to run more models to compare which classes each of the models performed best for. MFCC, for example, was very successful at classifying the audio of car horns and sirens but poorly sorted our air conditioner audio. Moving forward, we could also potentially explore more modern architectures and potentially use combinations of various features instead of all of them or only one of them in each model. The biggest issue to resolve moving forward would be how to prevent our model from overfitting on our training data, as even after trying adding an additional dropout layer, we weren't able to improve the performance of our models and prevent it from overfitting. Another limiting factor in what models were able to run is in how long it took to extract our features from the audio file data as well as the complexity and time it took to run our 10 fold cross validation process on each different model, especially when we tried to increase epoch count to experiment.

Overall, our results did in fact support our initial hypothesis and we gained valuable insight in how certain specific audio features were better at identifying certain sounds, which may be noteworthy in the field.

7 Member Contribution

7.1 Emma Cooper

Participated in Milestone I, was unable to participate in Milestone II due to a prior commitment. Compiled the Motivation section of the Project Proposal. Explored alternative architectures for Milestone III, configured LeNet-5, GoogLeNet, and VGG-16 architectures for baseline models and configured GoogLeNet further to use with 10 fold cross validation framework. Wrote up the sections related to GoogLeNet and other alternative models.

7.2 Chris Lee

Worked on the Literature Review Overview and Intended Experiments section of Milestone I. Extracted and parsed a datatable consisting of MFCC features. Ran the AlexNet model on this extraction, training the model on the first 9 predetermined folds and validating on the last fold. Ran 10 fold CV on AlexNet models for each of the 6 features individually. Updated the corresponding Experiments section of the final paper.

7.3 Alex Bass

Unable to participate in Milestone I due to prior commitments. For Milestone II, I extracted more features, implemented AlexNet model with those features, created and kept up GitHub, contributed writing to Results, Abstract, and Next Steps sections. For Milestone III, I ran the AlexNet CNN model with 10 - fold cross validation and wrote up that section in the paper. Also, I created the slide deck for our presentation.

7.4 Connie Cui

Compiled the summary in the Literature Review and the Dataset and Related Works sections of the Project Proposal in Milestone I. Prepared code framework for preprocessing and feature extraction process for the baseline models in Milestone II. Compiled the Method section and parts of the Preliminary Experiments and Next Steps sections of Milestone II. Prepared all coding framework for the 10 fold cross validation process and creation of metric evaluation table from this process for teammates' various models. Ran the AlexNet 10 fold cross validation model with an additional dropout later. Compiled the updated Methodology section and part of the Experiments and Conclusion sections in the final paper.

References

[1] National Geographic “Noise Pollution”, [Online]. Available: *https://education.nationalgeographic.org/resource/noise-pollution*

[2] Timo Hener, “Noise Pollution and Violent Crime”, 2022. [Online]. Available: *https://www.sciencedirect.com/science/article/pii/S0047272722001505*

[3] João Pedro Duarte Galileu, “Urban Sound Event Classification for Audio-Based Surveillance Systems”, 2020. [Online]. Available: *https://repositorio-aberto.up.pt/bitstream/10216/126838/2/392231.pdf*

[4] Christian Gunther, Kevin Le, Mike Ranis, and Derar Durubeh, “Urban Sound Classification”, 2019. [Online]. Available: *http://noiselab.ucsd.edu/ECE228*$_2$019$/Reports/Report36.pdf$

[5] Massoud Massoudi, Siddhant Verma, and Riddhima Jain, “Urban Sound Classification using CNN”, 2021. [Online]. Available: *https://ieeexplore.ieee.org/document/9358621/*

[6] Isha Agarwal, Parul Yadav, Neha Gupta and Sarita Yadav, “Urban Sound Classification Using Machine Learning and Neural Networks”, 2021. [Online]. Available: *https://link.springer.com/chapter/10.1007/978-981-33-4501-0*$_3$1

[7] Justin Salamon, Christopher Jacoby, Juan Pablo Bello, “A Dataset and Taxonomy for Urban Sound Research”. [Online]. Available: *http://www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon*$_urbansound_acmmm14.pdf$.

[8] Aaqib Saeed, ”Urban Sound Classification, Part 1.” [Online]. Available: *http://aqibsaeed.github.io/2016-09-03-urbansound-classification-part-1/?fbclid=IwAR013w3zKLA5qL0FjUztliqP2aW 9yyvlS2MlHLM plPSS1 fX21PERAlw*

[9] Amrit Khera, ”Urban Sound Classification” [Online]. Available: *https://github.com/AmritK10/Urban-Sound-Classification*

[10] Shipra Saxena, "Introduction to The Architecture of Alexnet" [Online]. Available: *https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/*

[11] Alex Bass, Chris Lee, Emma Cooper, Connie Cui, "Urban Audio Classification" [Online]. Available: *https://github.com/acbass49/urban_audio_classification*

[12] Richmond Alake, "Deep Learning: GoogLeNet Explained" [Online]. Available: *https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765*