Project Report

On

Classification of Urban Sounds

using

Deep Learning

By Team 5

Akshay Punwatkar
Xinwen (Melody) Li
Tzu-Chun
Andrew Patterson
Derek Wales

(April 2020)

Table of Contents

1 Abstract

While much of the focus of machine learning had been on text and image analysis, the audio analysis had gained scholars' attention and became a subdomain area. In this project, sound excerpts from ten different sources, provided under the UrbanSound8K [1] dataset, are analyzed, and Mel-frequency Cepstral Coefficients (MFCC) [2] extracted from the sounds are used to create multi-class classification models. Classification algorithms, namely Support-Vector-Classifier (SVC), Multi-Layer-Perceptron (MLP) and Convolutional Neural Networks (CNN), are used to model the classification of the sounds. Furthermore, the models are evaluated on the basis of the F-1 Score, Accuracy, and confusion matrix. Classification algorithm for sounds could serve several purposes ranging from security or surveillance to improvement of human-computer voice interaction.

2 Introduction

Sound classification is a rapidly growing, emergent field within the realm of machine learning. While speech and music recognition had been the primary focus of researchers, research related to the analysis of urban acoustics [3][4] is relatively low. Classification of urban sounds such as that of the sound of an air conditioner or an idle engine or a jackhammer may not seem appealing in a glance. However, such classification algorithms could serve several purposes. A simple example could be the identification of the sound of a jackhammer used for illegal drilling, or identification of engine idling sound which could help notify certain users in case their car is still running.

Sound, by definition, is a vibration, with properties such as frequency, amplitude, and depth. Moreover, each sound or in our case, each source of sound has a different set of properties that are easily identifiable by a human. However, it is not straightforward when a computer has to identify and differentiate between different sounds. In computers, sounds are encoded using specific formats such as mp3 or wav. For machine learning algorithms, sound needs to be converted into numerical features that could be used in a machine learning algorithm.

Mel-Frequency Cepstral Coefficients (MFCC), are one such set of features obtained via several complex transformations of the encoded sound. Once generated, MFCC features could be used for training machine learning algorithms for sound classification. Algorithms, namely SVC, MLP, and CNN, are used in this project to perform the classification.

Classifying sounds in an urban context has several benefits. Classifying gunshots can allow city officials to deem whether an area has high crime and needs additional measures to prevent murders and robberies. Classifying construction noise levels can provide city officials more information about construction work ongoing in their city. This project builds on previous work to correctly identify urban sounds.

3 Background

Sound classification is a popular topic of research. Combining the feature extraction and machine learning methods, computers can identify the sound and have advanced applications such as a virtual assistant, Siri or Alexa.

3.1 Sound Data and Classification

"Sound is a traveling longitudinal wave which is an oscillation of pressure."

The period and the amplitude of the sound define how it would sound like for humans. However, to make the computers understand sounds, transforming sounds to another format, numeric, is necessary. A microphone is used to sample from the sound to generate numeric data. The characteristic of the data is based on the duration and amplitude of the sound wave itself, along with other properties such as sampling rate (samples per second), channels (number of microphones used to collect the sound), and bit rate (the number of bits encoded per second). Feature extraction would be applied to the sound data to reduce the dimension to use machine learning models.

3.2 Mel-Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients (MFCC) is a technique that extracts features from sound data introduced by Davis and Mermelstein in the 1980's [2][5]. The coefficients make up the Mel Frequency Cepstrum (MFC), which is used as input data of the classification model. To obtain the coefficients, first, Fourier Transformation is applied to the sound signal to create a cepstrum. Based on human perception of sounds and to more closely mirrors human listening capabilities, the cepstrum is further transformed using the Mel scale, "a scale that relates the perceived frequency of a tone to the actual measured frequency." The MFC is highly effective in audio recognition and in modeling the subjective pitch and frequency content of audio signals.

3.3 Classification Methods

Following classification algorithm were used to model sound classification.

3.3.1 Support Vector Classifier (SVC)

A support vector classifier (SVC) is one of the supervised learning methods for classification under the support vector machine (SVM). This method uses a kernel function to

increase the dimension of the data then find the support vector classifier that best classifies the data. SVC is effective and memory-efficient in high dimensional data and is widely applied to classification problems in computer vision and sound identification.

3.3.2 Multi-Layer Perceptron (MLP)

A Multilayer Perceptron (MLP) or Feed Forward Neural Network (FFNN) is the most typical neural network model for classification. The input data goes through several fully connected layers, "*each of these layers is composed of units that perform an affine transformation of a linear sum of inputs*," and the model would give the prediction of the class for the input data. MLP is effective to approximate the complicated classification function and make accurate predictions.

3.3.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a class of deep learning network which is widely used to analyze images that achieves high accuracy. The encoding layer extracts essential features from the input data (picture) using convolution and feeds it to the fully connected layers or MLP to perform classification.

3.4 Previous Work

The original work referenced in the paper "*A Dataset and Taxonomy for Urban Sound Research*" on the earlier version of the UrbanSound dataset dates back to 2014[3], by research performed at New York University (NYU) by Justin Salamon, Et. al. The work discusses the properties of the audio files and baseline classification on it. Other research work includes, "Environmental sound classification with Convolutional Neural Networks", [6] done by Karol J. Piczak from Warsaw University of Technology, where he applied CNN to three different sources of sound data and proved that CNN could classify it in high accuracy.

Another research, "End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network," [7] conducted by University du Québec, discusses sound classification. The team presented an end-to-end 1D CNN initialized with Gammatone filter banks, which significantly reduced the number of features of the input data. They succeeded in achieving the same level of model accuracy with relatively small computation cost.

4 Data

The dataset used in this analysis is referred to as UbranSound8K [1] dataset. It contains 8732 labeled sound excerpts (<=4 seconds) of urban sounds from 10 classes. *Figure-1* shows the distribution of sounds sample among different classes with class names.

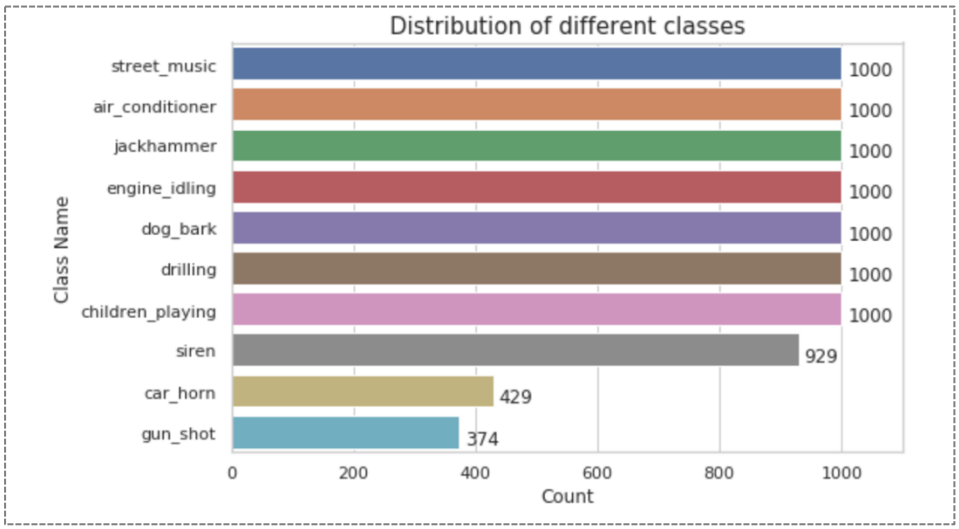

Figure 1: Distribution of the number of samples of different classes in UrbanSounds8K dataset

Based on the distribution, it can be established that the data is slightly un-balanced in cases of a couple of classes, namely *car_horn* and *gun_shot,* with 429 and 374 samples, respectively. Data imbalance could cause an issue with model training and performance, particularly for the under-balanced classes.

4.1 Visualizing Data

Sound data can be visualized in multiple forms. A wave plot shown in *Figure-2* shows the wave plots of a random sample of each class in the dataset, which shows the change in the amplitude (loudness) and the frequency (pitch) of the sound with time. As can be seen, sounds like children_playing or car_horn or gunshot are discreet in nature, while other sounds are continuous. Similarly, air conditioner and engine idling have a similar plot, which could cause issues with the classification process. Also, the duration varies for sound from a different class, which would cause issues after the MFCC feature extraction process and which would be handled using padding.

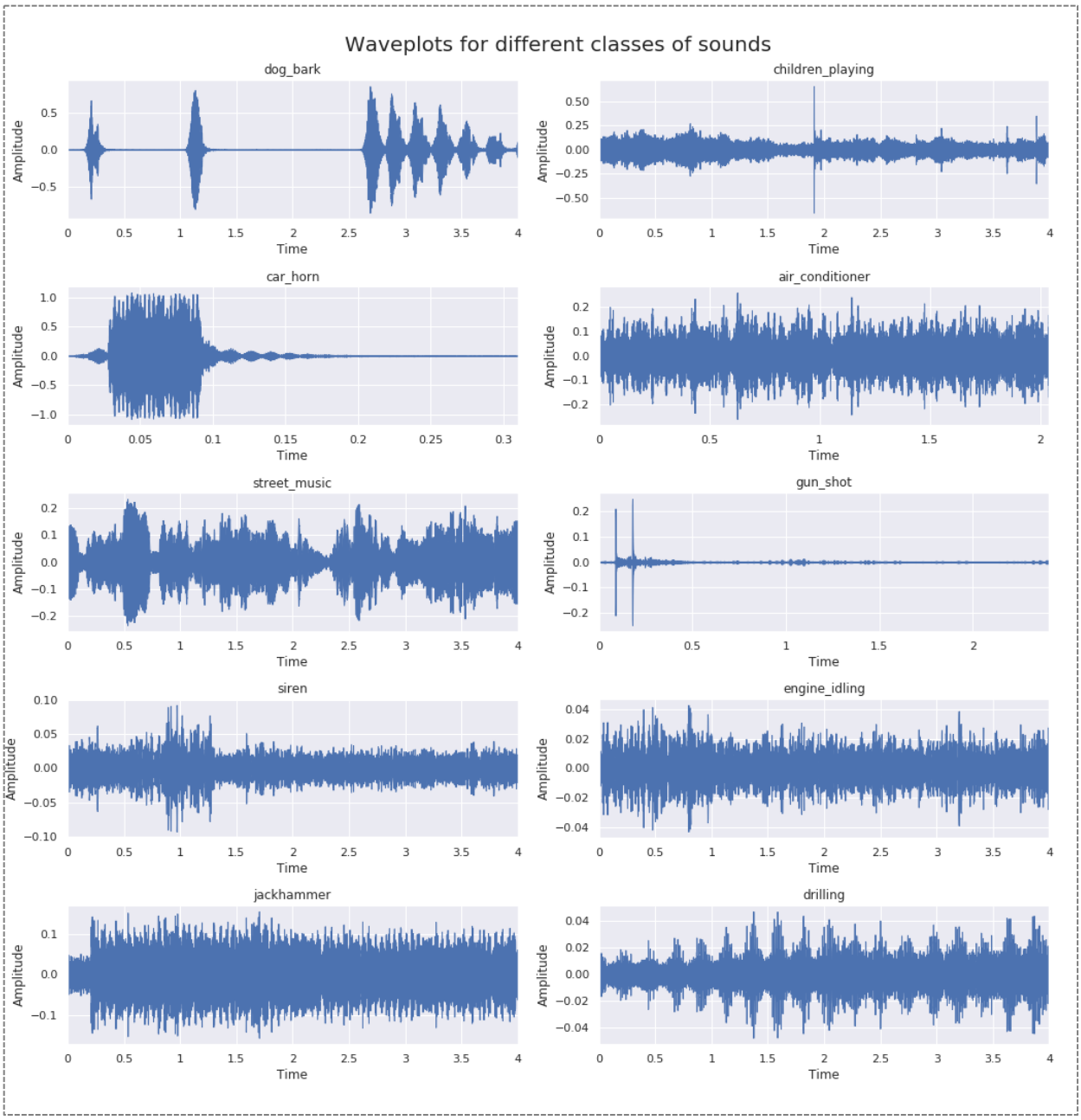

Figure 2: Wave-plots for different classes of Sound in the dataset

Another way of visualizing sound data is via spectrogram, which shows the energy in the sound signal across time. *Figure-3* shows the spectrograms of different classes. It can be seen that drilling and jackhammer have a very similar spectrogram, a similarity that was not very evident in the wave plot.

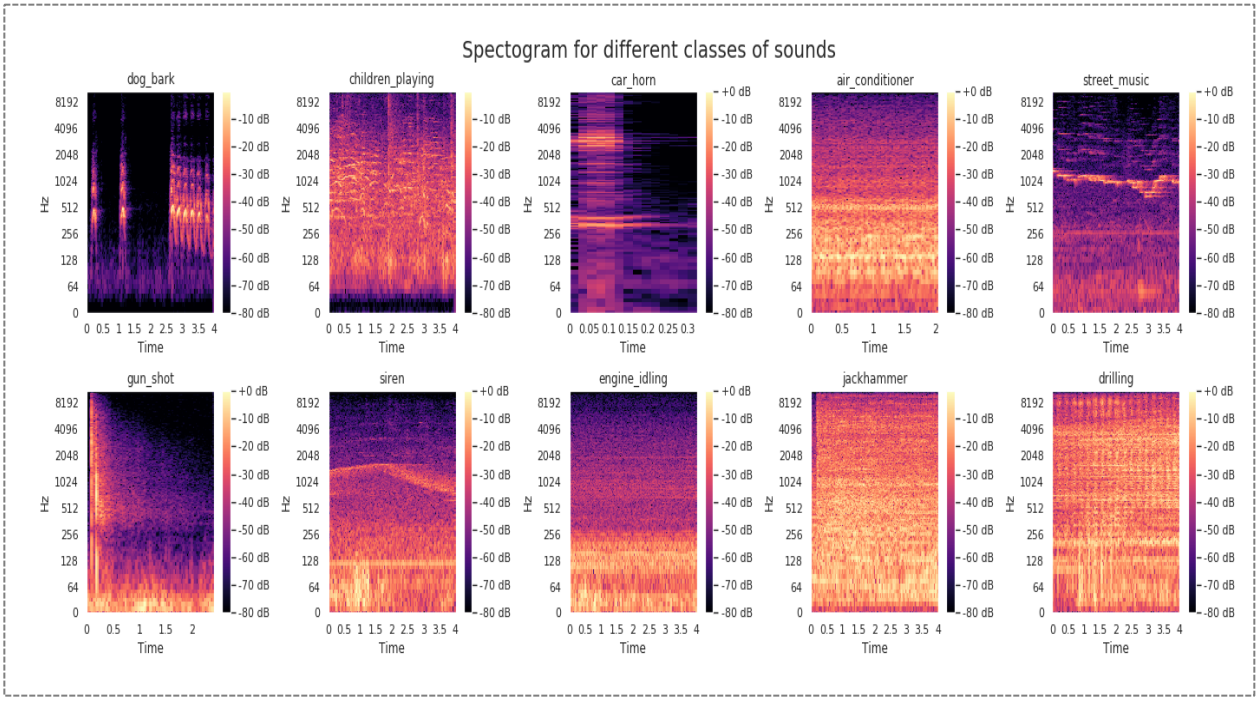

Figure 3: Spectrogram of original sounds from different classes

4.2 Feature Extraction

The sound excerpts were pre-processed before the extraction of MFCC features as the sound files had varying sampling-rate[12], bit-dept [10], and channels. Librosa [8] library was used to pre-process and generalize the data to the following metrics:

- Conversion of sampling rate to 22.05 KHz.
- Bit depth normalization into (-1,1)
- Merging of multiple audio channels into one.

Once the data was generalized, 40 MFCC features were extracted across time frames, depending on the length of the audio. The maximum time frames i.e., 173 was used as the threshold, and the samples with fewer frames were padded with zeros to make up the features of the missing time frames. A one-dimensional dataset with 40 features per sample was created by averaging out the features across all the timeframes. *Figure-4* shows the feature extraction process.

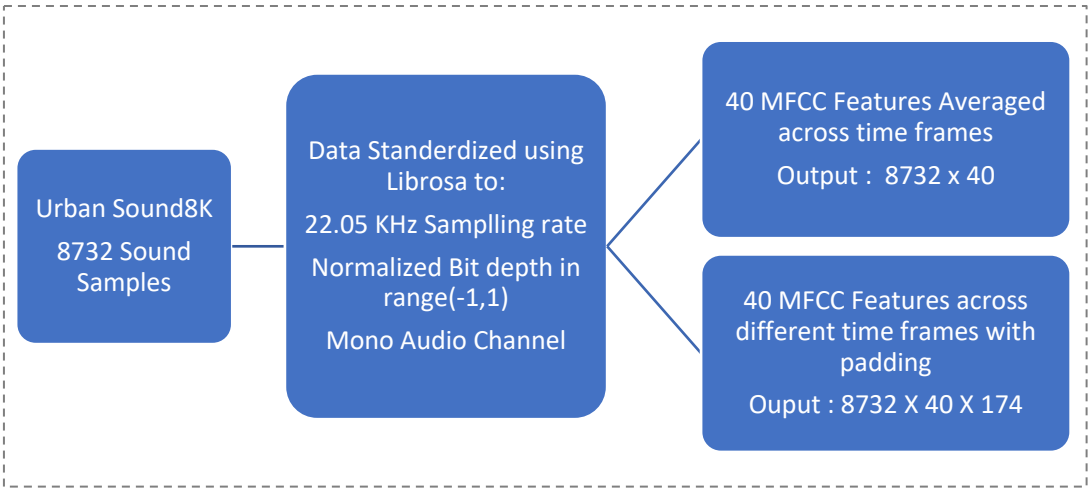

Figure 4: Flow chart for Data Pre-processing (standardization) using Librosa library, followed by MFCC feature extraction into two datasets

The MFCC features (before padding) were also visualized using a spectrogram. *Figure-5* shows the spectrogram created using the MFCC features. Based on the visual, it's evident that the MFCC has quite different features of each class of sound, even the jackhammer and drilling, which had a quite similar spectrogram earlier.

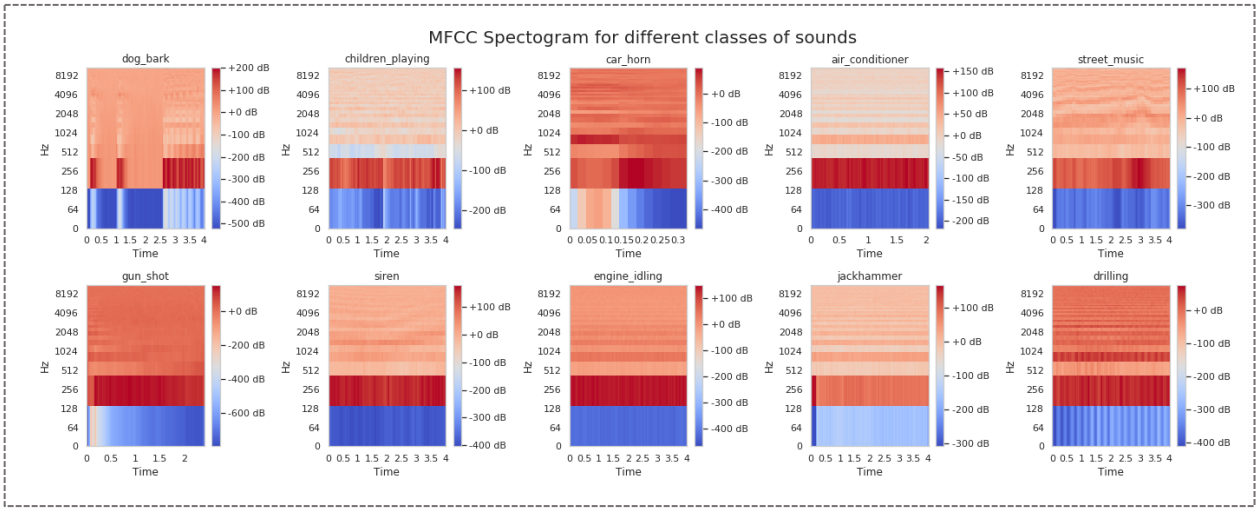

Figure 5: Spectrogram using the Mel frequency cepstral coefficients for different classes of sound

4.3 Data Splitting

For the purpose of model evaluation, data were randomly split into Training & Test data (80:20 ratio), generating 6986 training, and 1746 test samples.

5 Methods

Classification methods, namely Support-Vector-Classifier (SVC), Multi-Layer-Perceptron (MLP), and Convolutional-Neural-Networks (CNN), were used to model the classification of sound. F-1 score, and accuracy were used to evaluate model performance. *Figure-6* shows the dataset used for each model (Only the dataset corresponding to each algorithm's best performance has been reported).

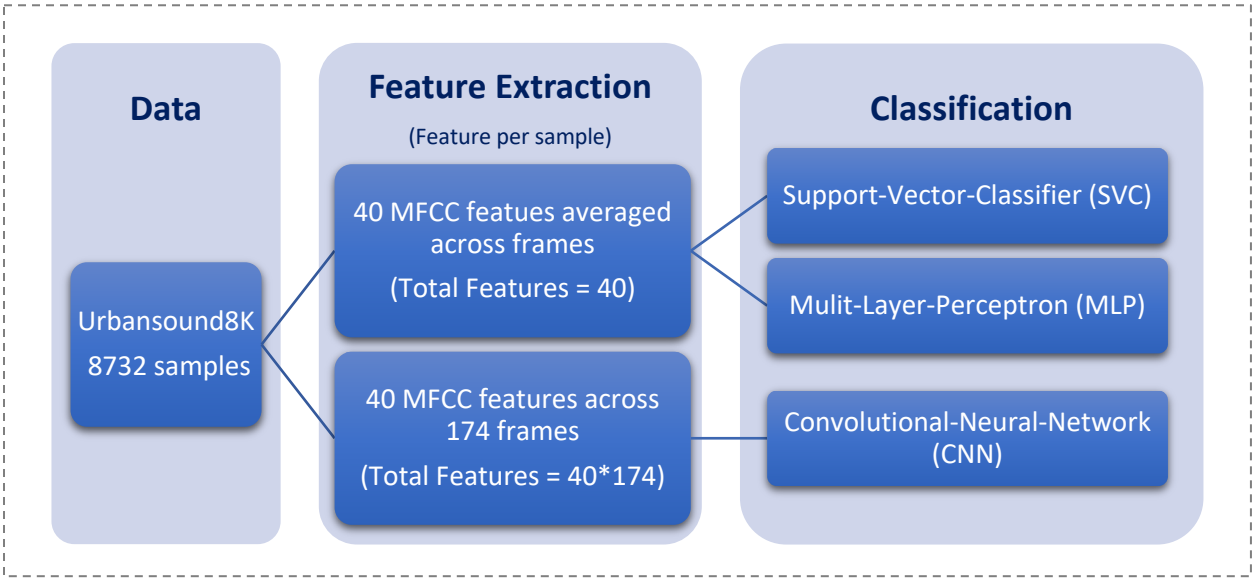

Figure 6: Flow-chart illustrating the classification algorithms used for the datasets after feature extraction

5.1 Support Vector Classifier (SVC)

An SVC classifier [9] was used as a baseline model for classification. With hyper-parameters selected using grid search to optimize performance on training data (one-dimensional data with 40 features), a classifier was trained using the below-mentioned parameters illustrated in *Table1.*

Table 1: Hyper-parameters for the SVC classifier

Parameters	Value
C (regularization parameter)	40
kernel	rbf
Gamma (kernel coefficient)	0.0001

5.2 Multi-Layer Perceptron (MLP) Classifier

To further improve the classification performance, a Multi-Layer Perceptron [14] model was used. After experimenting with several architectures, a three-layer model was used, with a batch size of 128 over 500 epochs. MLP was created using the Keras library. *Figure* 7 shows the model architecture. F*igure 8* shows the hyper-parameter selection process of the selected architecture. **Model 3**, with the highest f-1 score was selected as the final model.

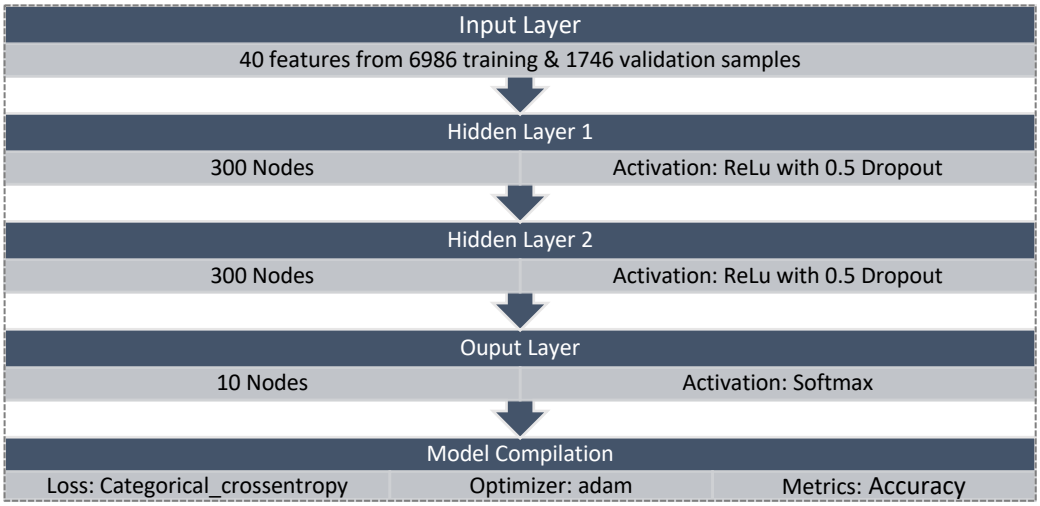

Figure 7: Multi-Layer perceptron model architecture

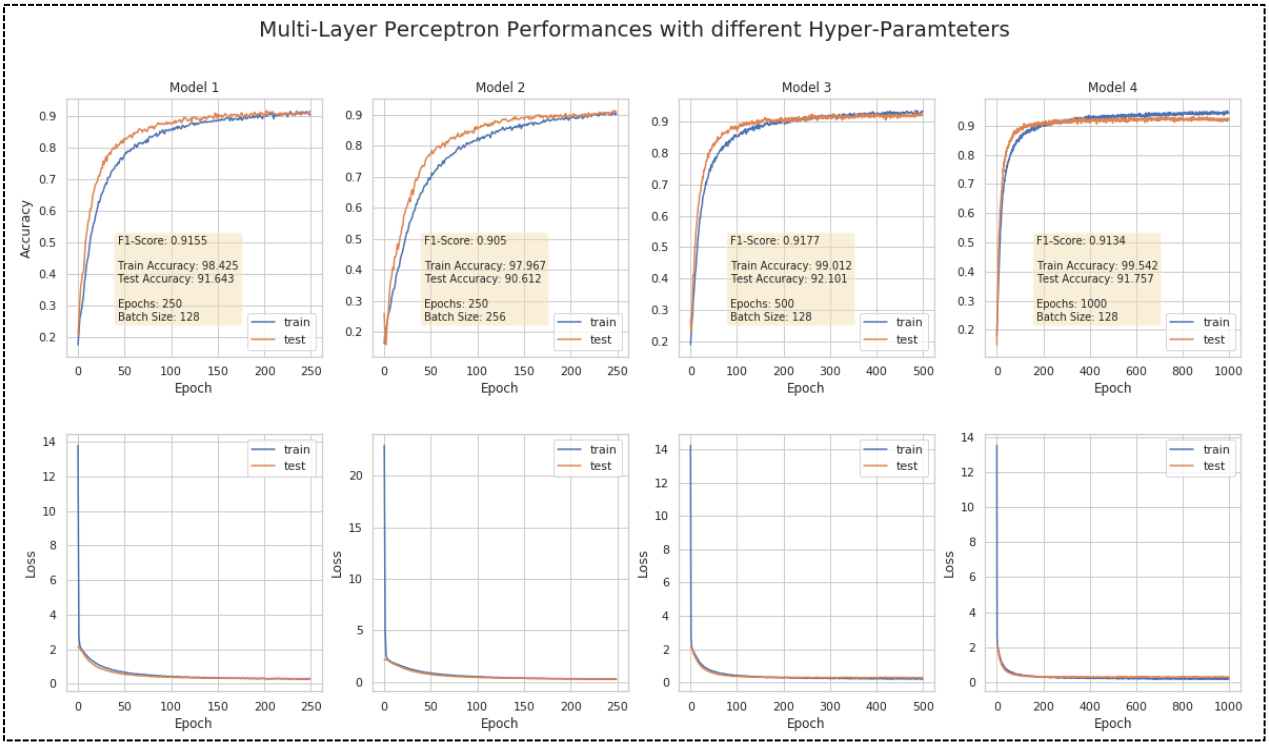

Figure 8: Hyper Parameter Selection for a Multi-Layer Perceptron Model

5.3 Convolutional Neural Network (CNN) Classifier

MFCC features originally provided a 2-dimensional feature space across time frames. The 2D data was passed through multiple convolution steps for feature extraction, followed by a multi-layer perceptron.[15] Several different architectures were experimented with to obtain the best performance. *Figure-9* shows the final model architecture with four convolutional layers and two hidden layers.

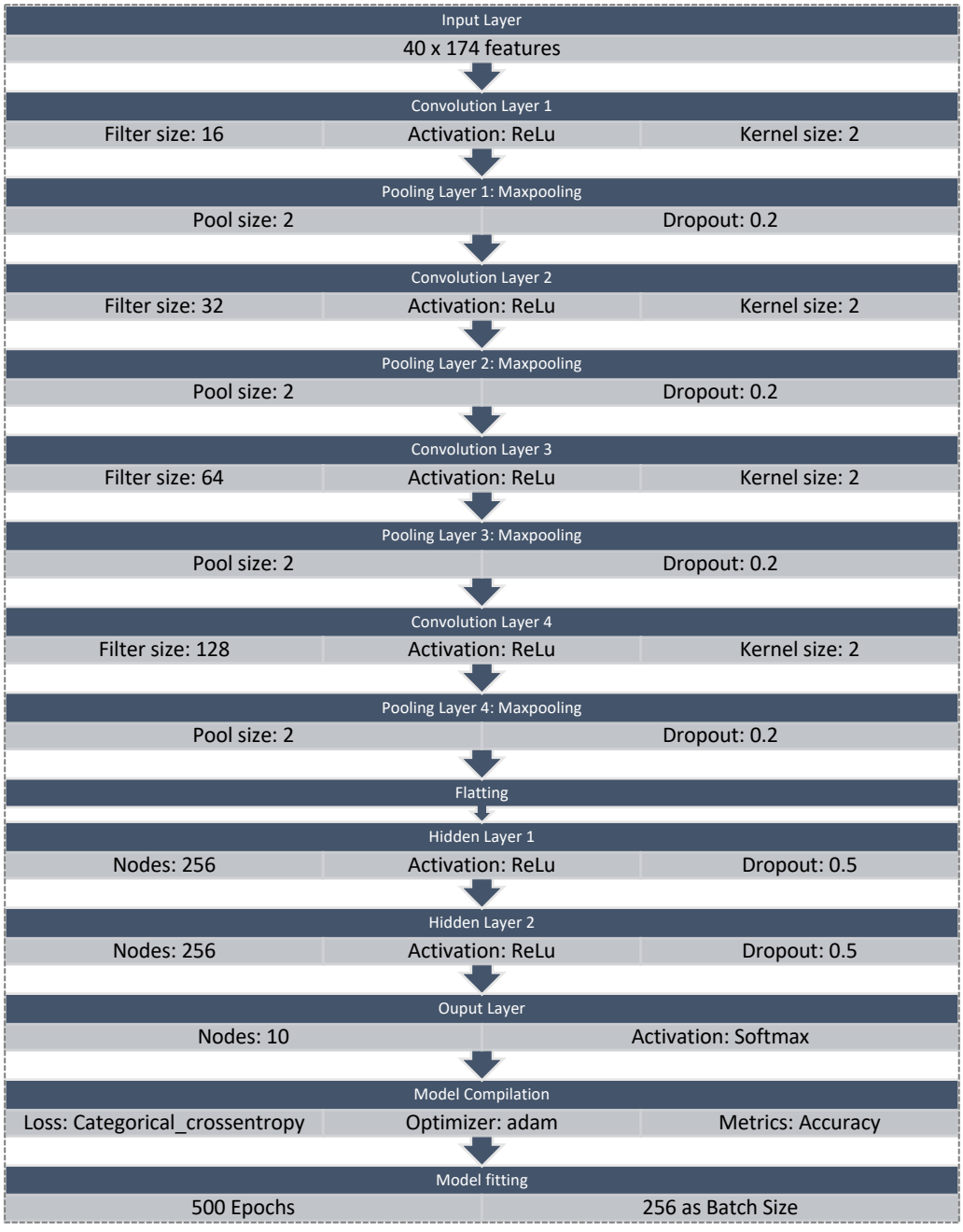

Figure 9: Architecture of the Convolutional Neural Network used for classification

Different configurations of hyper-parameters were used with the selected architecture to obtain the best performance in optimal time, and the configuration (Batch Size -256, with 500 epochs or **Model 2**) was selected as the final model. *Figure-10* illustrates the model performance with different hyper-parameters. Models were generated using the Keras library.

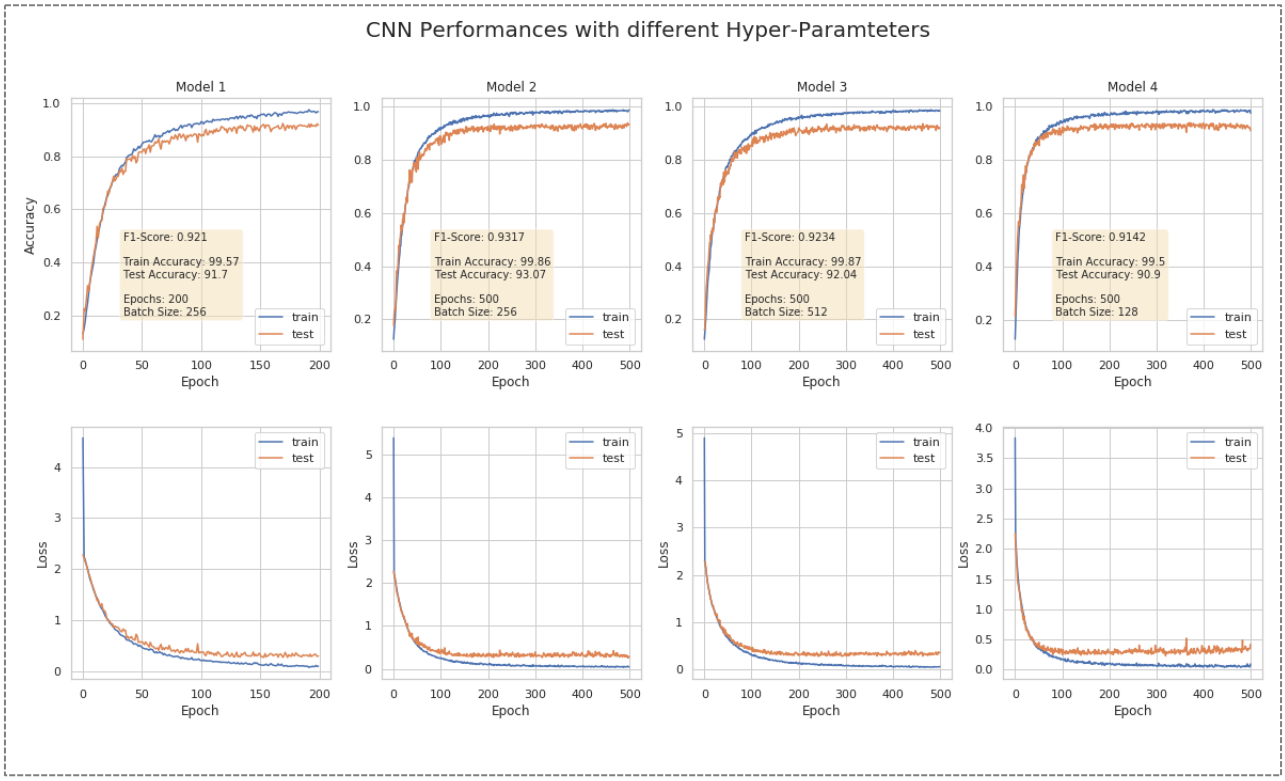

Figure 10: Hyper-Parameter selection for Convolutional Neural Network

6 Results

All the models performed extremely well, among which CNN performed the best, generating an F-1 score of 0.93, followed by MLP with 0.91, and SVC at 0.90. Models provided several other insights. Model performance was further evaluated using the train-test accuracy and confusion matrix.

SVC though performing well, appeared to have a problem with the *car_horn* class, which was one on the lesser sampled class. The misclassification could be seen in the confusion matrix shown in *Figure-11.* The misclassification could be due to overfitting of the model on the train data, with 99% accuracy on training data and 89% on the test.

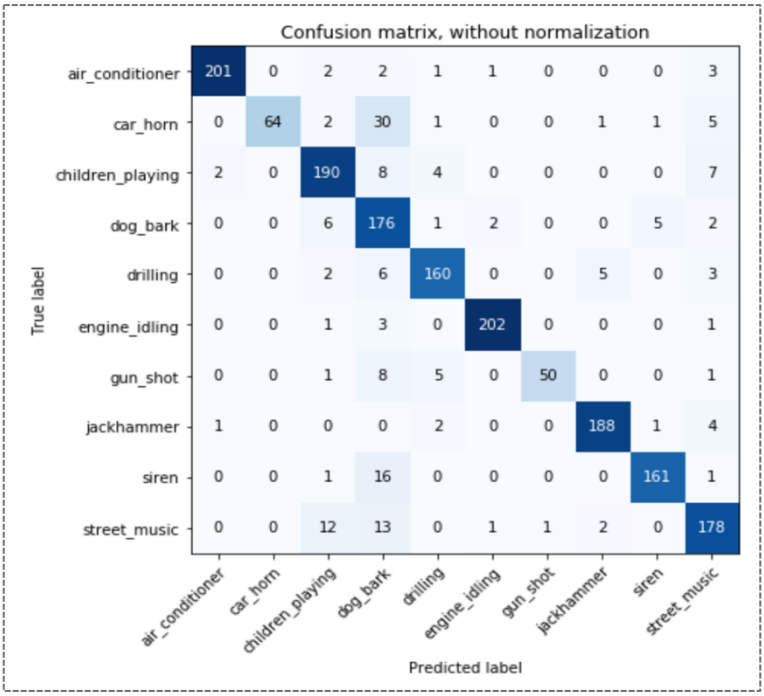

Figure 11: Confusion Matrix for Support Vector classifier over test data

The Multi-layer perceptron model certainly performed well as compared to the SVC model. However, looking at the training curves, as shown in *Figure-8,* it can be seen that the validation/test curve was ahead on the training curve, which shows the model training process had some issues. This could also be attributed to the data in the test and train split.

Finally, the CNN model which involved convolution steps for feature extraction before going into an MLP model architecture, performed very well. The learning curve in *Figure-10* shows a better learning process as compared to the standalone MLP model. *Figure-12* shows the confusion matrix based on the test data. The only major misclassification can be seen with the sound of *children_playing,* which are classified as *street_music.*

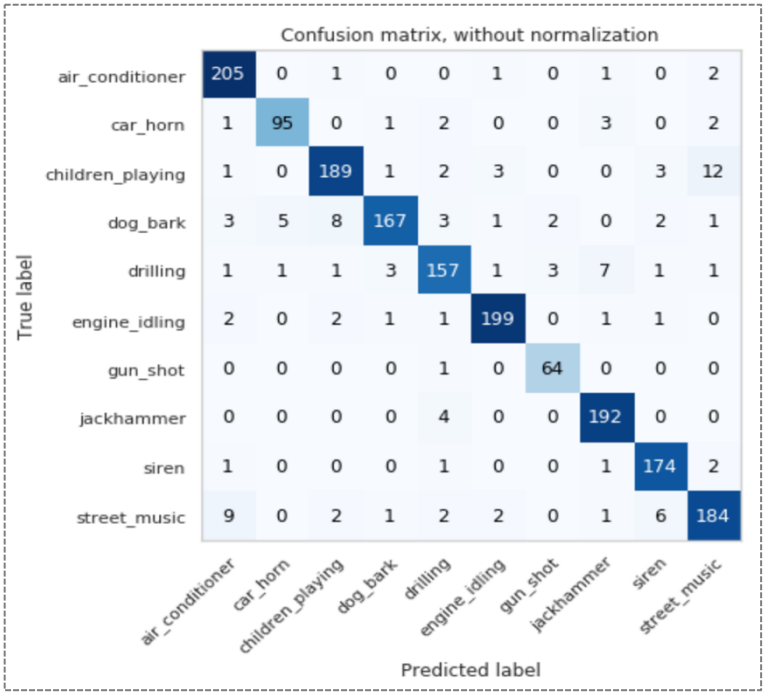

Figure 12: Confusion matrix for CNN model over Test dataset

As can be seen from *Table 2,* CNN had a much better F-1 score as compared to others, even when the SVC had a slightly more train accuracy as compared to CNN. However, the execution time for the SVC model was much faster than either of the MLP or CNN models. The performance of the model has been increased with the addition of more layers, though increasing the training time.

Table 2: Performance Metrics of Different models

	SVC	MLP	CNN
F-1 Score	0.8902	0.9177	0.9317
Training Accuracy	99.87 %	99.01 %	99.86 %
Testing Accuracy	89.86 %	92.10 %	93.07 %

7 Conclusions

Sound data is indeed an interesting avenue for classification problems. The project demonstrated the classification of sound excerpts provided by the UrbanSound8K dataset using several machine learning models. The sounds were visualized, and insights drawn from the wave-plot and spectrogram highlighted similarities between different classes. Sound files were standardized, and MFCC features were extracted, followed by data transformation in one dimension, and padding for the original two-dimensional feature space. The classification was performed using the one-dimensional MFCC features with Support-Vector-Classifier (SVC), Multi-Layer-Perceptron (MLP), and using the two-dimensional MFCC features with the Convolutional Neural Network (CNN). Model performance was evaluated using the F-1 score, Accuracy, and confusion Matrix. CNN outperformed other models with an F-1 score of 0.93 and a test accuracy of 93%, followed by MLP and then SVC.

These kinds of classification could be used as a part of the API service. Moreover, the scope of the classification could also be expanded to a greater number of classes and even speech recognition.

8 Roles

Akshay: Presented the project idea for the group, dedicated to build and tune several machine learning models including Multi-Layer Perceptron and Convolutional Neural Network (CNN), and presented the Data, Method, Model, Result, and Conclusion parts.

Angela: Extracted features from original sound data to create the MFCC graph. Dedicated to building and tuning the SVC model. Edited and finalized the introduction and background part.

Melody: Dedicated to creating the entire presentation video, wrote the abstract, created reference, and edited and finalized the introduction part.

Andrew: Wrote the draft for the Introduction of the report.

Derek: Wrote the draft for the Background of the report.

9 References

1. "UrbanSound8K." *Urban Sound Datasets*, https://urbansounddataset.weebly.com/urbansound8k.html

2. Mel-frequency cepstrum Wikipedia page https://en.wikipedia.org/wiki/Melfrequency_Cepstrum

3. Justin Salamon, Christopher Jacoby, Juan Pablo Bello "A Dataset and Taxonomy for Urban Sound Research"

4. medium.com/@mikesmales/sound-classification-using-deep-learning-8bc2aa1990b7

5. Nair, Pratheeksha. "The Dummy's Guide to MFCC." *Medium*, 27 July 2018, https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd. ---. "The Dummy's Guide to MFCC." *Medium*, 27 July 2018, https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd.

6. Piczak, Karol J. "Environmental Sound Classification with Convolutional Neural Networks." *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2015, pp. 1–6. *DOI.org (Crossref)*, doi:10.1109/MLSP.2015.7324337.

7. Abdoli, Sajjad, et al. "End-to-End Environmental Sound Classification Using a 1D Convolutional Neural Network." *Expert Systems with Applications*, vol. 136, Dec. 2019, pp. 252–63. *DOI.org (Crossref)*, doi:10.1016/j.eswa.2019.06.040.

8. *Librosa*. https://librosa.github.io/

9. *Support Vector Machines — Scikit-Learn 0.22.2 Documentation*. https://scikit-learn.org/stable/modules/svm.html. Accessed 18 Apr. 2020.

10. MicroPyramid. "Understanding Audio Quality: Bit Rate, Sample Rate." *Medium*, 4 Oct. 2017, https://medium.com/@MicroPyramid/understanding-audio-quality-bit-rate-sample-rate-14286953d71f.

11. https://medium.com/@mikesmales/sound-classification-using-deep-learning-8bc2aa1990b7

12. "Audio Data Analysis Using Deep Learning with Python (Part 1)." *KDnuggets*, https://www.kdnuggets.com/audio-data-analysis-using-deep-learning-with-python-part-1.html/

13. *Algorithms On Sound Data*. http://archive.oreilly.com/oreillyschool/courses/data-structures-algorithms/soundFiles.html

14. Kain, Nitin Kumar. "Understanding of Multilayer Perceptron (MLP)." *Medium*, 2 Dec. 2019, https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f.

15. Eijaz Allibhai Building a Convolutional Neural Network (CNN) in Keras. https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5

16. Daphne Cornelisse An intuitive guide to Convolutional Neural Networks. https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

17. https://keras.io

18. *Practical Cryptography*. http://practicalcryptography.com/miscellaneous/machinelearning/guide-mel-frequency-cepstral-coefficients-mfccs/

19. Xu, Min, et al. "Audio Keyword Generation for Sports Video Analysis." *Proceedings of the 12th Annual ACM International Conference on Multimedia - MULTIMEDIA '04*, ACM Press, 2004, p. 758. *DOI.org (Crossref)*, doi:10.1145/1027527.1027702.

20. https://towardsdatascience.com/sound-classification-using-images-68d4770df426