Urban Sound Classification with Neural Networks

Fatma Er, 967585

May 31, 2022

Abstract

Automatic urban sound classification is a growing area of research with applications in multimedia retrieval and urban informatics.There are many sounds around us and our brain can easily and clearly identify them. In addition, our brain constantly processes the received sound signals and provides us with relevant environmental information. Although not at the level of accuracy of the brain, necessary information can be extracted from an audio signal with the help of different algorithms.In this study, we aimed to develop a scientific and consistent model based on machine learning algorithms to help urban sound classification.For this reason, we used the UrbanSound8K dataset containing 8732 labeled sound audios($\leq$ 4s).Mel Spectrogram (MEL), Mel-Frequency Cepstral Coefficient (MFCC) and Short Time Fourier Transform (Chroma_STFT), three of the most commonly used features extraction in sound processing, are used.After feature extraction, Convolutional Neural Networks (CNN), Deep Neural Network (DNN) and Long-Short-Term Memory (LSTM) models were used for classification.The results obtained from the models were compared and tested.The best result are obtained from the CNN model with the success of 56% among the developed models.Reporting and interpretation of the mean accuracy and standard deviation obtained in the test folds were made.

Key words: Urbansound, Classification, Neural Networks ,MFCC,Mel Spectrogram,Chroma_STFT, CNN, DNN, LTSM

1 Introduction

Living in a world surrounded by different sound forms from different sources, our brain and auditory system constantly defines every sound it hears (Chachada and Kuo, 2013).

Environmental sound classification is one of the important issues in the audio recognition field. There are various audio effects in daily life and multimedia materials, such as car-horn, bell-ringing, and laughter. These special effects play important roles in humans' understanding of the high level semantics of the auditory context (Cai et al., 2006).

Recognizing environmental sounds is a basic audio signal processing problem(Chu et al., 2009).Audio classification is regarded as a great challenge in pattern recognition. Sound provides us with rich information about its producer and environment(Zeng et al., 2019).

The problem of automatic environmental sound classification has received increasing attention from the research community in recent years. Its applications range from context aware computing and surveillance to noise mitigation enabled by smart acoustic sensor networks(Salamon and Bello, 2016).

Sonic analysis of urban environments has aroused more and more interests recently. Unlike visual images, the urban sound is less constructed and full of noises due to complex acoustic scenes in real life. Thus, it is far more challenging to identify and classify these sound sources from different environments(Yang et al., 2019).

Researchers can take advantages of the background sounds classification and then separate the speech signals from different background noises for the advanced research Therefore, it is of great value for researchers to investigate in this field (Barchiesi et al., 2015).

While obtaining the kepstrum coefficients used as the feature vector, MFCC is generally used in speaker recognition applications. This is because MFCC imitates the frequency selectivity of the human ear, achieving good speaker discriminating values. In addition, the MFCC coefficients are much less affected by the changes and the sound wave structure(Furui, 1996).

There are three different stages that are attached with the classification of sound. These are pre-processing of the audio signal, specific feature extraction from that signal and finally the classification of the audio signal.

In this project, we have developed several deep learning algorithms such as CNN, DNN and LSTM to the sound classification task. The input to our models are short urban sound audios. By extracting three important features, namely the Melspectrogram (Mel) , Mel-frequency Cepstral Coefficients (MFCC) and Short Time Fourier Transform (Chroma_STFT), we trained the models with the above feature vectors and output the predicted category label that a specific sound audio belonged to. We compared the performance of multiple algorithms with different features and explored why several models achieved better performance.

The paper is organized as follows: Section II describes the related work and background acknowledge about urban sound classification problem and Section III introduces the dataset and features we are using in this project. The methods, models and model's average accuracy and standard deviation are presented in section IV, followed by the section V presenting the results and discussion details about each model. Finally we draw some perspectives about the future work in section VI.

2 RELATED WORKS

In this section, the studies published in the literature on sound classification are examined.

(Salamon and Bello, 2016) proposed a model on environmental sound classification. In the study, They evaluated the data of 8732 sound clips of up to 4 s in duration taken from field recordings of the UrbanSound8K. The clips span 10 environmental sound classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling,gun shot, jackhammer, siren and street music. They evaluated the proposed CNN (convolutional neural network) architecture and the influence of the different augmentation sets.They saw that the proposed SB-CNN performs comparably to SKM and PiczakCNN when training on the original dataset without augmentation (mean accuracy of 0.74, 0.73 and 0.73 for SKM, PiczakCNN and SB-CNN, respectively). They informed the original dataset is not large/varied enough for the convolutional model to outperform the "shallow" SKM approach. However, once they increased the size/variance in the dataset by means of the proposed augmentations, the performance of the proposed model increases significantly, yielding a mean accuracy of 79 %

(Mahmood and Köse, 2021) proposed a model on proposed an automatic speech recognition system based on two algorithms for features extraction. In the study, They evaluated the data of 64,727 one-second audio clips of 30 short words from Google with the goal of collecting single-word commands (rather than words as said and used in conversation).A group of 20 core words audio files were recorded and repeated 5 times by the most of speakers. The core words consist of (Yes, No, Down,up, right, left, off, on, go, stop) and the numbers zero through nine. Auxiliary words consist of "Bed", "Bird", "Cat", "Dog", "Happy", "House", "Marvin", "Sheila", "Tree" and "Wow". They evaluated the proposed the unique features of the speech acoustic signal has been extracted using MFCCs and feed theses features for the CNN algorithm for further features learning and classification.The Study compared three model architectures of the CNN in order to see the best results of the model with various options of hyperparameters and layers. They showed that the using of MFCC as a feature extraction and feed these features for the CNN model for further features extraction will improve the accuracy and reduce the complexity of the model. They evaluated 3 models with different hyperparameters configuration in order to choose the best model with higher accuracy. The highest accuracy achieved is 88.21%.

(Gunther et al., 2019)proposed a model on environmental sound classification. In the study, They evaluated the data of 8732 sound clips of up to 4 s in duration taken from field recordings of the UrbanSound8K. They offered, The data set is already shuffled and separated into 10 folds for a 10 fold cross validation and the authors of the data recommend not reshuffling the data during training due to the distribution of the sound classes within each fold.They used the Librosa audio package in python to extract features from the audio files in our dataset. The following were the features that they extracted: (1) Mel-Frequency Cepstral Coefficients (MFCC): Coefficients derived from a cepstral representation of the audio clip,(2) Chromagram: Pitch class profiles. They capture harmonic and melodic characteristics within the music, (3) Mel-scaled spectrogram: Psychoacoustic scales that capture the distances from low to high scale frequency, (4) Contrast: Difference between parts of a sound or different instrument sounds, (5) Spectral Contrast: Represents the strength of spectral peaks and valleys in each a sub-band as contrast distribution, (6) Tonnetz: Representation of tonal space. Of the methods implemented in their project, the CNN (73.4%) performed the best, the DNN(68.3%) was the second best, and the KNN (55.6%,K-NN (K-Nearest Neighbor)) came in last place.

(Yang et al., 2019) proposed a model on environmental sound classification. In the study, They evaluated the data of 8732 sound clips of up to 4 s in duration taken from field recordings of the UrbanSound8K. They extracted two features that are most frequently used in audio processing area, namely Melspectrogram and MFCC by using the Python library librosa.They developed several deep learning algorithms such as CNN, DNN, LSTM and other modern neural networks like VGG and Resnet to the sound classification task. In this project, they have built several deep learning models such as CNN (MFCC,0.87;Mel,0.88), DNN (MFCC,0.90;Mel,0.88), LSTM (MFCC,0.89;Mel,0.92) to explore how they performed in classifying the urban sound sources with the extracted Mel and MFCC features. Based on their experiments, LSTM have achieved much better performances in test data than CNN and DNN, which met their expectation. The audio signals are highly timerelated and meanwhile LSTM preserved the influence of the previous status, thus it has higher accuracy than the other two. Furthermore, different features may also have different performance in the classification task. For instance, the Melspecrogram behaves better in LSTM and DNN while MFCC behaves better in CNN, VGGNet (MFCC,0.96;Mel,0.95) and ResNet (MFCC,0.98;Mel,0.96). Meanwhile, their experiments showed that VGGNet11 and ResNet18 have very excellent performance on sound classification tasks.

(Das et al., 2020) proposed a model on environmental sound classification. They used same dataset, which is UrbanSound8K. They worked with of the features which are the following: MFCC, Mel Spectrogram, Chroma STFT, Chroma CQT, Chroma CENS, Spectral Contrast, Tonnetz as they provided distinguishable information and representations that are effective in classifying audio signals more accurately.They performed the classification via two different models which are CNN and LSTM.They showed, MFCC was the best feature showing the highest accuracy for both models compared to other features. Considering other approaches to research, they pooled different features to find out how the classification accuracy of the model changes accordingly.Some of the stacking worked significantly well in improving the performance of their models, for example stacking the MFCC and Chroma STFT helped them achieve a verification accuracy of 98.81%, the best performance to date exceeding 98.60%.The best results among the models used in the study; CNN (MFCC ,96.78 ;MFCC, Chroma STFT,95;90) and LSTM (MFCC,98.23; MFCC, Chroma LTFT,98.81).

(Lezhenin et al., 2020)proposed a model on environmental sound classification. They used same dataset, which is UrbanSound8K.They performed the classification via two different models which are CNN and LSTM.They proved that ,both models show the similar performance, While CNN provides 81.67% average accuracy, the proposed LSTM network achieves 84.25%. They specifiedThe two models outperform the baseline methods.

3 DATASET AND FEATURES

3.1 The Urban Sound Dataset

In this project, we used the UrbanSound8K dataset containing 8732 labeled sound audios($\leq$ 4s). These audios come from 10 urban sound classes including air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. This dataset contains 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren, and street_music. The classes are drawn from the urban sound taxonomy described in the "A Dataset and Taxonomy for Urban Sound Research" article, which also includes a detailed description of the data-set and how it was compiled (Salamon et al., 2014).

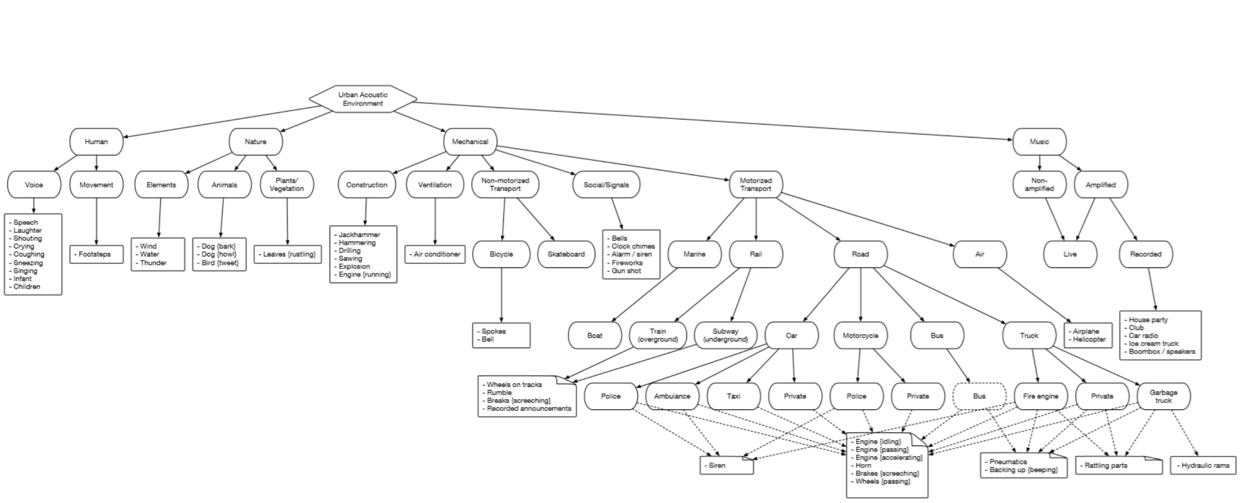
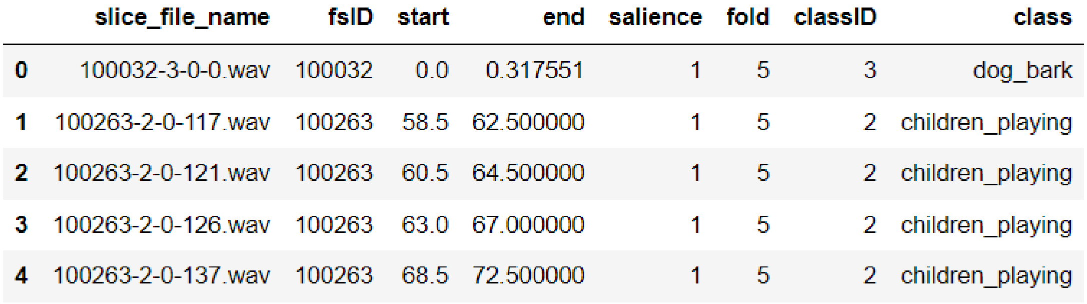

Urban Acoustic Environment
Human
Nature
Mechanical
Music
Voice
Movement
Elements
Animals
Plants/ Vegetation
Construction
Ventilation
Non-motorized Transport
Social/Signals
Motorized Transport
Non-amplified
Amplified
- Speech - Laughter - Shouting - Crying - Coughing - Sneezing - Singing - Infant - Children
- Footsteps
- Wind - Water - Thunder
- Dog {bark} - Dog {howl} - Bird {tweet}
- Leaves {rustling}
- Jackhammer - Hammering - Drilling - Sawing - Explosion - Engine {running}
- Air conditioner
Bicycle
Skateboard
- Bells - Clock chimes - Alarm / siren - Fireworks - Gun shot
Marine
Rail
Road
Air
Live
Recorded
- Spokes - Bell
Boat
Train (overground)
Subway (underground)
Car
Motorcycle
Bus
Truck
- Airplane - Helicopter
- House party - Club - Car radio - Ice cream truck - Boombox / speakers
- Wheels on tracks - Rumble - Breaks {screeching} - Recorded announcements
Police
Ambulance
Taxi
Private
Police
Private
Bus
Fire engine
Private
Garbage truck
- Siren
- Engine {idling} - Engine {passing} - Engine {accelerating} - Horn - Brakes {screeching} - Wheels {passing}
- Pneumatics - Backing up {beeping}
- Rattling parts
- Hydraulic rams

Figure 1: Urban Sound Taxonomy

The dataset,consists of two parts: Audio and Metadata

Audio : It has 10 sub-folders named ‘ fold1 ’ through ‘ fold10 ’. Each sub-folder contains several ‘ .wav ’ audio samples eg. ‘ fold2/14387-9-0-15.wav ’ The files are pre-sorted into ten folds (folders named fold1-fold10) to help in the reproduction of and comparison with the automatic classification results.

Metadata : 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, and so on.



	slice_file_name	fsID	start	end	salience	fold	classID	class
0	100032-3-0-0.wav	100032	0.0	0.317551	1	5	3	dog_bark
1	100263-2-0-117.wav	100263	58.5	62.500000	1	5	2	children_playing
2	100263-2-0-121.wav	100263	60.5	64.500000	1	5	2	children_playing
3	100263-2-0-126.wav	100263	63.0	67.000000	1	5	2	children_playing
4	100263-2-0-137.wav	100263	68.5	72.500000	1	5	2	children_playing

Figure 2: UrbanSound8K Metadata Description

As we can see from Figure 6, we will use this file for labeling data.

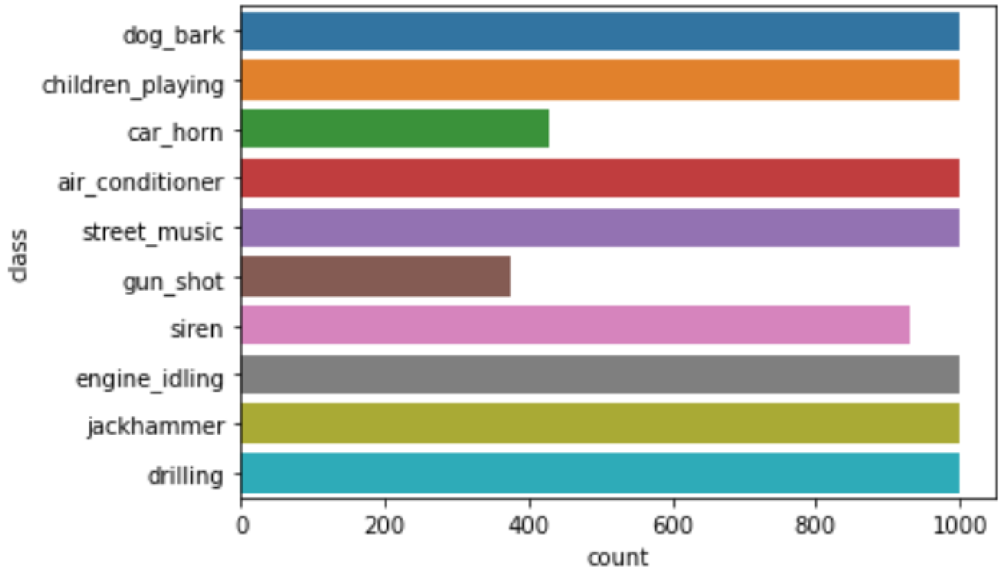


Figure 3: Distribution of Sound Classes

Figure 3 shows, the class labels to which each sound sample belongs and the number of sound recordings of the classes.

Respectively, we have 429 voice record for car horn, 374 for gun shot, 929 for siren and 1000 each for other sound recordings.

Figure 4 shows all voice record's distribution.Figure 5 and 6 show an example voice record's array and distribution. The example voice record is gunshot.

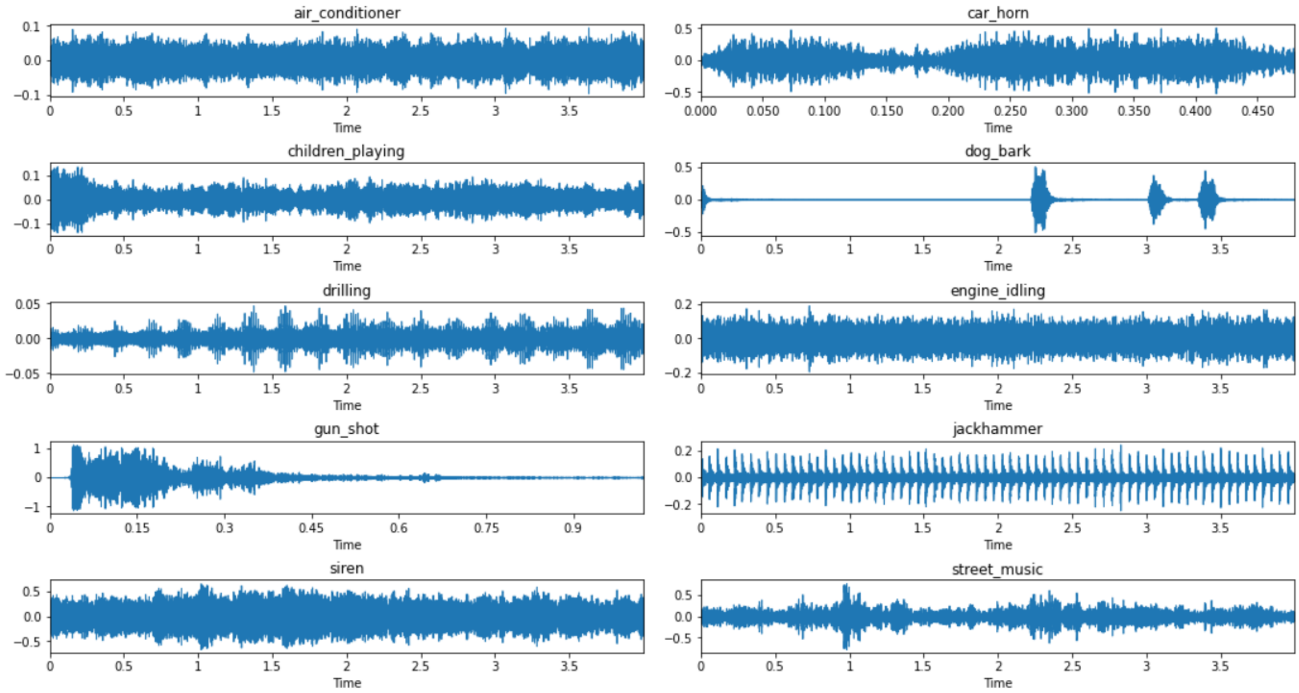
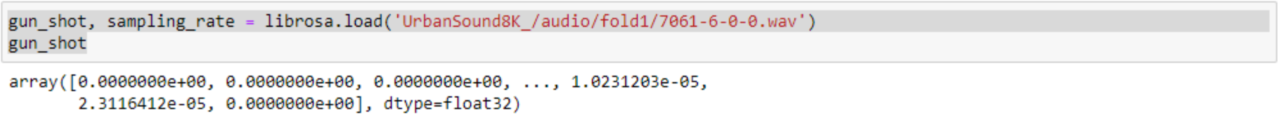


Figure 4: All audio records distribution(.waw)



```
gun_shot, sampling_rate = librosa.load('UrbanSound8K_/audio/fold1/7061-6-0-0.wav')
gun_shot
```

```
array([0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 1.0231203e-05,
       2.3116412e-05, 0.0000000e+00], dtype=float32)
```

Figure 5: Example array for sound recording(.waw), Gunshot

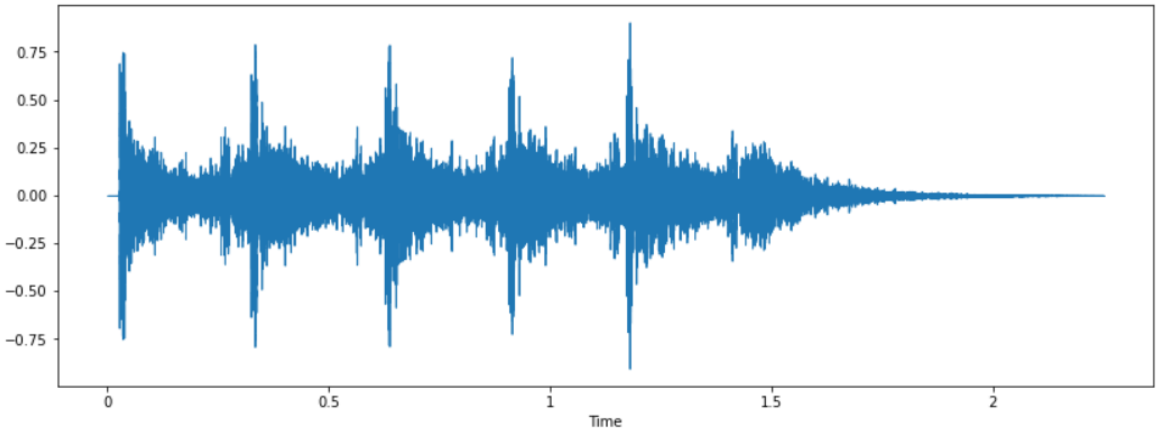


Figure 6: Example image for sound recording(.waw), Gunshot

3.2 FEATURE EXTRACTION

The audio file cannot directly enter the model. We need to load the audio data from the file and render it in a format the model expects.

Feature extraction lessens the size of information and helps in the representation of the data as the feature vectors(Das et al., 2020).

We usually convert audio data from time domain to frequency domain, then use spectrogram features as input to machine learning models. Here we extracted Mel Spectrogram (MEL), Mel-Frequency Cepstral Coefficient (MFCC) and Short-Time Fourier Transform (Chroma_STFT) , three of the most frequently used features in audio processing, using the Python library librosa.After analyzing the relevant studies, we decided on these three methods. These three methods are the most common for classification of sound.

1.Mel-Frequency Cepstral Coefficient (MFCC)

The Mel frequency cepstral coefficients, which are compact representations of the spectrum, are typically used to automatically identify speech and are also used as a primary feature in many research fields involving audio signals. Davis and Mermelstein introduced them in the 1980s and have been state-of-the-art ever since (Davis and Mermelstein, 1980).

Mel Frequency Cepstral Coefficents (MFCCs , MFCC) is a way of extracting features from an audio. The MFCC uses the MEL scale to divide the frequency band to sub-bands and then extracts the Cepstral Coefficents using Discrete Cosine Transform (DCT). MEL scale is based on the way humans distinguish between frequencies which makes it very convenient to process sounds (ILM).

MFCCs has been the most common approach for the last years as it has proved to perform much better than normal spectrograms (Rajo).

There are five fundamental steps used to extract mfcc features from a one dimensional signal(Magre et al., 2014).

There are five fundamental steps used to extract mfcc features from a one dimensional signal.These are framing and blocking, windowing, fft, mel-scale, inverse-fft shown in figure 7.

1)Frame the signal into short frames. 2)For each frame calculate the periodogram estimate of the power spectrum. 3)Apply the mel filterbank to the power spectra, sum the energy in each filter. 4)Take the logarithm of all filterbank energies. 5)Take the DCT of the log filterbank energies. 6)Keep DCT coefficients 2-13, discard the rest.

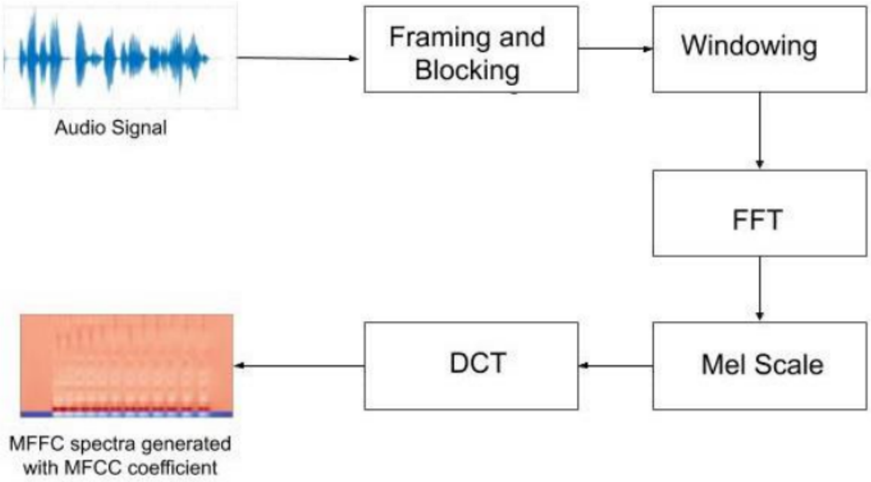


Figure 7: Steps to extract MFCCs from an audio signal

MFCCs – The MFCC summarizes the frequency distribution across the window size. So, it is possible to analyze both the frequency and time characteristics of the sound. This audio representation will allow us to identify features for classification. So, it will try to convert audio into some kind of features based on time and frequency characteristics that will help us to do classification (Agrawal).

Figure 8 shows the MFCC values for 5 randomly selected classes in the metadata and Figure 9 shows MFCC for gunshot sound.

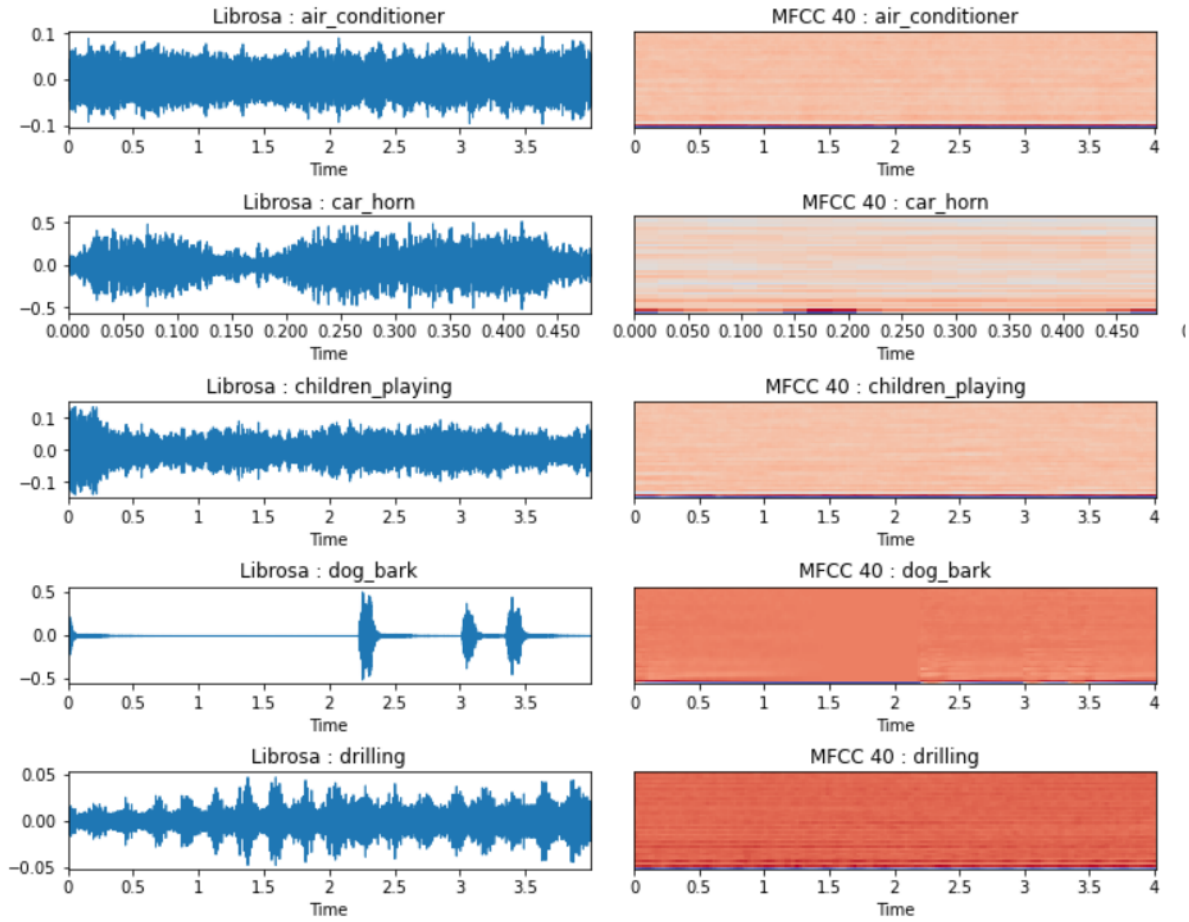


Figure 8: Extract MFCC from 5 audio signal distrubition

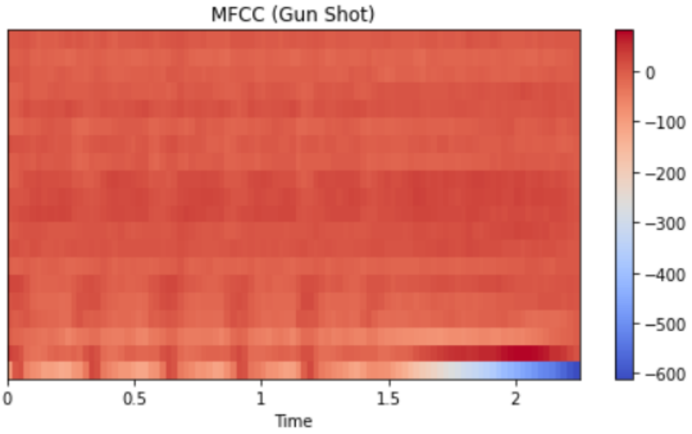


Figure 9: Example extract MFCC from sound recording(.waw), Gunshot

2.Mel Spectrogram (MEL)

Psychoacoustic scales that capture the distances from low to high scale frequency (Gunther et al., 2019).

A Mel Spectrogram is a collaboration of the Mel scale and a spectrogram. Here, the mel scale represents the nonlinear transformation of the frequency scale. The steps required to generate the Mel features are somewhat similar to those required for the MFCC coefficient (Qiao et al., 2019). Figure 10 shows the Mel Spectrogram's steps.

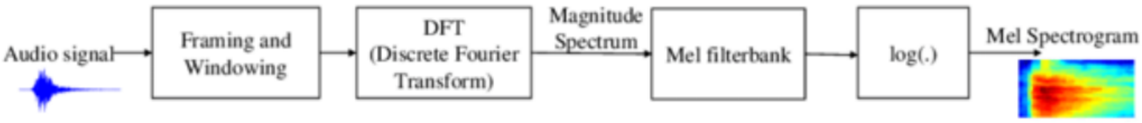


Figure 10: Steps required to extract Mel spectrogram

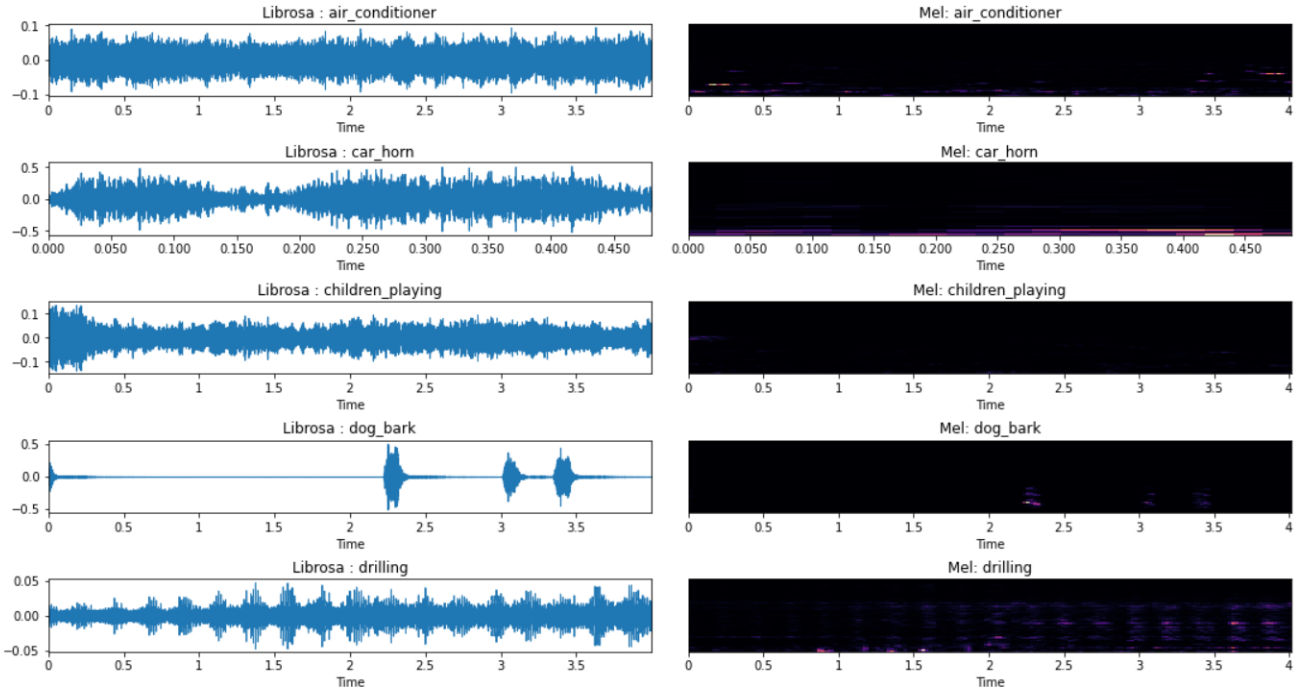


Figure 11: Extract Mel Spectrogram from 5 audio signal distributions

Figure 11 shows the Mel Spectrogram values for 5 randomly selected classes in the metadata and Figure 12 shows Mel Spectrogram for gunshot sound.

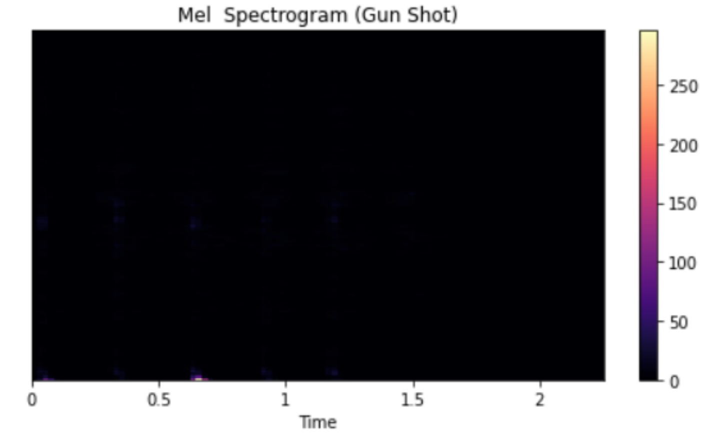


Figure 12: Example extract Mel Spectrogram from sound recording(.waw), Gunshot

3.Short-Time Fourier Transform (Chroma_STFT)

Chromagram is another feature technique for extracting music audio signals. Chroma-based features are particularly useful for pitch analysis. The chroma notation tells us the intensity of the 12 different musical chroma of the octave in each time frame. They can be used to differentiate pitch class profiles between audio signals (Shah et al., 2019).We used chroma STFT as the feature for our model.

The Fourier transform, which is used to convert a time-dependent signal to a frequency-dependent signal, is one of the most important mathematical tools in audio signal processing. Applying the Fourier transform to local sections of an audio signal, one obtains the short-time Fourier transform STFT (Müller, 2017).

Chromagrama STFT represents information about classification of pitch and signal structure.

Figure 13 shows the Chroma_STFT values for 5 randomly selected classes in the metadata and Figure 14 shows Chroma_STFT for gunshot sound.

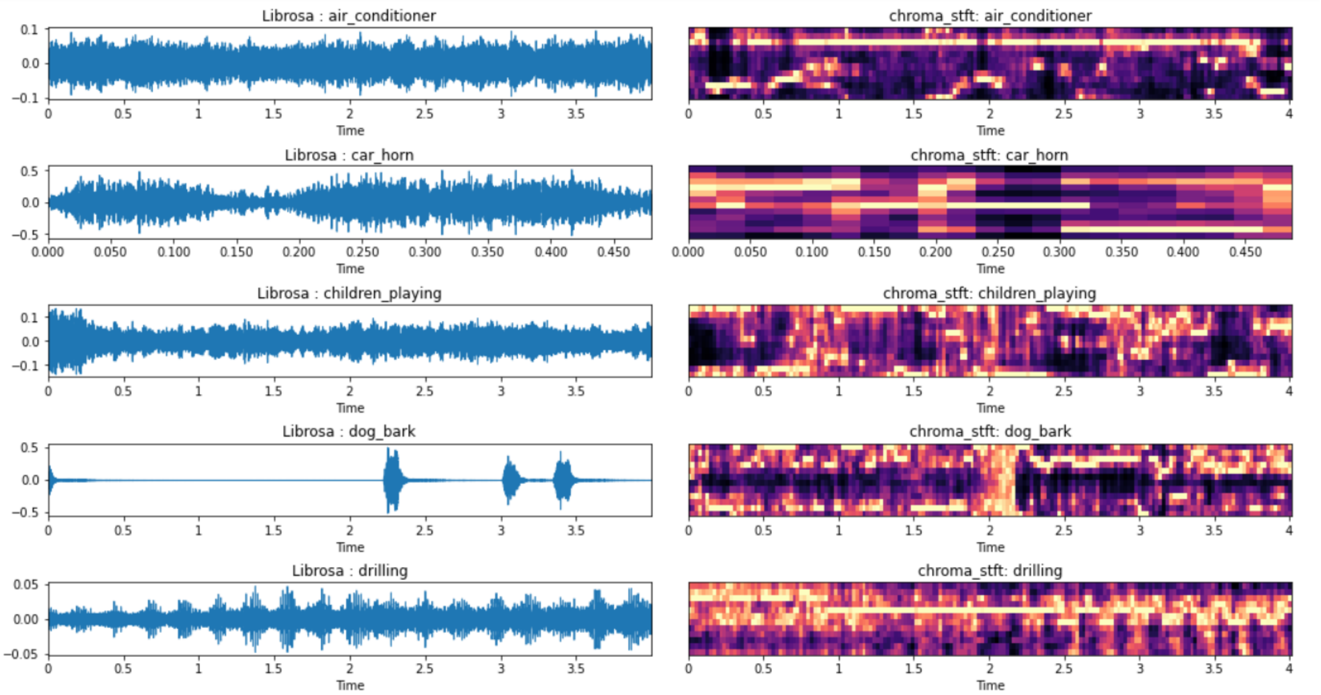


Figure 13: Extract Chroma_STFT from 5 audio signal distributions

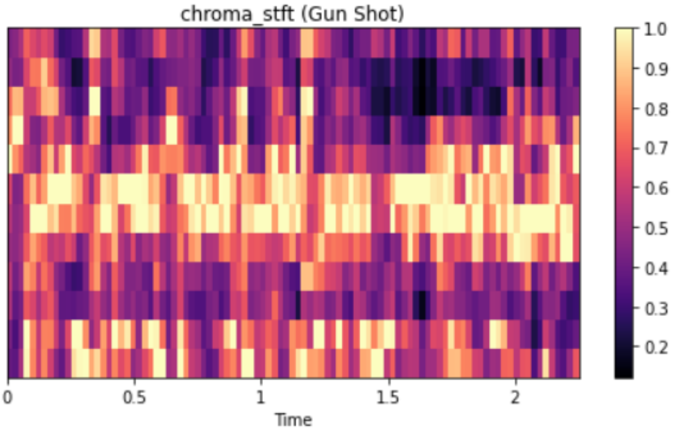


Figure 14: Example extract Chroma_STFT from sound recording(.waw), Gunshot

4 METHODS

This section describes Convolutional Neural Networks (CNN), Deep Neural Network (DNN), and Long-Short-Term Memory (LSTM).

1. Convolutional Neural Networks(CNN)

The CNN model is a method that has proven to be extremely effective in learning deep architectures.It is a neural structure consisting of several layers, each of which consists of the majority of individual neurons. With more discreet constructions, CNN is incredibly productive at learning theoretical emphases. Weight sharing, spatial inspection, and close association are three key features of CNN.To design the model, CNN provides the information of all the integrals in the same order, but with an incredible computational measure(Yang et al., 2019).

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

1)Convolutional layer

2)Pooling layer

3)Fully-connected (FC) layer

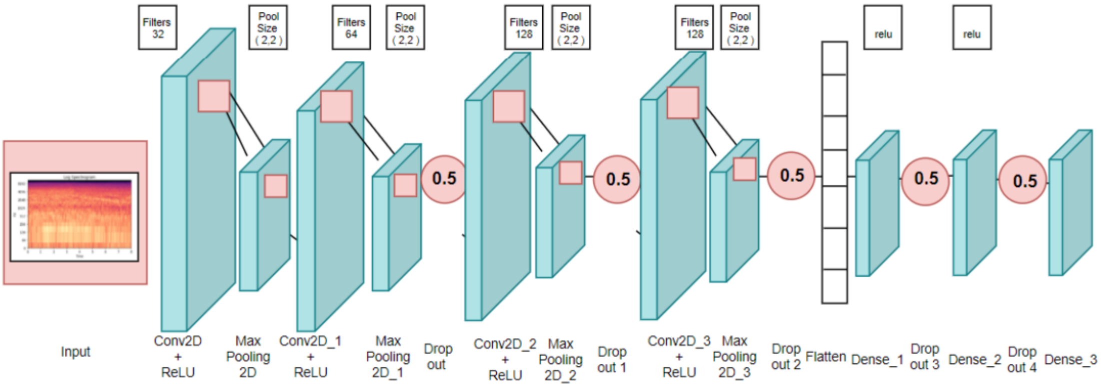


Figure 15: An example CNN Layer Architecture

2. Deep Neural Network (DNN)

Deep Neural Network (DNN) shows great advantages in speech recognition and image recognition. It uses sensors to mimic the learning process of our brain. The learning process consists of two stages: feed forward and back propagation. With back propagation we update the weights and bias of each layer. Each layer learns some properties of the input. If enough training data is available, the network can show pretty good results (Yang et al., 2019).

An artificial neural network with multiple hidden layers (Véstias, 2020).

It is a technology in the field of Machine Learning that can use statistical learning methods to extract high-level features from original sensory data and obtain an adequate representation of the input field in large amounts of data.It is a network with more than two layers and the word “deep” refers to the number of layers through which the data is transformed (Nahid et al., 2021).

A machine learning model class. The main difference between Classical and Deep network scheme is the number of hidden layers and the training process. By using more hidden layers, DNN can achieve higher level of interrelationship.The deep neural network (e.g., DNN) is an artificial neural network (e.g., ANN) with multiple layers between the input and output layers that allow to learn the features that optimally represent the given training data (Raut and Albena, 2020).

3) Long-Short Term Memory (LSTM)

Long short-term memory (LSTM) is an artificial neural network used in the fields of artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network can process not only single data points (such as images), but also entire sequences of data (such as speech or video) (Hochreiter and Schmidhuber, 1997). LSTM can be applied to tasks such as non-compartmental, linked handwriting recognition, speech recognition, machine translation, robot control, video games, and healthcare. LSTM has become the most cited neural network of the 20th century (Schmidhuber, 2021).

A common LSTM unit consists of a cell, an entry gate, an exit gate, and a forget gate. The cell remembers values at arbitrary intervals, and three gates regulate the flow of information in and out of the cell (Hochreiter and Schmidhuber, 1997).

LSTM networks are well suited for classifying, processing and making predictions based on time series data, as there may be delays of unknown duration between significant events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training conventional RNNs (Gers et al., 2000).

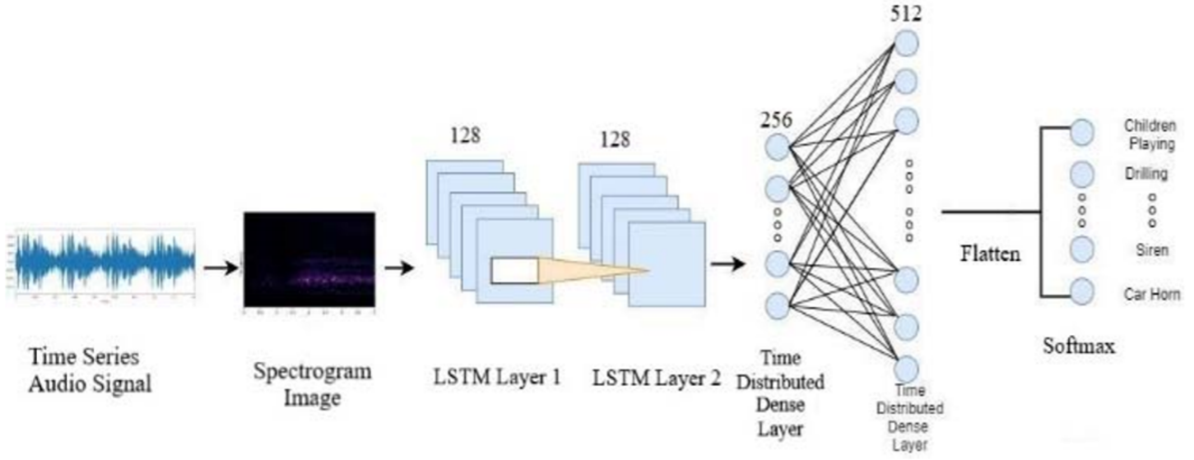


Figure 16: Architecture of an Example LSTM Model

4.1 Models, Classification Urban Sound with Machine Learning Algorithms

As mentioned earlier, we perform the classification of the UrbanSound8K dataset containing 8732 tagged audio sounds ($\leq$ 4s). Sound recordings used come from 10 classes of urban sound, including air conditioning, car horn, children playing, dog barking, drilling, engine idling, gunfire, hammer drills, sirens, and street music.

In the scope of this study,three methods that are widely used in the literature were used for the sound classification. These methods are; Convolutional Neural Networks(CNN),Deep Neural Network (DNN) and Long-Short Term Memory (LSTM). These algorithms are coded with the Python and tensorflow, NumPy, Sklearn, Keras, librosa libraries were used for analysis.Also reporting and interpretation of the mean accuracy and standard deviation obtained in the test folds were made.

We converted the audio data from the time domain to the frequency domain, then used the spectrogram features as input to the machine learning models. Here we extracted Mel Spectrogram (MEL), Mel-Frequency Cepstral Coefficient (MFCC) and Short-Time Fourier Transform (Chroma_STFT) from the most commonly used features in audio processing using the Python library librosa.

The data set was divided into two parts and the part containing files 1,2,3,4,6 of the dataset was used for training the model, and the part containing files 5,7,8,9,10 was used for testing.In this study, each feature extraction was tried separately with models.

4.2 Classification Urban Sound with Convolutional Neural Networks(CNN)

The results obtained from the CNN model are given in Table 1. The best predictive value was obtained by combining MFCC, MEL, Chroma STFT (All) three feature extraction. While applying the CNN model, the model was applied for various layer levels and batch_size, epoch size values.Input Dimension (64,128,256,512) , model Conv2D,activation="relu" ,"softmax" , batch_size=64, epochs=120 were the most predictive results.Stochastic objective functions Adam optimizer'dır.The success of the model in classifying a new audio recording was 57 %.

Table 1: Implementation with CNN and Result

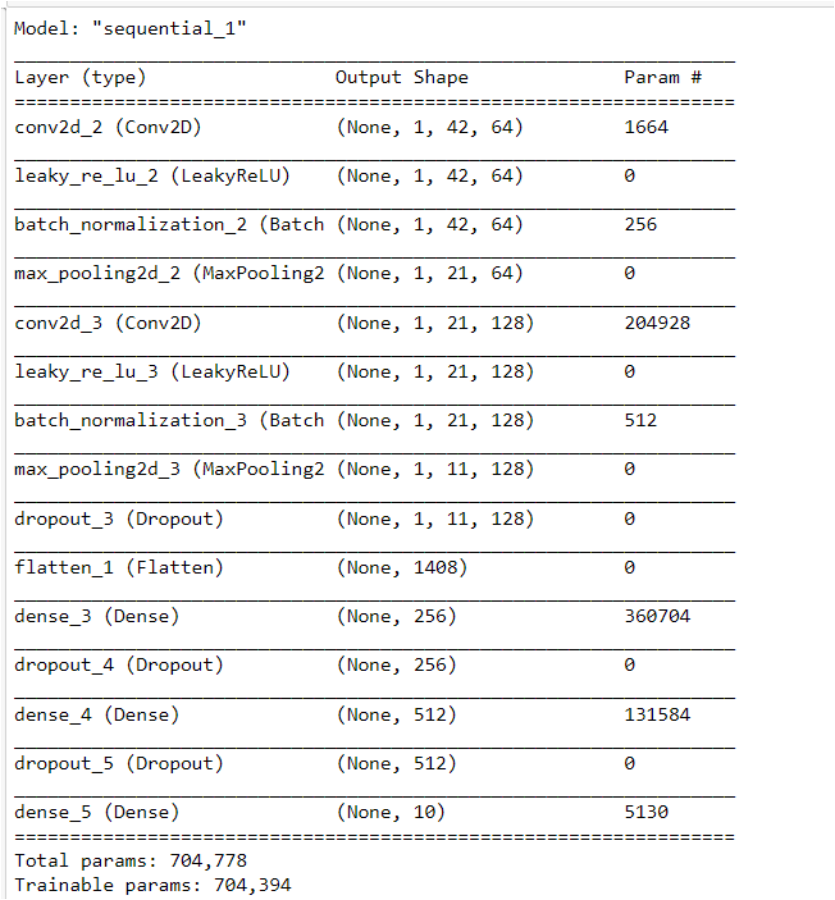
Features	**Train**	**Accuracy (%)**
MFCC	batch_size=256, epochs=250	52%
MEL	batch_size=256, epochs=250	20%
Chroma STFT	batch_size=256, epochs=250	21%
MFCC, MEL,Chroma STFT (All)	batch_size=256, epochs=250	56%
MFCC	batch_size=64, epochs=120	53%
MEL	batch_size=64, epochs=120	19%
Chroma STFT	batch_size=64, epochs=120	22%
MFCC, MEL,Chroma STFT (All)	batch_size=64, epochs=120	**57%**

Table 2 shows the average accuracy of the model and the standard deviation across the test folds. When the standard deviations are analyzed, we can see that the models accuracy and models average accuracy are very close to each other.

Table 2: CNN Model Average Accuracy and Standard Deviation(STD)

Features	**Accuracy (%)**	**Avg Accuracy (%)**	**Std**
MFCC	52%	50%	0,10
MEL	20%	18%	0,03
Chroma STFT	21%	20%	0,02
MFCC, MEL,Chroma STFT (All)	56%	53%	0,11
MFCC	53%	51%	0,07
MEL	19%	18%	0,02
Chroma STFT	20,29%	20,23%	0,02
MFCC, MEL,Chroma STFT (All)	57%	54%	0,08

Figure 17,18,19 shows the CNN model Hyper parameter setting and Model Loss, accuracy.



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_2 (Conv2D)            (None, 1, 42, 64)         1664      
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 1, 42, 64)         0         
_________________________________________________________________
batch_normalization_2 (Batch (None, 1, 42, 64)         256       
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 21, 64)         0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 21, 128)        204928    
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 1, 21, 128)        0         
_________________________________________________________________
batch_normalization_3 (Batch (None, 1, 21, 128)        512       
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 11, 128)        0         
_________________________________________________________________
dropout_3 (Dropout)          (None, 1, 11, 128)        0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 1408)              0         
_________________________________________________________________
dense_3 (Dense)              (None, 256)               360704    
_________________________________________________________________
dropout_4 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense_4 (Dense)              (None, 512)               131584    
_________________________________________________________________
dropout_5 (Dropout)          (None, 512)               0         
_________________________________________________________________
dense_5 (Dense)              (None, 10)                5130      
=================================================================
Total params: 704,778
Trainable params: 704,394
```

Figure 17: Hyper parameter setting of CNN

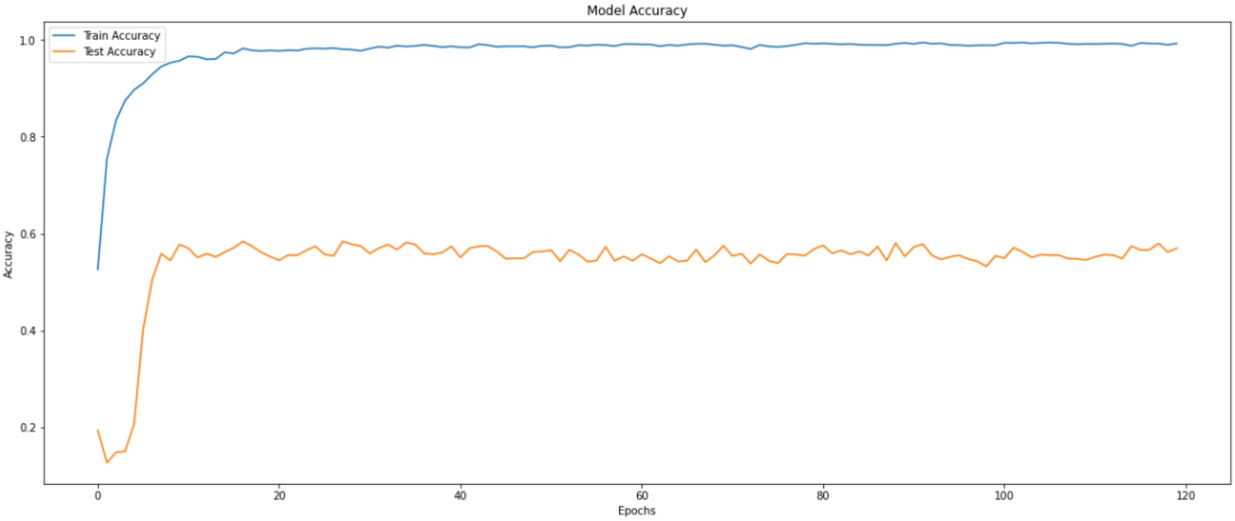


Figure 18: CNN Accuracy

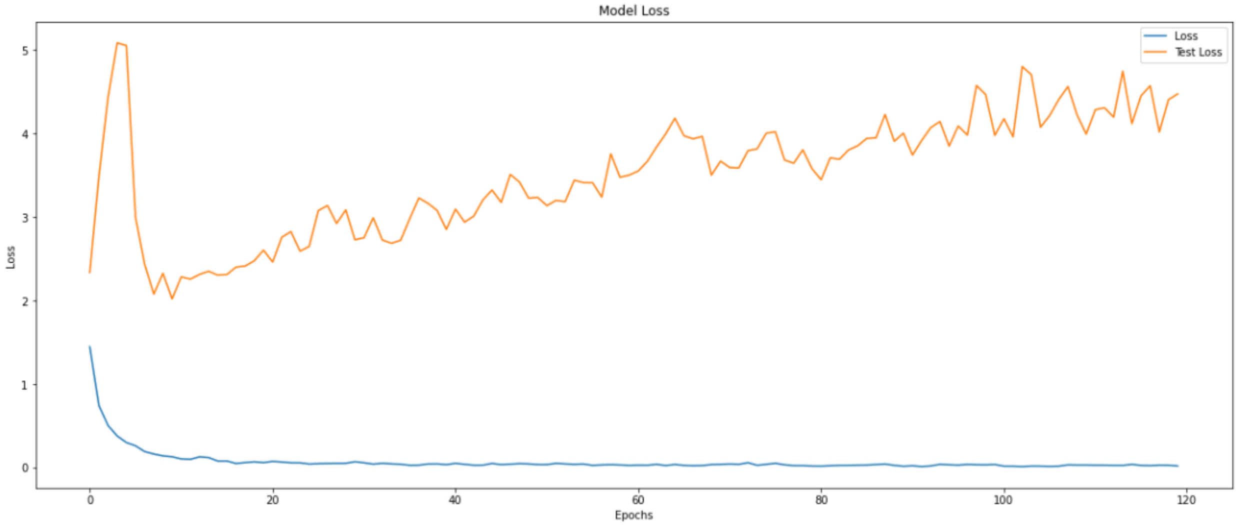


Figure 19: CNN Loss

4.3 Classification Urban Sound with Deep Neural Network (DNN)

The results obtained from the DNN model are given in Table 3. The best predictive value was obtained by combining MFCC, MEL, Chroma STFT (All) three feature extraction. While applying the DNN model, the model was applied for various layer levels and batch_size, epoch size values.Input Dimension (1000,750,500,250,100,50) , model activation="relu" ,"softmax" , batch_size=256; 32, epochs=250;100 were the most predictive results.Stochastic objective functions Adam optimizer'dır.The success of the model in classifying a new audio recording was 54 %.

Table 3: Implementation with CNN and Result

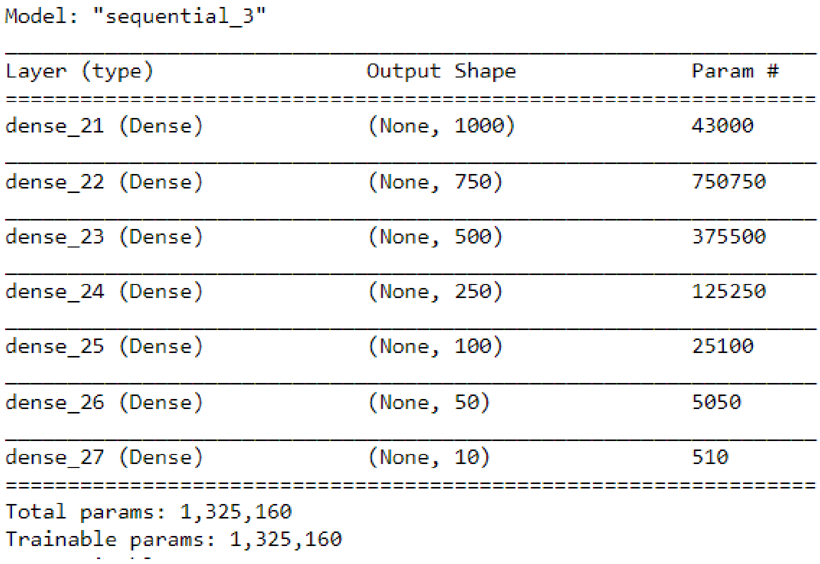
Features	**Input Dimension**	**Train**	**Accuracy (%)**
MFCC	1000,750,500,250,100,50	batch_size=32, epochs=100	48%
MEL	1000,750,500,250,100,50	batch_size=32, epochs=100	19%
Chroma STFT	1000,750,500,250,100,50	batch_size=32, epochs=100	21%
All	1000,750,500,250,100,50	batch_size=32, epochs=100	50%
MFCC	64,128,256,512	batch_size=32, epochs=100	48%
MEL	64,128,256,512	batch_size=32, epochs=100	18%
Chroma STFT	64,128,256,512	batch_size=32, epochs=100	21%
All	64,128,256,512	batch_size=32, epochs=100	54%
MFCC	64,128,256,512	batch_size=64, epochs=120	47%
MEL	64,128,256,512	batch_size=64, epochs=120	19%
Chroma STFT	64,128,256,512	batch_size=64, epochs=120	20%
All	64,128,256,512	batch_size=64, epochs=120	54%
MFCC	1000,750,500,250,100,50,10	batch_size=256, epochs=250	51%
All	1000,750,500,250,100,50,10	batch_size=256, epochs=250	**54%**

Table 4 shows the average accuracy of the model and the standard deviation across the test folds. When the standard deviations are analyzed, we can see that the models accuracy and models average accuracy are very close to each other.the standard deviations are very small.

Table 4: DNN Model Average Accuracy and Standard Deviation(STD)

Features	**Accuracy (%)**	**Mean Accuracy (%)**	**Standart Devision**
MFCC	48%	49%	0.013
MEL	18,7%	18,53%	0.013
Chroma STFT	21%	20%	0.009
All	50%	54%	0.015
MFCC	47,5%	47,2%	0.011
MEL	18%	18,62%	0.007
Chroma STFT	20,60 %	20,50%	0.008
All	53,57%	53,79%	0.015
MFCC	46,7%	46,82%	0.011
MEL	18,7%	18,82%	0.008
Chroma STFT	20,17%	20,42%	0.008
All	54,05%	54,42%	0.012
MFCC	51%	51%	0.012
All	54%	55%	0.015

Figure 20,21,22 shows the DNN model Hyper parameter setting and Model Loss, accuracy.



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_21 (Dense)             (None, 1000)              43000
_________________________________________________________________
dense_22 (Dense)             (None, 750)               750750
_________________________________________________________________
dense_23 (Dense)             (None, 500)               375500
_________________________________________________________________
dense_24 (Dense)             (None, 250)               125250
_________________________________________________________________
dense_25 (Dense)             (None, 100)               25100
_________________________________________________________________
dense_26 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_27 (Dense)             (None, 10)                510
=================================================================
Total params: 1,325,160
Trainable params: 1,325,160
```

Figure 20: Hyper parameter setting of DNN

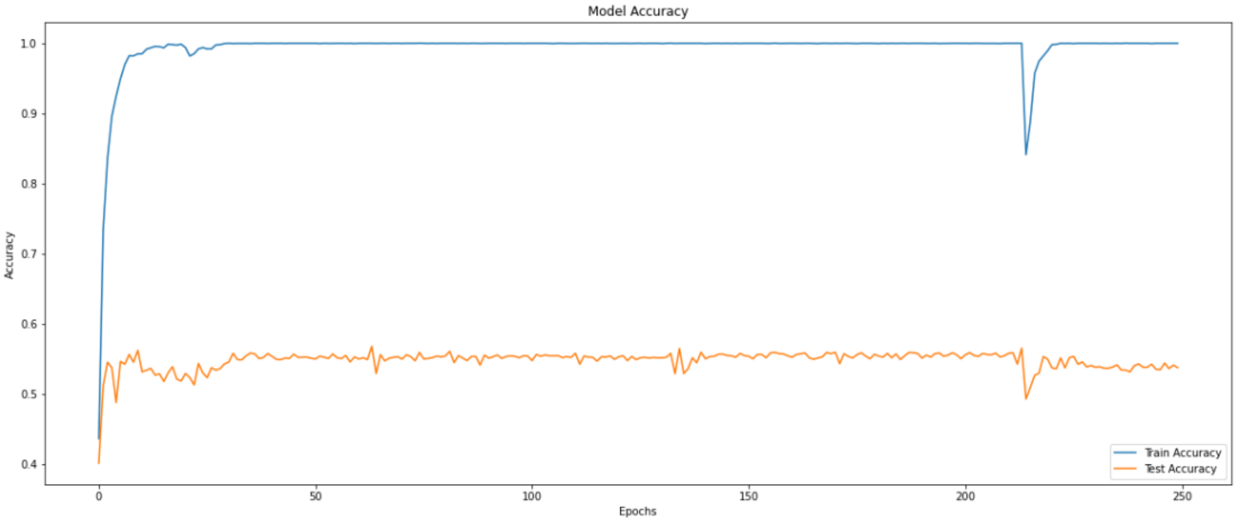


Figure 21: DNN Accuracy

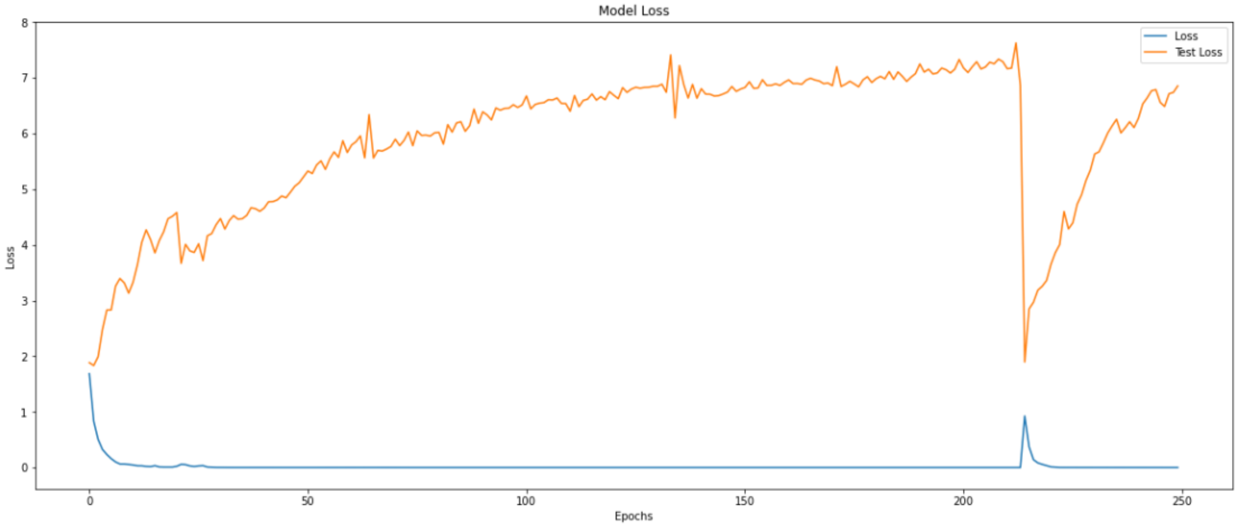


Figure 22: DNN Loss

4.4 Classification Urban Sound with Long-Short Term Memory (LSTM)

The results obtained from the LSTM model are given in Table 5. The best predictive value was obtained by combining MFCC, MEL, Chroma STFT (All) three feature extraction. While applying the LSTM model, the model was applied for various layer levels and batch_size, epoch size values.Input Dimension (128,64,32,10) , model activation="relu" ,"softmax" , batch_size=64, epochs=120 were the most predictive results.Stochastic objective functions Adam optimizer'dir.The success of the model in classifying a new audio recording was 45 %.

Table 6 shows the average accuracy of the model and the standard deviation across the test folds. When the standard deviations are analyzed, we can see that the

Table 5: Implementation with LSTM and Result

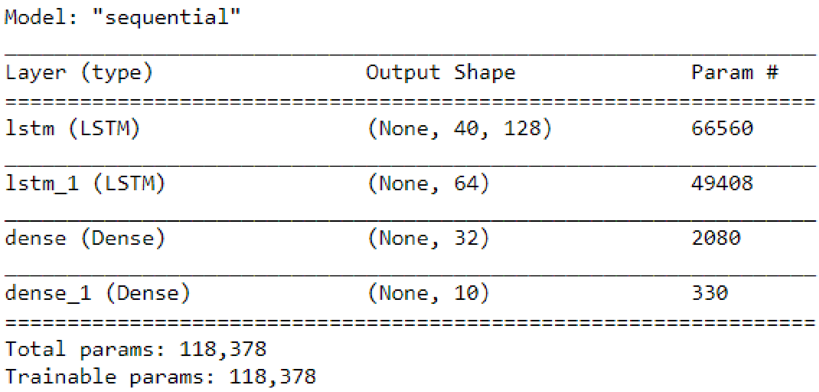
Features	Input Dimension	Train	Accuracy (%)
MFCC	128,64,32,10	batch_size=64, epochs=120	42%
MEL	128,64,32,10	batch_size=64, epochs=120	19%
Chroma STFT	128,64,32,10	batch_size=64, epochs=120	21%
All	**128,64,32,10**	**batch_size=64, epochs=120**	**45%**
MFCC	128,64,32,10	batch_size=256, epochs=250	40%
MEL	128,64,32,10	batch_size=256, epochs=250	19%
Chroma STFT	128,64,32,10	batch_size=256, epochs=250	21%
All	128,64,32,10	batch_size=256, epochs=250	42%
MFCC	64,128,256,512	batch_size=64, epochs=120	40%

models accuracy and models average accuracy are very close to each other.the standard deviations are very small.

Table 6: LSTM Model Average Accuracy and Standard Deviation(STD)

Features	Accuracy (%)	Mean Accuracy (%)	Standart Devision
MFCC	42%	41%	0.036
MEL	19,15%	18,59%	0.011
Chroma STFT	21%	20%	0.008
All	**45%**	**42%**	**0.045**
MFCC	40%	36%	0.05
MEL	18,54%	18,16%	0.012
Chroma STFT	21.00%	20%	0.009
All	42%	41%	0.052
MFCC	40%	39%	0.037

Figure 23,24,25 shows the DNN model Hyper parameter setting and Model Loss, accuracy.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 40, 128)           66560
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense (Dense)                (None, 32)                2080
_________________________________________________________________
dense_1 (Dense)              (None, 10)                330
=================================================================
Total params: 118,378
Trainable params: 118,378
```

Figure 23: Hyper parameter setting of LSTM

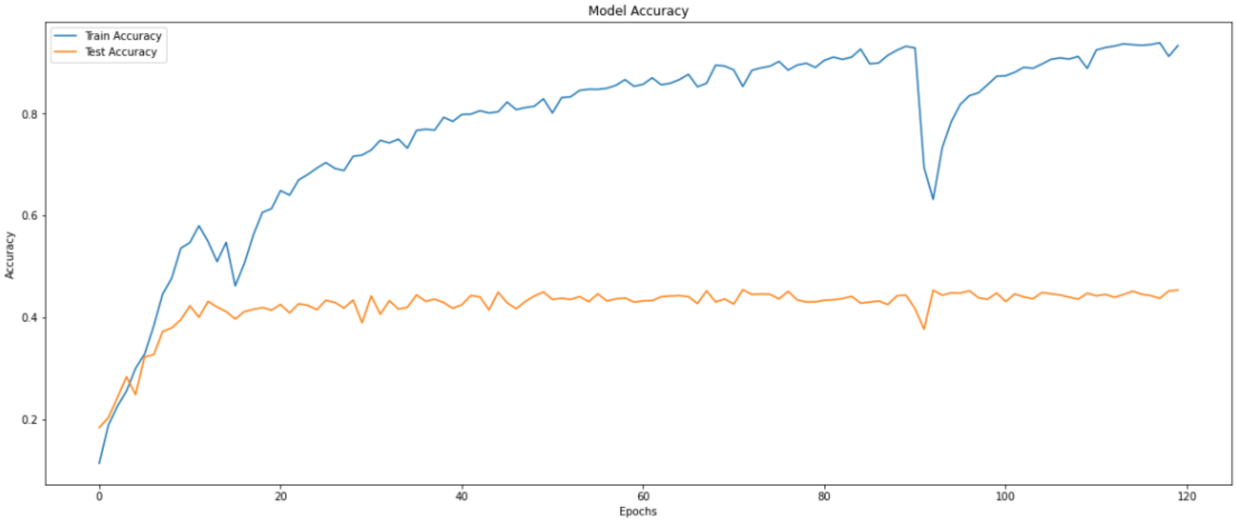


Figure 24: LSTM Accuracy

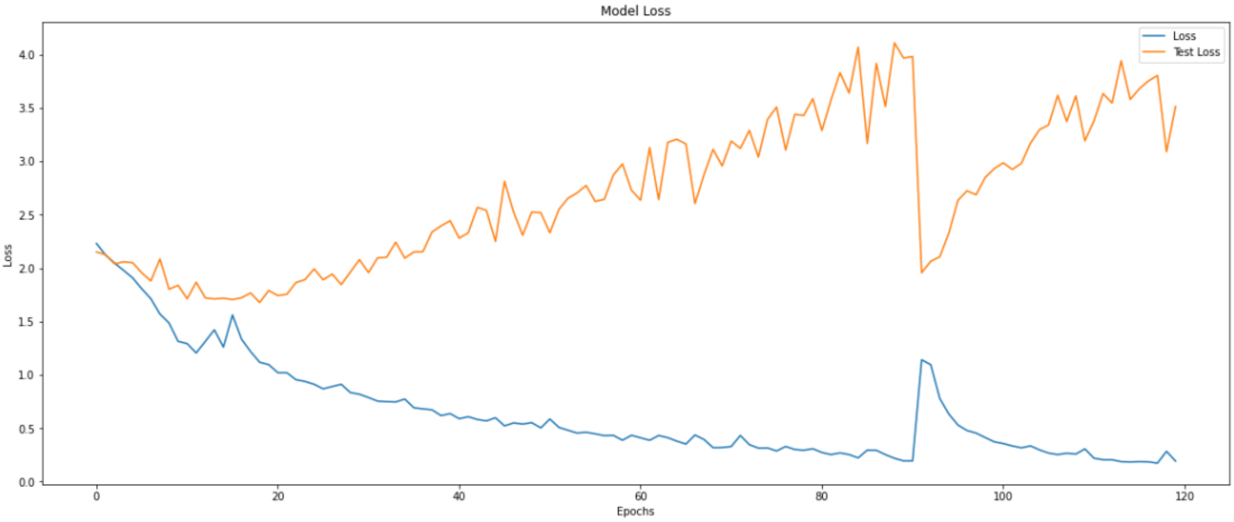


Figure 25: LSTM Loss

5 RESULTS AND DISCUSSION

At this stage, the best model is defined by comparing the results obtained from the developed three models. Besides, the best results obtained in the study were compared with those obtained in the sound classification studies published in the literature.

5.1 Comparison of The Results

In our study, UrbanSound8K dataset containing 8732 labeled audio sounds ($\leq$ 4s) was classified.Convolutional Neural Networks(CNN),Deep Neural Network (DNN) and Long-Short Term Memory (LSTM)models were applied to the collected data. The overall results of these models are given in Table 7. It is clearly seen that CNN is the best model value was obtained by combining MFCC, MEL, Chroma STFT (All) three feature extraction with a success rate of 57 %.

Table 7: Comparison of the results obtained from developed models

Model	**Features**	**Accuracy (%)**
CNN	MFCC, MEL,Chroma STFT (All)	57%
DNN	MFCC, MEL,Chroma STFT (All)	54%
LSTM	MFCC, MEL,Chroma STFT (All)	45%

5.2 Discussion

From the literature review, it can be concluded that CNN, DNN and LSTM models are the most widely used methods for sound classification. MFCC, Mel and Chroma STFT are the most used feature extraction methods. All the features generally extracted in the studies were combined. This has been shown to increase prediction success.

(Salamon and Bello, 2016) used MFCC feature extraction in the SKM model and the success of the model is obtained as 74%. However, its performance improves significantly with data augmentations, giving an average accuracy of 79 %. (Mahmood and Köse, 2021) used MFCC feature extraction in the CNN model and the success of the model is obtained as 88%.(Gunther et al., 2019) used MFCC,Chromagram,Mel feature extraction in the CNN model and the success of the model is obtained as 73%. (Yang et al., 2019) used MFCC,Mel feature extraction in the ResNet18 model and the success of the model is obtained as 98%.(Das et al., 2020) used MFCC,Chroma_STFT,Mel feature extraction in the LSTM model and the success of the model is obtained as 98%. (Lezhenin et al., 2020) used Mel feature extraction in the LSTM model and the success of the model is obtained as 84%.

In Tables 8 and 9, the results of the models for the classification urban sound are given collectively. In our study, we developed CNN model using MFCC,Mel,Chroma STFT feature extraction. We obtained accuracy value of the model as 57 %.

Table 8: The developed models for the Classification of urban sound

Features	**Model**
MFCC,Mel,Chroma_STFT	CNN,DNN,LSTM
MFCC	SKM,PiczakCNN,SB-CNN
MFCC	CNN,DNN
MFCC,Chromagram,Mel	CNN
MFCC,Mel	CNN,DNN,LSTM,VGG11,ResNet18
MFCC,Chroma_STFT,Mel	CNN,LSTM
Mel	CNN,LSTM

Table 9: Comparison of models developed for classification of urban sound

Author	**Best Model**	**Accuracy (%)**
Er&Cesa-Bianchi(2022)	CNN	57%
Salamon & Bello (2016)	SKM	74%
Köse & Mahmood (2021)	CNN	88%
Gunther et al. (2019)	CNN	73%
Yang et al. (2019)	ResNet18	98%
Das et al. (2020)	LSTM	98%
Lezhenin et al. (2020)	LSTM	84%

As given in Table 8, our model offers worse result than many models given in the literature. The main reason for this part can be explained as follows;

1.The original dataset is not large/diverse enough for the models to perform well.

2.Usually, the separation of train and test data is done randomly.In our study, the data set was manually divided into two parts, and the part of the data set containing files 1,2,3,4,6 was used for training the model, and files 5,7,8, 9,10 were used for testing the model.Therefore, it shows that there is not enough diversity of different sampling in the training set for the recognition and identification of some sounds.

6 CONCLUSION

In this research paper, an urban sound classification system based on three algorithms is proposed for feature extraction, firstly, unique features of urban sound recordings are extracted using MFCCs. CNN, DNN, LSTM algorithms were used for classification. Better results were obtained by using MFCC, Mel and Chroma STFT, which are used as features in the models, together. The reporting and interpretation of the mean accuracy and standard deviation obtained in the test folds were made. Thus, how much the overall accuracy deviated from the mean was interpreted.

7 Declaration

I declare that this material, which we now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

References

R. Agrawal. Analytics vidhya :implementing audio classification project using deep learning. URL `https://www.analyticsvidhya.com/blog/2022/03/implementing-audio-classification-project-using-deep-learning/`.

D. Barchiesi, D. S. Dimitrios Giannoulis, and M. Plumbley. Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34, 2015.

R. Cai, L. Lu, A. Hanjalic, H. Zhang, and L. Cai. A flexible framework for key audio effects detection and auditory context inference. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 14(3):1–14, 2006.

S. Chachada and J. Kuo. Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processin*, 3:1–9, 2013.

S. Chu, S. Narayanan, and J. Kuo. Environmental sound recognition with time–frequency audio features. *Multimedia Tools and Applications*, 17(6):1142–1158, 2009.

J. K. Das, A. Ghosh, A. K. Pal, and A. Chakrabarty. Urban sound classification using convolutional neural network and long short term memory based on multiple features. *Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, pages 1–10, 2020. doi: https://www.researchgate.net/publication/346659500.

S. Davis and P. Mermelstein. Acomparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357– 366, 1980. doi: https://courses.engr.illinois.edu/ece417/fa2017/davis80.pdf.

S. Furui. Speaker recognition tecnology. *NNT Review*, 7(2):30–31, 1996.

F. Gers, F. Cummins, and J. Schmidhuber. Learning to forget: Continual prediction with lstm. *Neural Computation*, pages 1–20, 2000.

C. Gunther, K. Le, M. Ranis, and D. Durubeh. Urban sound classification. 2019. doi: http://noiselab.ucsd.edu/ECE228_2019/Reports/Report36.pdf.

S. Hochreiter and J. Schmidhuber. Long short- term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

ILM. Kaggle: Mfcc implementation and tutorial. URL `https://www.kaggle.com/code/ilyamich/mfcc-implementation-and-tutorial/notebook`.

I. Lezhenin, N. Bogach, and E. Pyshkin. Urban sound classification using long short-term memory neural network. *Proceedings of the Federated Conference on Computer Science and Information Systems*, 18(ISSN 2300-5963 ACSIS):57–60, 2020. doi: https://www.readcube.com/articles/10.15439%2F2019f185.

S. Magre, P. V. Janse, and R. Deshmukh. A review on feature extraction and noise reduction technique. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(2):352– 356, 2014.

A. Mahmood and U. Köse. Speech recognition based on convolutional neural networks and mfcc algorithm. *Advances in Artificial Intelligence Research (AAIR)*, 1(1):6–12, 2021.

M. Müller. Chroma feature extraction. *Friedrich Alexander Universitat Erlangen Nürnberg International Audio Laboratories Erlangen*, pages 1–10, 2017.

F. A. Nahid, N. Madhu, and T. Laopaiboon. *Hybrid Neural Networks for Renewable Energy Forecasting: Solar and Wind Energy Forecasting Using LSTM and RNN*. IGI Global, 2021.

T. Qiao, S. Zhang, Z. Zhang, S. Cao, and S. Xu. Sub-spectrogram segmentation for environmental sound classification via convolutional recurrent neural network and score level fusion. *Shanghai Institute for Advanced Communication and Data Science Shanghai University*, pages 1–6, 2019.

E. G. Rajo. github: Urban audio classifier. URL `https://github.com/GorillaBus/urban-audio-classifier`.

R. Raut and F. Albena. *Deep Learning for Moving Object Detection and Tracking*. IGI Global, 2020.

J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Magazine Letters*, pages 1–4, 2016.

J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. *22nd ACM International Conference on Multimedia,Orlando USA*, 2014.

J. Schmidhuber. The most cited neural networks all build on work done in my labs. *AI Blog*, 2021.

A. K. Shah, A. Nepal, and M. Kattel. Chroma feature extraction. *Department of Computer Science and Engineering, School of Engineering Kathmandu University*, pages 1–14, 2019.

M. Véstias. Deep learning on edge: Challenges and trends. *Instituto Politécnico de Lisboa*, pages 1–20, 2020.

Y. Yang, K. Liu, L. Hong, and C. Dai. Urban sound source classification and comparison. pages 1–6, 2019.

Y. Zeng, H. Mao, D. Peng, and Z. Yi. Spectrogram based multi-task audio classification. *Multimedia Tools and Applications*, 78(3):1–19, 2019.