# NOISY SOUND CLASSIFICATION USING DEEP LEARNING

*A project report submitted in fulfillment of the requirements for B.Tech. Project*

**B.Tech.**

*by*

**Divya Lakhwani (2016IPG-035)**
**Khushal Bisani (2016IPG-043)**
**Saurabh Shukla (2016IPG-095)**

विश्वजीवनामृतं ज्ञानम्

**ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR-474 010**

**2019**

# CANDIDATES' DECLARATION

We hereby certify that the work, which is being presented here in this report, entitled **Noisy Sound Classification using Deep Learning**, in fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *May 2019* to *September 2019* under the supervision of **Dr. Neetesh Kumar** and **Dr. Yash Daultani**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

| | | |
|---|---|---|
| Divya Lakhwani | Khushal Bisani | Saurabh Shukla |

Date:

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

| | |
|---|---|
| Dr. Neetesh Kumar | Dr. Yash Daultani |

Date:

## ABSTRACT

The learning capabilities of deep convolutional neural networks (CNNs) can be used to develop sound classification systems to solve traditional system effectiveness problems. In this study, first we deal with various properties of sound and convert raw data into consistent data. We learn different spectro-temporal patterns of sound. we use melspetrogram and mel frequency cepstral coefficient (MFCC) features of sound and apply deep learning models on them.
The CNN architecture of melspectogram has 4 hidden layers and the MFCC contains 5 hidden layers with an input and a output layer. The data augmentation is used to overcome the problem of data scarcity. Also we show that the SpecAugment augmentation combined with CNN model outperforms the CNN model without augmentation as the data augmentation increases the efficiency of the model drastically.
Finally, we compare which feature gives better results for different sound data and also examine the influence of each feature set on the models classification accuracy for each class, and observe that the accuracy for each class is influenced differently by each feature set. At last we suggest that accuracy of classification can be further improved by applying hybrid model of recurrent neural network and convolutional neural network.

*Keywords:* Sound classification, Deep Learning, Convolutional Neural Networks, SpecAugment

## ACKNOWLEDGEMENTS

We owe it to **Dr. Neetesh Kumar** and **Dr. Yash Daultani** for offering us the sovereignty of exploring our domain and experimenting with our ideas circuitously. We would like to extend our major gratitude to both these mentors for their incessant guidance and motivating reciprocation towards our dissertation. They continually and consistently conveyed a spirit of sincerity in regard to our project. They mustered up confidence within us and pulled together our ideas into a meaningful and successful project. Their persistent assistance and guidance throughout the course nurtured our work. They were on their toes to clear even our minor doubts and their judgment, suggestions, criticism were invaluable in bringing our thesis to the stature it has reached.
We would also like to pay special homage to our institution for providing us a platform and all the requisite resources for carrying out this work.
In addition to this, we would like to express our sincere gratitude to our colleagues who infused in us a competitive spirit, also lending their helping hands. Thanks to the Almighty for giving us the strength and ability to accomplish our target

(Divya Lakhwani) (Khushal Bisani) (Saurabh Shukla)

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| CNN | Convolutional Neural Networks |
| RNN | Recurrent Neural Networks |
| DCNN | DeepConvolutional Neural Networks |
| DNN | Deep Neural Networks |
| ReLU | Rectified Linear Unit |
| MelSpec | Mel Spectrogram |
| MFCC | Mel Frequency cepstral Coefficient |
| FFT | Fast Fourier Transform |
| STFT | Short Term Fourier Transform |
| SGD | Stochastic gradient descent |
| Adam | Adaptive Moment Estimation |
| FM | Frequency Masking |
| TM | Time Masking |

# CHAPTER 1

# INTRODUCTION AND LITERATURE SURVEY

## 1.1 Introduction

Recent studies in acoustics, psychoacoustics, psychology, and cognitive sciences have greatly extended our understanding of sound and music nature and perception. Automatic classification of environmental noise is an increasing area of study with so many applications (e.g. - Music Recommendation , Audio Source Separation etc.) in the actual globe [1]. While there is a wide range of studies in associated audio areas such as speech and music, there is relatively little work on classifying environmental sounds [2, 3]. Likewise, observing latest advances in image classification where CNNs are used to classify images with more precision and scale, the issue of the applicability of these methods in other areas, such as sound classification, is raised [4].

The main problem and barriers in the field of audio analysis is the scarcity of categorized audio files of few seconds. Previous researches focused on audio from carefully produced by clipping films or TV tracks, specific areas such as elevators or office spaces, and business. The great effort to manually annotate real-world information means that field-based datasets tend to be relatively small(for example, the IEEE AASP Challenge incident detection dataset consists of 24 recordings per 17 classes).

The next challenge faced by the research community is that there is no prevalent vocabulary when working with noisy sounds [5]. This implies that sound classification into semanthetic groups may vary from research to study, making it difficult to compare results.

The goal of this research is to identify various sound properties that help each sound clip have different identity. We normalize sound data then extract audio features MFCC(Mel-Frequency Cepstral Coefficients) and Mel-Spectrogram which convert audio data to images so that they can be fed to Deep Learning Model [6, 7].

Due to the fundamental principles in Algorithm's design, statistical learning algorithms can not learn well from a few examples. Nature is not following a rigid pattern, fluctuations always exist. So, if we observe an infinitely large amount of data then we see a well-visible trend in the big picture, describing the law behind the problem and a slight distortion around it. It is known as the issue of "signal and noise". Now, while our objective is to get that pattern, all we can do with a computer realistically is to sample a very tiny portion of information and attempt to extract the pattern. The signal-to-noise ratio is very low in statistical learning, and tiny data sets can either be highly misleading or non-representative in relation to the fundamental trend.

An effective solution to this issue is Data Augmentation [8, 9]. Data augmentation is the method through which we generate fresh samples of synthetic training by adding tiny disturbances to our original training set. The goal is to invariate our model to these disturbances and to enhance its capacity to generalize. In our research we use SpecAugmentation technique, which is used for data augmentation [10]. There are 2 types in which we modify audio files Time Masking and Frequency Masking [11]. SpecAugment is applied directly to the neural network feature inputs (i.e. coefficients of the filter bank). The strategy of augmentation comprises of distorting the characteristics, masking frequency channel blocks, and masking time step blocks. We use SpecAugment for end-to-end speech recognition duties on listening, attending, and spell networks.

## 1.2 Literature Survey

Environmental sound classification is an emerging research area with numerous applications in the real world. A lot of research has been done on the topic in various research papers [6]. Various research papers give us various ways to apply our machine learning knowledge on the topic. The earlier study focuses primarily on a profound CNN architecture for noisy sound classification and proposes the use of distinct audio augmentation techniques to overcome the issue of information scarcity and to prevent overfitting problems. Deep convolutional neural networks (CNNs) are appropriate for classifying distinct noisy environmental sounds because they are able to represent noise over moment and frequency in their energy modulation patterns [12]. The profound CNN architecture suggested in past study consists of numerous convolutional layers with pooling activities, followed by few completely connected layers [6]. Here different audio features of sound such as MFCC(Mel Frequency Cepstral Coefficients), Melspectrogram, spectral flatness, RMS value etc can be used. The suggested model, combined with increased information , generates adequate outcomes for classification of environmental sound. It increases its efficiency by combining a profound, high-capacity model with an increased training set: this combination surpasses both the suggested CNN without an increase and a shallow dictionary learning model with an augmentation. The model examines the impact of each augmentation on the precision of classification systems for each class and observes that each augmentation influences the precision for each class differently. In this model, various methods of data augmentation were used, such as time stretching, pitch shifting, dynamic range compression, etc.

In our model we are using deep CNN model with more number of layers and we have also introduced various parameters to which will help in improving the overall accuracy of the model. We are utilising various audio features such as MFCC and Mel spectrogram simultaneously working on both the features [7]. We are also using SpecAugmentation which is applied directly to features input of neural network [10].It decreases the data scarcity problem and greatly improves the accuracy.

## 1.3 Research Gaps

The research paper addresses the scarcity of audio data for the classification of environmental noise [13]. Data augmentation techniques can be applied to achieve a good level of accuracy. Also the choice of audio feature is important here as some audio files might give better results for some particular features. The accuracy of previous research is 75% without data augmentation and 79% with data augmentation techniques, so there is very good chance to improve the accuracy further [6].

## 1.4 Objectives

The main objectives of our project can be summarized in the following points:

(a) To classify noisy sounds using some efficient methods like deep learning architecture.

(b) To learn various features of audio signals by applying various features extraction techniques like mel-frequency cepstrum coefficient(MFCC) and mel-spectrogram representation of audio signal.

(c) To solve the data scarcity problem by implementing methods for data augmentation such as the method of SpecAugmentation technique.

# CHAPTER 2

# DESIGN DETAILS, IMPLEMENTATION AND TESTING

## 2.1 Design Details

Figure 2.1 describes the methodology comprising of subjecting audio files to data preprocessing, Feature extraction, data augmentation, CNN model and finally classification result. First, we preprocess data and then move through feature extraction. After

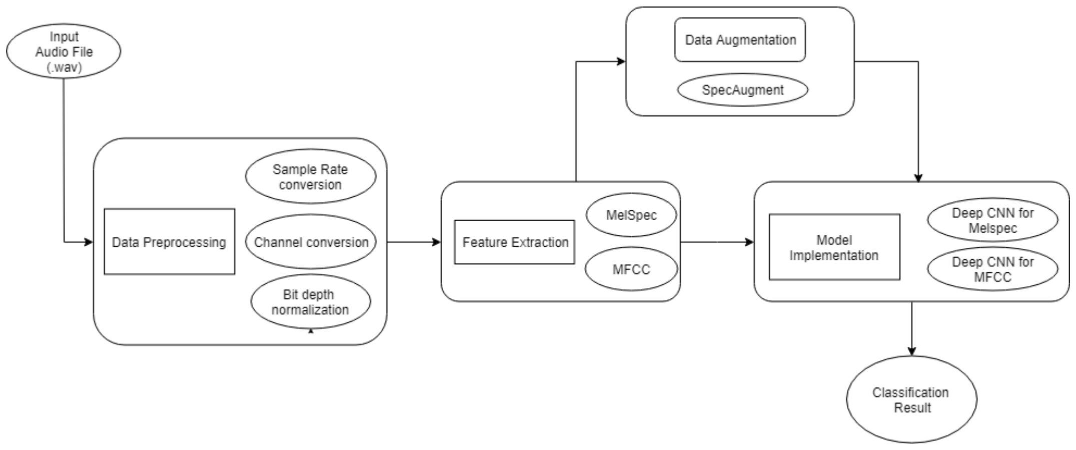


Figure 2.1: Methodology

extracting feature, those feature fed into suitable neural network to get suitable classification result. Here, we also apply new data augmentation method.

### 2.1.1 Data Preprocessing

As we understand, the process of data mining converts raw information into a comprehensible format. Raw data (real data from the world) is always incomplete and can not be transmitted through a model. That would make some mistakes. That's why we need to preprocess information before we send a model.

In this, we are using dataset UrbanSound8K which contain different type of noisy sound [13]. Here, dataset is in raw form. These audio files possess are of different length(time duration), different sample rate, different bit depth and in many files there may be repetition of same sound multiple times.So, first we have to deal with all these. In this phase, audio data and information about them are analyzed by learning their numerous audio properties including bit depth, sample rate, audio channel, sample duration and volume level. We would also convert them to that kind of format that it maintains consistency throughout the set of data like each and every audio signal posses same sample rate, bit depth and other properties.

### 2.1.2 Feature Extraction

Extraction of features is a very important part in analyzing and finding relations between different things. The data provided of audio cannot be understood by the models directly. So, There is a need to extract some features and characteristics of audio files. The signal is the physical representation of positive knowledge. This knowledge can be in different forms like a picture, data voice and many more other types of representation [14]. Feature extraction is the way of representing audio in a complex numerical representation that can be used to differentiate audio by uniquely characterizing a segment of audio. It is a process that explains most of the data but in an understandable way.
In this, for representing an audio signal two features are used. One is Mel Frequency Cepstral Coefficient (MFCC) and Mel Spectrogram. MFCC(mel frequency cepstral coefficient) and mel spectrogram audio features which summarizes the distribution of frequencies across the window size so that the sound's frequency and time characteristics can be analyzed[7, 15].
To compute MelSpec, first pre-emphasis filter is apply on signal to amplify high frequencies. This step is useful because it balance the frequency spectrum and avoid numerical problem during FFT. After pre-emphasis, signal is splitted into short time frames. As frequency of signal is changing over time very frequently, it doesn't make sense to apply fourier transform for whole audio signal because it lose the frequency contours of the signal over time. To avoid this, fourier transform applied for a very short time period by assuming that frequency is constant over that small period of time. After partitioning the signal into frames, a window function is applied to each frame. Then we can do the N-point FFT on each frame to compute the frequency spectrum, also known as the Short-Time Fourier Transform (STFT). The final stage in the calculation of filter banks is to apply Mel-scale triangular filters to the power spectrum to obtain frequency bands [16].
In some machine learning and deep learning algorithms, MelSpec is highly correlated. It could be problematic. To decorrelate the filter bank coefficients and yield a com-

pressed representation of the filter banks, discrete cosine transformation is done which gives us MFCC TF-patches [16].

### 2.1.3 Deep CNN

A convolutional neural network (CNN) is a type of artificial neural neural network that uses perceptrons, a machine learning unit algorithm, for supervised learning, for modeling [17].In this study, two different architecture of deep CNN proposed, one for each feature.The architecture of models is in a sequential pattern, starting with Conv2D convolutional layers, with our final output layer being a dense layer. Each layer has its own significance.

- **Convolutional layer:-**This layer consists of a number of measurable filters with a small receptive field extending across the depth of the input volume. Each filter is convolved across the input volume by computing a dot product between input and filter weights to produce a multi-dimensional output of that filter. The filter maps of all filters are stacked together along the depth dimension and serve as output of the convolution layer. Convolution layer preserves the spatial relationship by learning features using small receptive field. Therefore convolution layer helps in capturing local dependencies in the image.
  In general, operation of convolution layer can be mathematically represented as:

  $$o_{\mathrm{i}} = \sigma(\sum a_{\mathrm{ij}} * * o_{(\mathrm{i}\text{-}1)}) \tag{2.1}$$

  *Where,*
  *** represents convolution operation*
  *$o_i$ = output of jth feature map*
  *$o_{i\text{-}1}$ = outputs of (i-1) layer*
  *$a_{ij}$ = value of $i^{th}$ layer in $j^{th}$ feature map*
  *$\sigma$ = Activation function applied to the convolution layer*

  The allocation of depth column around height and width are controlled with the help of stride. Stride specifies the way we slide the filter and has a direct impact on spatial dimensions of output. Larger stride leads to significant reduction in dimensionality of convolution output.
  Contribution of all input units in the output of convolution layer is not uniform. Centre units tend to contribute more than corner units. This discrepancy can be countered by zero padding the corners. Further it prevents reduction of spatial dimensionality. The depth of output of convolution layer is equal to number of

filters used as the 2-dimensional output of each filter is stacked against each other end to end.
Convolution layers use a shared variables scheme to control the number of free variables. All neurons in a single slice of depth share the same variables. It is based on the presumption that, if a feature is useful for computing at some spatial position, it should also prove helpful for computing at other positions.
We are using convolutional layers to extract spatially distributed features from the images. To obtain a larger receptive field with relatively lower number of parameters we have used stacks of convolutional layers with each stack consisting of multiple convolutional layers of equal filters, kernel size and padding. Further, to counter the effect of feature loss by pooling operations, we have increased the number of filters after every pooling operation.

- **Pooling layers:-**This section would decrease the number of parameters if the images are too large. Max pooling is being used to shrink the size of each map but retains valuable information. And global average pooling, it has a great significance in architecture because it is responsible for removing overfitting. The reason is that fully connected layers, that became very large as networks grown in size, wound up having a very large number of weights. It requires a lot of power, but they're also responsible for over-fitting. As a result, the use of global average pooling enables networks to operate with less computing power and better generalisation performance.

- **Activation function:-**In the architecture, the transform function of the Rectified Linear Unit (ReLU) only activates the node when the input is over a certain amount, while the input is below zero, the output is zero, but if the input is above a certain limit, it has a linear relationship with the dependent variable. The ReLU activation function is used which is fast in computation relative to the Sigmoid activation function.

$$R(x) = max(0, x) \tag{2.2}$$

*Where,*
*x = input to the neuron*
And Softmax regression is a form of logistic regression that normalizes an input value into a vector of values that follows a probability distribution whose total sums up to 1. Softmax turns arbitrary real values into probabilities. The outputs of the Softmax transform are always in the range [0,1] and add up to 1.

$$\mathrm{s}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{2.3}$$

*Where,*
*x = a vector of the inputs to the output layer*
*i = indexes the output units, so j = 1, 2, ..., K*

- **For compiling our model, we use the following three parameters:-**

  1. **Loss function -** we will use categorical cross-entropy. This is for single label categorization. This is applied when only one category applicable for each data point.
  2. **Metrics -** we will use the accuracy metric which will allow us to view the accuracy score on the validation data when we train the model.
  3. **Optimizer -** Here, we use two optimizer. One is Stochastic Gradient Descent, which is for MelSpec feature input and another is Adaptive Moment Estimation, which is for MFCC feature input.
     For each training instance, SGD conducts an update on parameter. Due to these constant updates, the update parameters have an increased variance and cause the loss function to change dramatically to different intensities. This is a good thing because it allows us to examine fresh, and possibly better, local minima. Here specialty in SGD is that nesterov momentum is used. Nesterov's momentum is a simple adjustment to a standard momentum. Here, the gradient value is not calculated from the present position in the space parameter, but rather from the intermediate position. This enables, because while the term gradient always points in the correct direction, the term momentum may not. If the momentum phrase points in the wrong direction or overshoots, the gradient can still go back and correct in the same update phase.
     Adam is another method that computes adaptive learning rates for each parameter. In addition, Adam also keeps an exponentially decaying average of past gradients.

### 2.1.4 Data Augmentation

Due to the lack of sufficient quantity of training data, the efficient size of data can be increased through the different augmentation methods, which has considerably improved the efficiency of profound networks in the image and sound classification domain. In the case of audio classification, augmentation traditionally includes deforming or adding background noise to the audio waveform used for training in some manner

(e.g. speeding or slowing it down). This makes the dataset effectively larger, as different improved versions of a single input are fed into the network during training, and also makes it possible for the network to become robust by promoting helpful features [8], [9]. Existing standard techniques of augmenting audio input, however, introduce extra cost of computing and sometimes require extra data.
In this project, SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, we adopt a distinctive approach to increase audio data and treat it as a visual problem rather than an audio problem [10]. Instead of augmenting the input audio waveform as usual, SpecAugment applies an augmentation to the audio spectrogram directly on an image representation of the waveform. It's implementation is easy, computationally inexpensive, and requires no extra data.

## 2.2 Experimentation Setup

### 2.2.1 Programming Setup

The models are trained on colaboratory which is a Jupyter notebook environment that requires no setup and runs entirely in the cloud. With Colaboratory one can write and execute code, save and share analyses, and access powerful computing resources like GPU.

#### 2.2.1.1 Programming Languages

Python is a high level language which is interpreted in other words bytecode-compiled and dynamic language. Python has following features which make it an excellent choice for this project:

- It is an object oriented programming language with structured and functional programming methods.
- It has extensive library support for deep learning applications.
- It has automatic garbage collection.
- It provides support for dynamic data types and dynamic or runtime type checking.

#### 2.2.1.2 Modules and Dependencies

- **Numpy** Numpy is a python package which is used as fundamental package in scientific developments and computing [18]. Numpy provide support for multidimensional array objects and its various derivatives like matrices and masked arrays. It provides a wide collection of routines and methods for faster operations on the arrays like selecting, logical, basic linear algebra, random simulation, mathematical operations, I/O, sorting shape manipulations etc.

- **Pandas** Pandas is an open source python library used for data analysis and manipulations. Pandas has wide variety of tools for reading and writing data from different file formats. Datasets can be reshaped and pivoted and it has support for time series functionality, frequency conversion, data range conversion, date shifting and lagging and moving window regression.

- **LibROSA** LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. It includes the nuts and bolts to build a MIR(Music information retrieval) system. To fuel more audio-decoding power, you can install ffmpeg which ships with many audio decoders [19].

- **nlpaug** nlpaug is a python data augmentation package for text and audio. Goal of this package is improving deep learning model performance by generating textual and audio data. It also able to generate adversarial examples to prevent adversarial attacks.

- **Tensorflow** Tensorflow is the leading software library for the numerical computation and it uses data flow graphs. Tensorflow is developed by Google and is an open source library. Mathematical operations are represented as nodes of the graphs while multidimensional data arrays are called tensors and represented as edge of the graph which is communicated between mathematical operations. Tensorflow has flexible architecture which allows user to have computation extended to one or more GPU and CPU, server, desktop or mobile devices having single API. Tensorflow was originally developed for the purpose of conducting deep learning and machine learning research within Machine Intelligence research in the Google organization by Google Brain Team but the systems were general and can be extended to wide variety of domains as well.

- **Keras** Keras is a high level deep learning API which implements basic architecture of the CNN and RNN. It is written in python and uses tensorflow or theano as back-end. It allows user to develop prototype faster through extensibility, user friendliness and modularity. It also interface to write new types of layers which is compatible with existing layers hence making it easier to build hybrid models. Keras support GPU accelerated deep learning thereby decreasing the model training time.

## 2.3 Implementation and Testing

### 2.3.1 Data Exploration and Feature Extraction

- **UrbanSound dataset:-**In this, well-known data set UrbanSound8k is used [13]. This data set contains 8732 sound excerpts of total 27 hours of audio over 10 classes of acoustic sound with 18.5 hours of sound event occurrences. Classes are Children Playing, Air Conditioner , Engine Idling, Dog bark, Drilling , Gun Shot , Jackhammer, Siren , Car Horn and Street Music.These sound excerpts are digital audio files in .wav format.

| Class ID | Class |
|---|---|
| 0 | air conditioner |
| 1 | car horn |
| 2 | children playing |
| 3 | dog bark |
| 4 | drilling |
| 5 | engine idling |
| 6 | gun shot |
| 7 | jackhammer |
| 8 | siren |
| 9 | street music |

Table 2.1: Class ID corresponding to their classes

- **Visual inspection:-**Each sound has its unique representation in time and frequency domain. From the time domain representation, it is found that few audio signals look very similar like - dog bark and gun shot, street music and air conditioner. It is difficult to visualize the difference between some of the classes. Particularly, the wave forms for repetitive sounds for air conditioner, drilling, engine idling and jackhammer are similar in shape.
  Likewise, the peak in the dog bark waveform is identical in shape to the gunshot waveform (the waveform differ in that there are five peaks for five gunshots compared to the seven peaks for seven times dog bark) shown in fig 2.2. A human can manually detect the type of sound by listening. It will be interesting to see how well a deep learning model will be able to extract the necessary features to distinguish between these classes.

- **Class distribution:-**It is found that class labels are unbalanced. Out of 10 classes, 7 classes have 1000 audio files. While siren, car horn and gun shot consist of 929, 429 and 374 respectively. For car horn and gun shot, it is too less as compared to other classes.

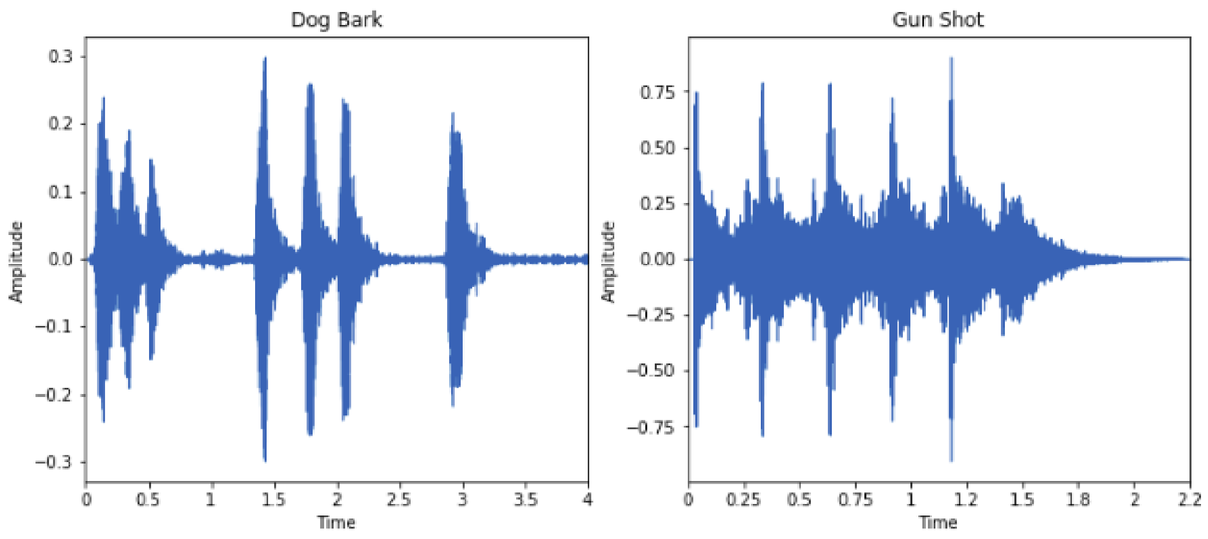


Figure 2.2: Similarity between different classes of audio signal

- **Audio sample properties:-** There are many properties that sound possess. Here, by iterating through all audio files, it is found that many audio files possess different types of sample rate, audio channel and bit-depth.

  In table 2.1, there is a wide range of sample rates that have been used which is ranging between 8 kHz to 192 kHz. Sample rate conversion is applied and convert all the sample rate to 22050 Hz using python package Librosa.

| Sample rate(Hz) | Audio files |
|---|---|
| 44100 | 5370 |
| 48000 | 2502 |
| 96000 | 610 |
| 24000 | 82 |
| 16000 | 45 |
| 22050 | 44 |
| 11025 | 39 |
| 192000 | 17 |
| 8000 | 12 |
| 11024 | 7 |
| 32000 | 4 |

Table 2.2: Sample rates of the audio files

In table 2.2, Most of the samples have two audio channels (stereo) with a few with just the one channel (mono). Here the stereo channel is converted into the mono channel by averaging the values of the two channels using python package

Librosa. Fig 2.3 shows the time domain representation of audio in stereo and mono channel form.

In table 2.3, There is also a wide range of bit depths. It normalizes by taking

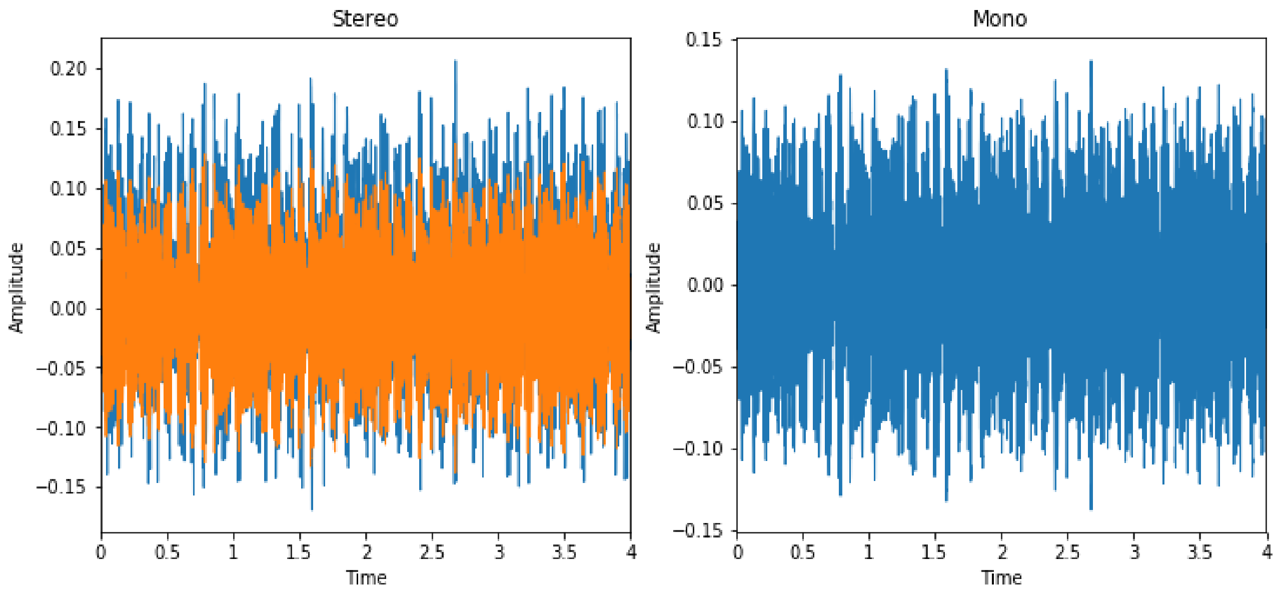


Figure 2.3: Stereo to Mono channel conversion

| Audio channel | Audio files |
|---|---|
| 2 | 7993 |
| 1 | 739 |

Table 2.3: audio channels of the audio files

the maximum and minimum amplitude values for a given bit-depth.

| Bit depth | Audio files |
|---|---|
| PCM 16 | 5758 |
| PCM 24 | 2753 |
| FLOAT | 169 |
| PCM U8 | 43 |
| MS ADPCM | 8 |
| IMA ADPCM | 1 |

Table 2.4: bit depths of the audio files

- **Features of audio:-**To extract these features, we use Librosa python package. We extract Mel-spectrograms with 128 components covering the audible frequency range (0 - 22,050 Hz), using sample rate at 22,050 Hz, the length of FFT window is 2048, the number of samples between successive frames is 512. Each frame of audio is windowed by a window(). The window will be of length 2048. To extract the MFCC feature, we use the same parameters which are used for Mel-spectrogram including one more parameter - cepstral coefficient 40 for discrete cosine transform.

Mel spectrogram of 128x128 dimension and MFCC of 40x174 dimension feature of each audio file are extracted. Only those audio files feature is extracted that has time duration greater than 3 second. Total 7,468 features of audio TF-pathes are extracted for each type of feature. Out of total 7,468 TF-patches, 6500 is kept for training and rest for validation and testing. Fig 2.4 and 2.5 show the MelSpec and MFCC TF-patches of dog bark audio respectively.

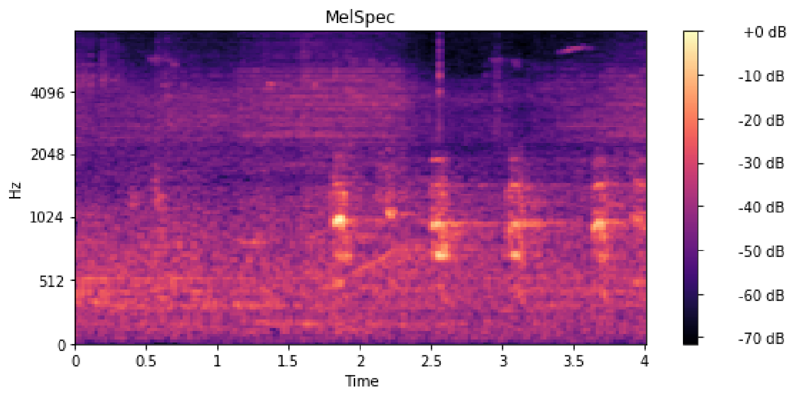


Figure 2.4: MelSpec representation of dog bark audio

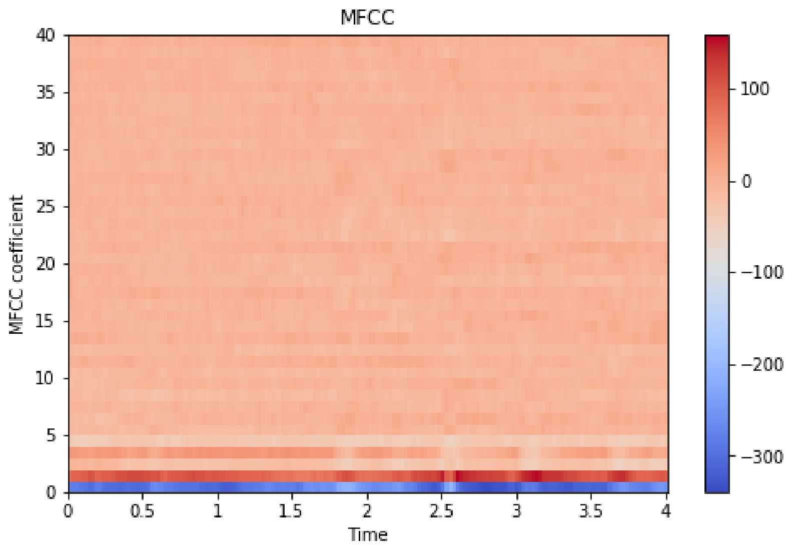


Figure 2.5: MFCC representation of dog bark audio

### 2.3.2 Deep CNN Architecture

- **The proposed CNN architecture for the input of Mel-spectrogram feature is**

**parameterized as follows:**

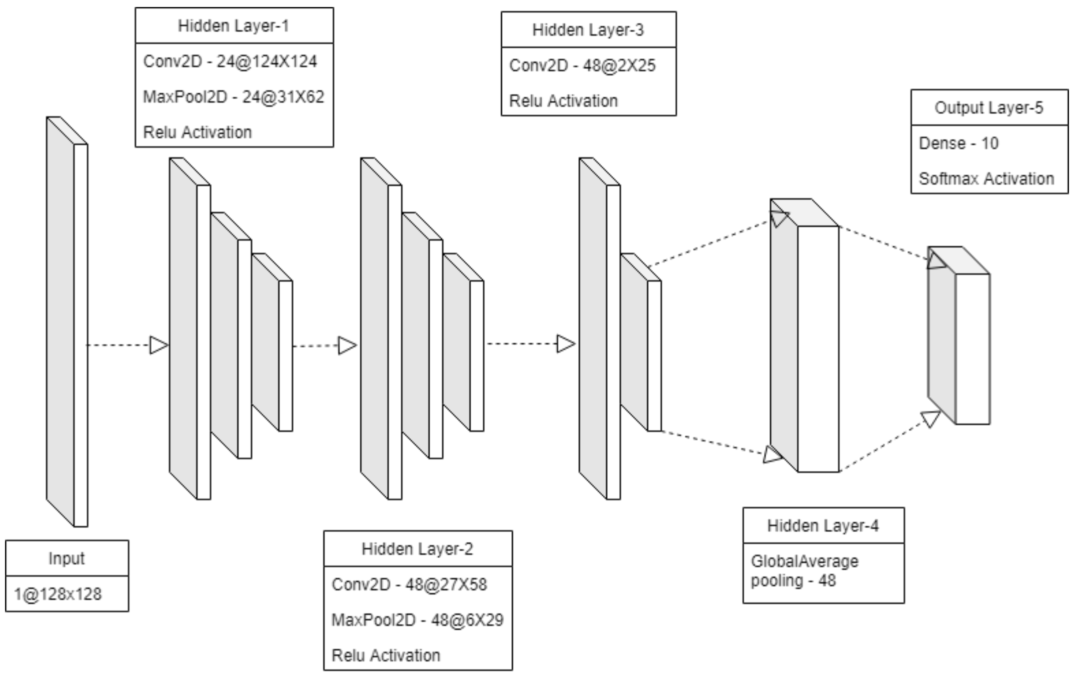


Figure 2.6: Architecture of MelSpec-CNN model

Layer 1:- It consists of Conv2D convolutional layer having kernel size (5,5), filters 24, stride (1,1), input shape of (128,128,1). This is followed by (4,2) strided Maxpooling layer of pool size (4,2) and a rectified linear unit (ReLU) activation function.
Layer 2:- It consists of Conv2D convolutional layer having kernel size (5,5), filters48. This is followed by (4,2) strided Maxpooling layer of pool size (4,2) and a rectified linear unit (ReLU) activation function.
Layer 3:- It consists of Conv2D convolutional layer having kernel size (5,5), filters48. This is followed by a rectified linear unit (ReLU) activation function.
Layer 4:- It consists of GlobalAveragePooling2D layer.
Layer 5:- It consists of Dense layer having 10 units for 10 classes. This is followed by a Softmax activation function.

Figure 2.6 shows the architecture to train the audio Melspec feature. The model optimizes cross-entropy loss via stochastic gradient descent [20]. SGD is set at learning rate 0.01 and nesterov momentum 0.9. Each batch consists of 64 TF-patches randomly selected from the training data (without repetition).The model is trained for 30 epochs. A validation set is used to identify the parameter setting (epoch) achieving the highest classification accuracy.Here keras ReduceLROn-Plateau callback function is called. It is used to reduce the learning rate when the metric stop improving. It monitors validation accuracy if it is not improving at 3 epochs than it is decreased by factor 0.2. If the learning rate is reduced to 0.0001 after this if it is not improving then it will not reduce the learning rate further. The CNN is implemented in Python with keras and tensorflow backend.

- **The proposed CNN architecture for the input of MFCC feature is parameterized as follows:**

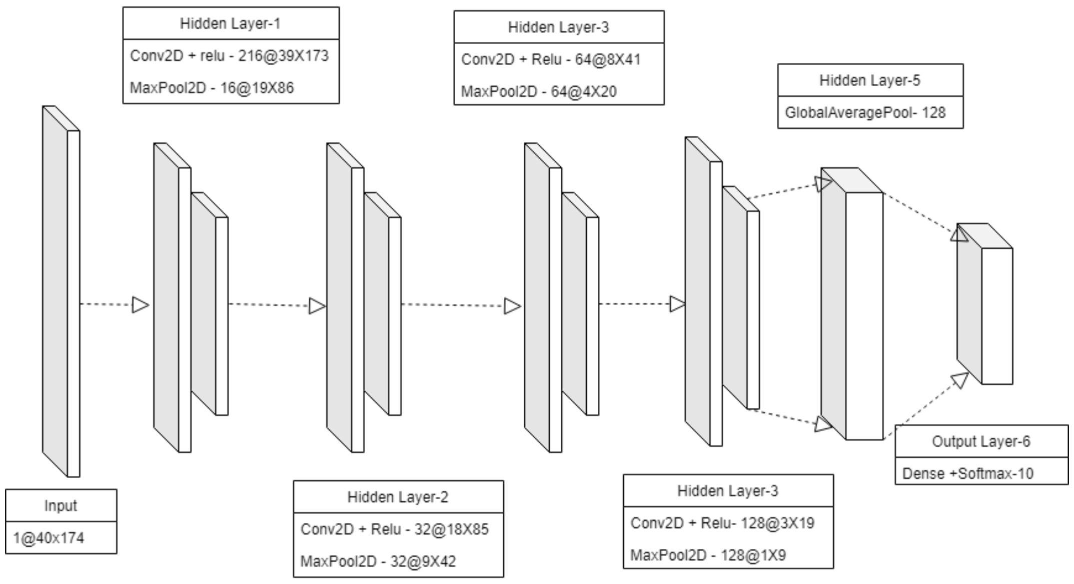


Figure 2.7: Architecture of MFCC-CNN model

Layer 1:- It consists of Conv2D convolutional layer having kernel size (2,2), filters 16, input shape of (40,174,1) and ReLU activation function. This is followed by Maxpooling layer of pool size (2,2).
Layer 2:- It consists of Conv2D convolutional layer having kernel size (2,2), filters 32 and ReLU activation function. This is followed by Maxpooling layer of pool size (2,2).
Layer 3:- It consists of Conv2D convolutional layer having kernel size (2,2), filters 64 and ReLU activation function. This is followed by Maxpooling layer of pool size (2,2).
Layer 4:- It consists of Conv2D convolutional layer having kernel size (2,2), filters 128 and ReLU activation function. This is followed by Maxpooling layer of pool size (2,2).
Layer 5:- It consists of GlobalAveragePooling2D layer.
Layer 6:- It consists of Dense layer having 10 units for 10 classes and Softmax activation function.

Figure 2.7 shows the architecture to train the audio MFCC feature. The model optimizes cross-entropy loss via Adaptive moment estimation [21]. Adam is set at learning rate 0.001, beta-1 0.9 and beta-2 0.999. Each batch consists of 128 TF-patches randomly selected from the training data (without repetition).The model is trained for 15 epochs. A validation set is used to identify the parameter setting (epoch) achieving the highest classification accuracy. The CNN is implemented in Python with keras and tensorflow backend.

### 2.3.3 Data SpecAugmentation

As we discussed about different type of SpecAugment, In this we applyed all those type. Using time masking and frequency masking we augment the data by masking the MelSpec TF-patches. Originally, there are 7,467 MelSpec pattern of dataset. By augmenting, we generated 7,467 MelSpec augmented pattern for each masking method.

After generating the syntactic data for audio, it is trained along with original audio data in the same CNN model which is defined for MelSpec feature .For both masking methods, model train for 50 epochs having batch size 64.

- **Time masking:-**In this, we apply time based masking to spectrogram input. Mask factor will be picked randomly. Mask range will be between [0, tau - master factor) while tau is time range of input. In this project, mask factor is 80. Fig 2.8 represents the time masked melspectrogram representation of dog bark audio.

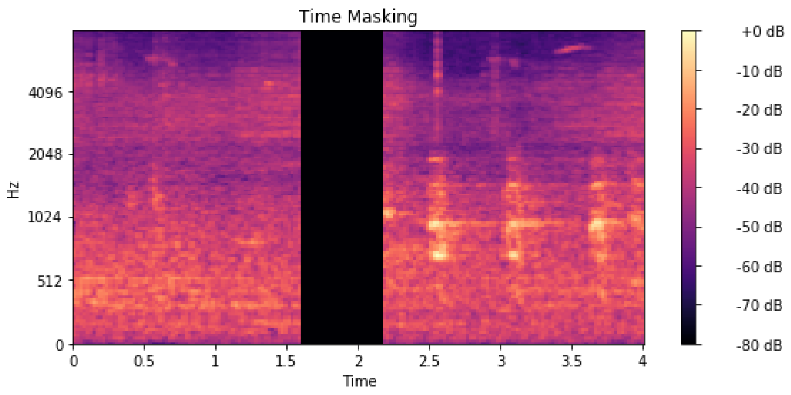


Figure 2.8: Time masked MelSpec representation of dog bark audio

- **Frequency masking:-**In this, we apply frequency based masking to spectrogram input. Mask factor will be picked randomly. Mask range will be between [0, v - master factor) while v is the number of mel frequency channels. In this project, mask factor is 80. Fig 2.9 represents the frequency masked melspectrogram representation of dog bark audio.

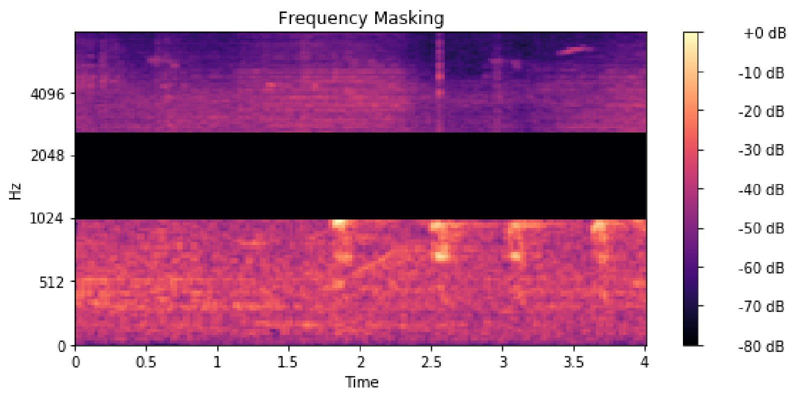


Figure 2.9: Frequency masked MelSpec representation of dog bark audio

# CHAPTER 3

# RESULTS AND DISCUSSIONS

## 3.1 Results

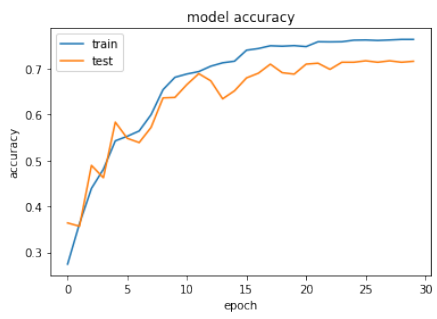
The models are evaluated in a 10-fold (UrbanSound8K) cross-validation regime. CNN models are trained and test for two sets of features. Their result are discussed below through learning curves for a deep learning model during training on both the training and validation datasets and the confusion matrix reflects the efficiency of each class on the test data. In several circumstances, it is prevalent to produce learning curves for numerous metrics, such as for classification problems with predictive modelling, where the model can be optimized using classification accuracy based on cross-entropy loss and model efficiency assessment. In this case, two plots are created, one for the learning curves of each metric.

- Optimization Learning Curves: This is based on the metric by which the model's parameters are optimized, e.g. loss.
- Performance Learning Curves: This is based on the metric by which the model will be evaluated and selected, e.g. accuracy.

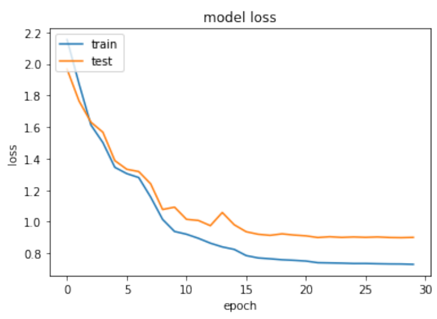
For each experiment, their performance and optimization learning curve are present at the left(a) and the right(b).

### 3.1.1 Trained CNN model using MelSpec feature

The classification accuracy of the proposed CNN model for MelSpec feature on trained and test datasets are 0.79 and 0.73. Their corresponsing dual learning curve and confusion matrix presented in Fig. 3.1 and 3.2. Fig 3.1(a) shows that for 30 epochs, there is overfitting of 6%. From fig 3.2, we find that chirldren playing, street music and siren classes affect each other adversely.



(a)



(b)

Figure 3.1: Accuracy and loss curve of MelSpec feature

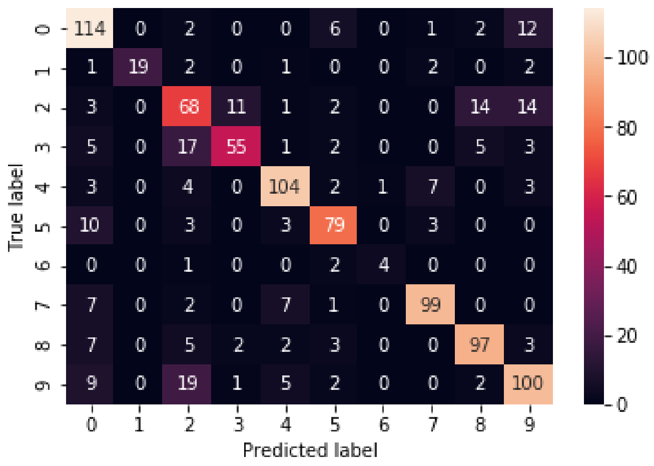
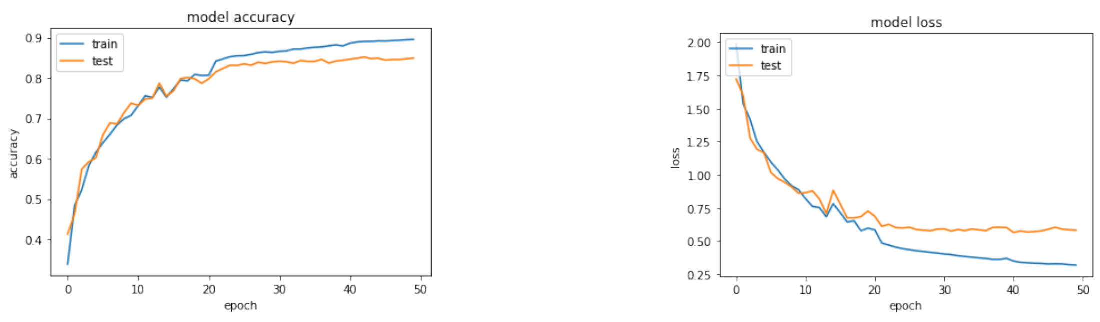


Figure 3.2: Confusion matrix on MelSpec feature for the proposed CNN model

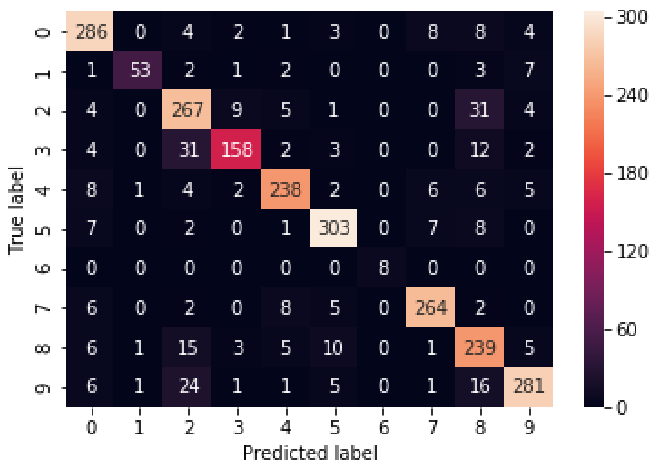
After this, we tested for frequecy masked and time masked SpegAugment augmented data for Melspec feature. With FM SpecAugment, classification accuracy of the proposed CNN model on trained and test datasets are 0.90 and 0.86. Their corresponsing dual learning curve and confusion matrix presented in Fig. 3.3 and 3.4. Fig 3.3(a) shows that model train 50 epochs which is higher as compare to melspec feature result but it reduced the overfitting to 4%. From fig 3.4, we found that children playing is affected by siren and street music is affected by children playing class.

With TM SpecAugment, classification accuracy of the proposed CNN model on trained and test datasets are 0.90 and 0.85. Their corresponsing dual learning curve and confusion matrix presented in Fig. 3.5 and 3.6. Fig 3.5(a) shows that model train 50 epochs which is higher as compare to melspec feature result but it reduced the overfitting to 5%. From fig 3.6, we found the same case as in fig 3.4.



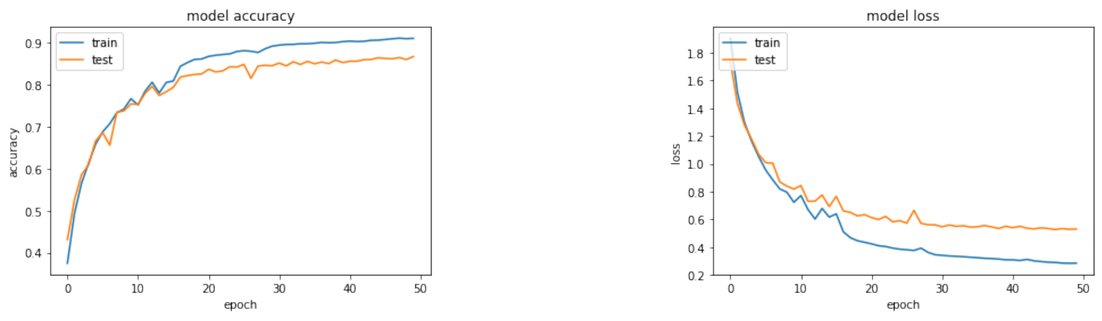
(a) (b)

Figure 3.3: Accuracy and loss curve of MelSpec feature with FM SpecAugment



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 286 | 0 | 4 | 2 | 1 | 3 | 0 | 8 | 8 | 4 |
| 1 | 1 | 53 | 2 | 1 | 2 | 0 | 0 | 0 | 3 | 7 |
| 2 | 4 | 0 | 267 | 9 | 5 | 1 | 0 | 0 | 31 | 4 |
| 3 | 4 | 0 | 31 | 158 | 2 | 3 | 0 | 0 | 12 | 2 |
| 4 | 8 | 1 | 4 | 2 | 238 | 2 | 0 | 6 | 6 | 5 |
| 5 | 7 | 0 | 2 | 0 | 1 | 303 | 0 | 7 | 8 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 |
| 7 | 6 | 0 | 2 | 0 | 8 | 5 | 0 | 264 | 2 | 0 |
| 8 | 6 | 1 | 15 | 3 | 5 | 10 | 0 | 1 | 239 | 5 |
| 9 | 6 | 1 | 24 | 1 | 1 | 5 | 0 | 1 | 16 | 281 |

Figure 3.4: Confusion matrix on MelSpec feature with their FM SpecAugment for the proposed CNN model



(a) (b)

Figure 3.5: Accuracy and loss curve of MelSpec feature with TM SpecAugment

### 3.1.2 Trained CNN model using MFCC feature

The classification accuracy of the proposed CNN model for MFCC feature on trained and test datasets are 0.91 and 0.89. Their corresponsing dual learning curve and confusion matrix presented in Fig. 3.7 and 3.8. Fig 3.7(a) shows that for 15 epochs, the problem of overfitting is almost gone. From fig 3.8, we find that chirldren playing is still affected by street music.

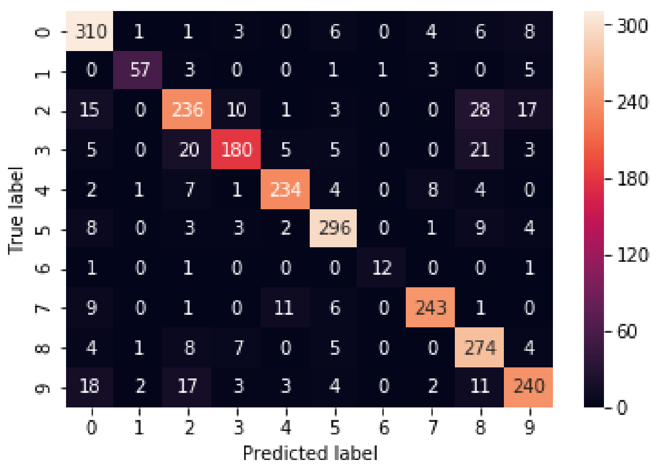
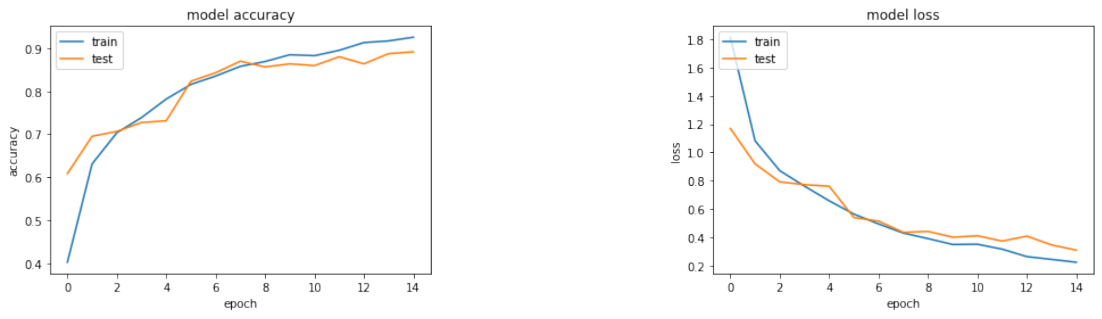


Figure 3.6: Confusion matrix on MelSpec feature with their TM SpecAugment for the proposed CNN model



(a) (b)

Figure 3.7: Accuracy and loss curve of MelSpec feature

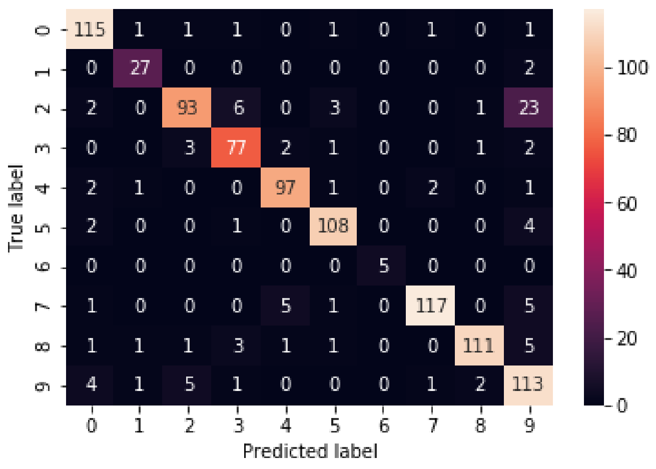
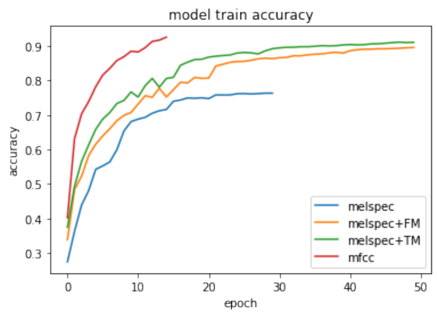


Figure 3.8: Confusion matrix on MFCC feature for the proposed CNN model

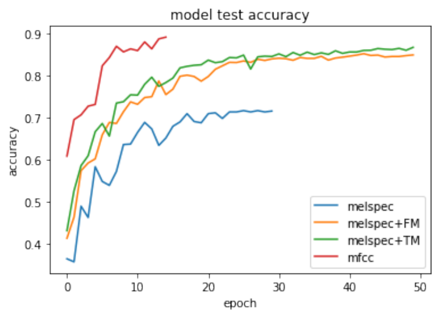
## 3.2 Discussion

From the results which we have mentioned above and also summarized in table 3.1, There is an increase in the accuracy and drastic decrease in the loss from without augmentation to augmentation of frequency masking and time masking for MelSpec TF-patches. Due to FM SpecAugment augmentation, accuracy for train and test increase by 9% and 10% ,which is almost same case for TM SpecAugment augmentation.
We also observed that MFCC outperformed all results which is achieved through MelSpec and their augmented features. There is a drastic increase in accuracy of the train and test scores ( 12% and 15% compared to previous result). Though the difference remains low so the model has not suffered from overfitting.

Figure 3.9(a) shows the performance curve of train data for different feature. It clearly shows that MFCC performed well as compare to other features. After MFCC, both augmented feature set of MelSpec dominated MelSpec without augmentation. Figure 3.9(b) shows the same results for test accuracy.



(a)



(b)

Figure 3.9: Accuracy curve of for different feature sets

| Model | Train Accuracy | Test Accuracy |
|---|---|---|
| LP-CNN [22] | 0.73 | - |
| SB-CNN [6] without augmentation | 0.74 | - |
| SB-CNN [6] with augmentation | 0.79 | - |
| MelSpec CNN | 0.79 | 0.73 |
| MelSpec+FM SpecAugment CNN | 0.90 | 0.86 |
| MelSpec+TM SpecAugment CNN | 0.90 | 0.85 |
| MFCC | 0.91 | 0.89 |

Table 3.1: classification accuracy for different sets of feature and their CNN model

Table 3.2 shows the individual class performance, 0 to 9 represents the class ID and rest four columns represent per-class accuracy for different sets of features and their corresponding CNN model. It seems that for all classes there is increase in accuracy.

For class ID 2,3 and 6 which represents children playing ,dog bark and gun shot, there is a very bad performance of model for MelSpec feature but has a very good performance for MFCC feature. For almost same model complexity, it is found that MFCC feature is able to classify these noisy sounds perfectly as compare to MelSpec feature. And for same model, Augmented features help model in better learning as compare to without augmented MelSpec features.

| Class ID | SB-CNN [6] | MelSpec | MelSpec+FM | MelSpec+TM | MFCC |
|---|---|---|---|---|---|
| 0 | 0.49 | 0.83 | 0.90 | 0.91 | 0.95 |
| 1 | 0.90 | 0.70 | 0.76 | 0.81 | 0.93 |
| 2 | 0.83 | 0.60 | 0.83 | 0.76 | 0.72 |
| 3 | 0.90 | 0.62 | 0.74 | 0.75 | 0.89 |
| 4 | 0.80 | 0.83 | 0.87 | 0.89 | 0.93 |
| 5 | 0.80 | 0.80 | 0.92 | 0.90 | 0.93 |
| 6 | 0.94 | 0.57 | 1.00 | 0.80 | 1.00 |
| 7 | 0.68 | 0.85 | 0.91 | 0.89 | 0.90 |
| 8 | 0.85 | 0.81 | 0.83 | 0.90 | 0.89 |
| 9 | 0.84 | 0.72 | 0.83 | 0.80 | 0.88 |

Table 3.2: Per-class accuracies for different sets of feature and their CNN model

# CHAPTER 4

# CONCLUSION AND FUTURE WORKS

In this, we have proposed two deep CNN architecture. One for MFCC feature and another for Melspec feature of audio. Second architecture in combination with a set of audio SpecAugment augmentation, produces state-of-the-art results for environmental sound classification. Application of Deep learning in this field has certainly exhibited tremendous results. Deep Learning has increased efficiency in enormous aspects of life including healthcare, transportation etc. Advancement in deep learning has been positively influenced by increasing amount of data, GPU technology, decrease in prices and improved techniques.

CNN has shown to be an excellent feature extractor from Melspec and MFCC visual representation of audio. We have presented the detailed architecture including the design of parameters and layers. Finally we have performed enough experiments on UrbanSound8K Dataset to test our CNN architecture with augmentation and without augmentation. For MelSpec feature, we found that our CNN architecture with SpecAugment augmentation gave better accuracy as compared to without augmentation. On the other side, CNN architecture for MFCC feature performed well and gave better accuracy without augmentation. Experimental results have not only shown an increase in accuracy but also gave an insight that MFCC feature representation worked well as compared to Melspec feature.

A hybrid architecture by combining Recurrent Neural Networks (RNN) with Convolutional neural network (CNN) can be used for classification purpose. It can also integrated in smart devices for assisting deaf individuals in their daily activities and also in industries for predictive maintenance which will be an extended work of this project because the problem of low accuracy is removed at much level.

# REFERENCES

[1] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[2] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, Oct 2014.

[3] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for lvcsr. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8614–8618, May 2013.

[4] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Convolutional neural networks for large scale remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):645–657, Feb 2017.

[5] Justin Salamon and Juan Bello. Unsupervised feature learning for urban sound classification. 04 2015.

[6] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, March 2017.

[7] Yi Wang and Bob Lawlor. Speaker recognition based on mfcc and bp neural networks. pages 1–4, 06 2017.

[8] B. McFee, E. Humphrey, and Juan Bello. *A Software Framework for Musical Data Augmentation*. 10 2015.

[9] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963, Aug 2003.

[10] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *arXiv e-prints*, page arXiv:1904.08779, Apr 2019.

[11] J. Skoglund and W. B. Kleijn. On time frequency masking in voiced speech. *IEEE Transactions on Speech and Audio Processing*, 8(4):361–369, July 2000.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to

document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[13] Justin Salamon, Christopher Jacoby, and Juan Bello. A dataset and taxonomy for urban sound research. pages 1041–1044, 11 2014.

[14] S. Chu, S. Narayanan, and C. . J. Kuo. Environmental sound recognition with time frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1142–1158, Aug 2009.

[15] M. Asgari, I. Shafran, and A. Bayestehtashk. Inferring social contexts from audio recordings using deep neural networks. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2014.

[16] Lawrence R Rabiner and Ronald W Schafer. *Theory and applications of digital speech processing*, volume 64. Pearson Upper Saddle River, NJ, 2011.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[19] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt Mcvicar, Eric Battenberg, and Oriol Nieto. *librosa: Audio and Music Signal Analysis in Python*. 01 2015.

[20] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *19th Int. Conf. Comput. Statist., Paris, France*, pages 177–186, Aug 2010.

[21] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[22] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2015.