Major project Report

On

Acoustic Scene Classification using Convolutional Neural Network

By

Adhyyan Tripathi (201700403)
Lakshit Agarwal (201700382)

Group Id – 04

In the partial fulfilment of requirements for the award of degree in Bachelor of Technology in Computer Science and Engineering

(2020)

Under the Project Guidance of

Dr. Biswaraj Sen

Sikkim Manipal Institute of Technology, Majitar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent college of Sikkim Manipal University) MAJITAR, RANGPO, EAST SIKKIM – 737136

Project Completion Certificate

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from 27 Feb 2021 to 30 Jun 2021 and have successfully completed the project entitled “Acoustic Scene Classification using Convolutional Neural Network” in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student(s)	Course
201700403	Adhyyan Tripathi	CSE
201700382	Lakshit Agarwal	CSE

Dr. Biswaraj Sen
Designation
Sikkim Manipal Institute of Technology
Majitar, East Sikkim – 737136.

Project Review Certificate

This is to certify that the work recorded in this project report entitled **"Acoustic Scene Classification using Convolutional Neural Network"** has been jointly carried out by **Mr. Adhyyan Tripathi (Reg 201700403)** and **Mr. Lakshit Agarwal (Reg 201700382)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Major Project Viva Examination.

Dr. Biswaraj Sen
Designation
Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology
Majitar, East Sikkim – 737136.

Certificate of Acceptance

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Dr. Biswaraj Sen** of **Sikkim Manipal Institute of Technology Majitar, East Sikkim** from **27 Feb 2021** to **30 Jun 2021** on the project entitled **"Acoustic Scene Classification using Convolutional Neural Network"**. The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student(s)	Project Venue
201700403	Adhyyan Tripathi	SMIT
201700382	Lakshit Agarwal	SMIT

Dr Kalpana Sharma
Professor & HOD
Computer Science & Engineering Department
Sikkim Manipal Institute of Technology
Majhitar, Sikkim – 737136

Declaration

We the undersigned, hereby declare that the work recorded in this project report entitled **"Acoustic Scene Classification using Convolutional Neural Network"** in partial fulfillment for the requirements of award of B.Tech in Computer Science & Engineering from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at "**Sikkim Manipal Institute of Technology Majhitar, Sikkim"** under the supervision and guidance of **Dr. Biswaraj Sen** of **Sikkim Manipal Institute of Technology Majhitar, Sikkim.**

The results of this investigation reported in this project have so far not been reported for any other Degree / Diploma or any other Technical forum. The assistance and help received during the course of the investigation have been duly acknowledged.

Mr. Adhyyan Tripathi (Reg 201700403)

Mr. Lakshit Agarwal (Reg 201700382)

Acknowledgement

We take this opportunity to acknowledge indebtedness and a deep sense of gratitude to my guide **Dr. Biswaraj Sen** whose valuable guidance and kind supervision gave us throughout the course which shaped the present work as it shows.

We pay our deep sense of gratitude to **Prof. (Dr.) Kalpana Sharma**, H.O.D, Computer Science & Engineering Department, **Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and provide all support required.

We obliged to our project coordinators **Mr .Tawal Kumar Koirala, Mr. Dhruba Ningombam, Ms. Nitisha Pradhan, Mr. Nitai Paitya, Mr. Debanjan Konar and Mr. Sunil Dhimal** for elevating, inspiration and kind supervision in completion of our project.

We would like to specially thank **Mr. Vikash Kumar Singh** for helping us with the project.

We would also like to thank any other staff of the Computer Science & Engineering Department, Sikkim Manipal Institute of Technology for giving us continuous support and guidance that has helped us in completion of our project.

Mr. Adhyyan Tripathi (Reg 201700403)

Mr. Lakshit Agarwal (Reg 201700382)

DOCUMENT CONTROL SHEET

1	Report No	CSE/Major Project/Internal/B.Tech/04/2021
2	Title of the Report	Acoustic Scene Classification using Convolutional Neural Network
3	Type of Report	Technical
4	Author(s)	Mr Adhyyan Tripathi (Reg 201700403) , Mr Lakshit Agarwal (Reg 201700382)
5	Organizing Unit	Sikkim Manipal Institute of Technology
6	Language of the Document	English
7	Abstract	The project targets to accurately classify various acoustic scenes from one another using deep learning methodology.
8	Security Classification	General
9	Distribution Statement	General

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The project targets to accurately classify various acoustic scenes from one another using deep learning methodology. Acoustic scenes can be defined by various ambient sounds in the environment. An acoustic scene for example indoor shopping malls can be defined by a combination of various sounds such as people talking to one another, ambient noise of elevators and escalators, an announcement or soft music etc. The deep learning model will try to classify such urban acoustic scenes as accurately as possible.

As of now there are no proven audio classifiers, and it is currently an active field of research. Acoustic scene classification has only become popular in the past few years and there are not a lot of algorithms to accurately classify audio. The method to tackle this is by converting the acoustic scene classification problem into a problem which has already received much attention. One of the most researched advanced fields of deep learning is image classification, so it makes sense to convert this audio classification problem into an image classification problem, which we target to do in this project.

1. INTRODUCTION

1.1 General Overview of the problem

Acoustic scene classification (ASC) is an active research area which has recently gained attention. Acoustic scenes can be defined by various ambient sounds in the environment. An acoustic scene for example indoor shopping malls can be defined by a combination of various sounds such as people talking to one another, ambient noise of elevators and escalators, an announcement or soft music etc.

Why bother to classify acoustic scenes? It is because Acoustic scene classification can be used in multiple areas such as - for example searching for multimedia based on its audio content, robots, cars, an intrusion detection system which takes advantage of both audio and video feeds, and etc.

This project can be categorized into 3 main stages – Feature extraction, model construction and model evaluation and comparison. The feature extraction part deals with the selection of the most appropriate feature to train the model. The model construction deals with building the most optimized classifiers which can classify accurately most of the acoustic scenes from one another, and then compare that optimized model with the already existing models.

1.2 Literature Survey

Author / Year / Name of Journal or Conference	Title	Finding	Relevance
Kong, Q., Cao Y., Iqbal, T., Xu, Y., Wang, W., & Plumbley. M. (2019, April). In Detection and Classification of Acoustic Scenes and Events 2019.	Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems.	Analysis of 5, 9 and 13 layer CNN architecture on all tasks / subtasks of DCASE 2019 challenge.	CNN-9 architecture can be used as it performs better than CNN-5 and CNN-13 architecture in ASC task.
Mesaros, A., Heittola, T., & Virtanen, T. (2018, July). In Detection and Classification of Acoustic Scenes and Events 2018.	A multi-device dataset for urban acoustic scene classification	Dataset containing multiple classes of acoustic scenes.	The dataset is used to train the model.
Heittola, T., Mesaros, A., & Virtanen, T. (2019).	TAU Urban Acoustic Scenes 2019,	The dataset contains multiple classes of acoustic scene.	The dataset to be used for training the model.

TAU Urban Acoustic Scenes 2019, Development dataset	Development dataset		
Geiger, J. T., Schuller, B., & Rigoll, G. (2013). IEEE Workshop on Applications of Signal Processing to Audio and Acoustics	Large-scale audio feature extraction and SVM for acoustic scene classification.	Dataset containing multiple classes of acoustic scenes.	SVM can be used in conjunction with CNN, both trained on separate features
Nisar, S., Khan, O., U., and Tariq, M. (2016, August) Princeton University, New Jersey, NJ 08544, USA.	An Efficient Adaptive Window Size Selection Method for Improving Spectrogram Visualization.	Tuned parameters of Short-time Fourier transform (STFT).	Improved spectrogram for more efficient learning.

1.3 Problem Definition

Sounds carry a large amount of information about our everyday environment and physical events that take place in it. We can perceive the sound scene we are within (busy street, office, etc.), and recognize individual sound sources (car passing by, footsteps, etc.). The research work in this field is very limited for now, so the number of algorithms to classify acoustic scenes accurately are very less. Carrying out this task with the help of deep learning algorithms is the main agenda of this project.
Few challenges that make it harder for machines to classify a acoustic scene:

- High Inter-class similarity
- High intra-class variance
- Too difficult to characterize between background and foreground noise

1.4 Software Requirements And Specifications

- Hardware Specification of Developing Environment

  Processor: i7-7700HQ

  RAM: 16 GB

  Hard Drive: 1 TB

  VRAM: 4GB (Nvidia GeForce GTX 1050).

- Software Specification

  Python 3.8.5

  Librosa (Python library).

  TensorFlow and keras.

1.5 Proposed Solution Strategy

Using a log-mel spectrogram as a main feature for the model since temporal and spec-tral structures of acoustic scenes give a good visual representation and can be seen as 2D spectrograms. Due to the lack of algorithms to solely classify an audio sample, the audio classification problem is converted into image classification problem.

The dataset provides different classes of audio files of 3-5 minutes each. The data is augmented in order to prevent overfitting of the model by segmenting each audio file.

The log-mel spectrogram for each audio recording will be dumped into image. Once the conversion is done, the Convolutional Neural Network (CNN) will then be trained to classify each image into it's correct acoustic scene class.

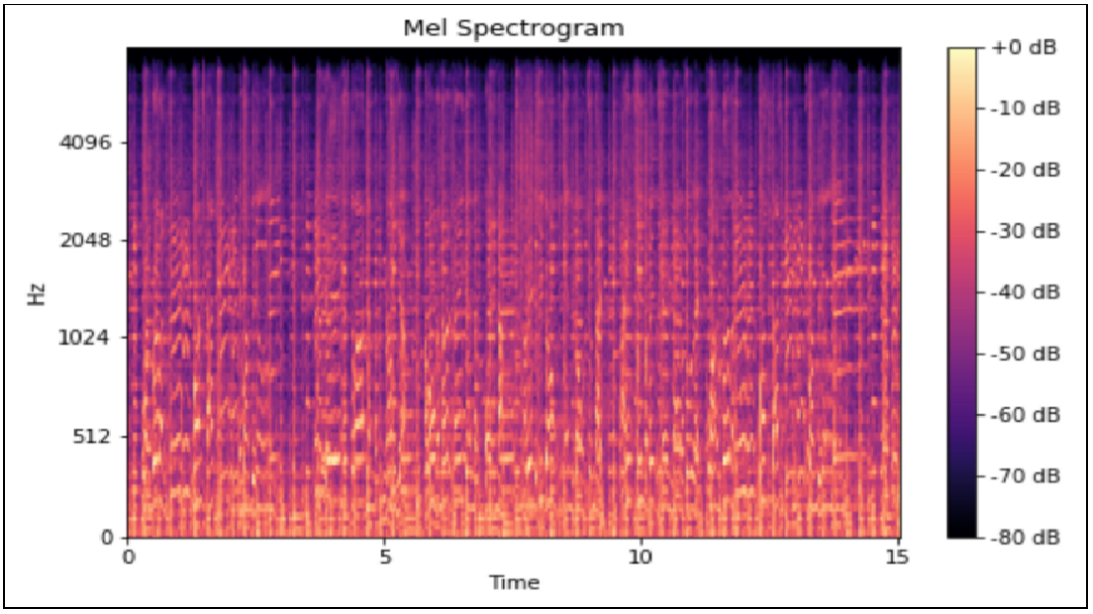


Fig – 1 Example log-mel spectrogram

2. DESIGN STRATEGY FOR SOLUTION

2.1 Block diagram

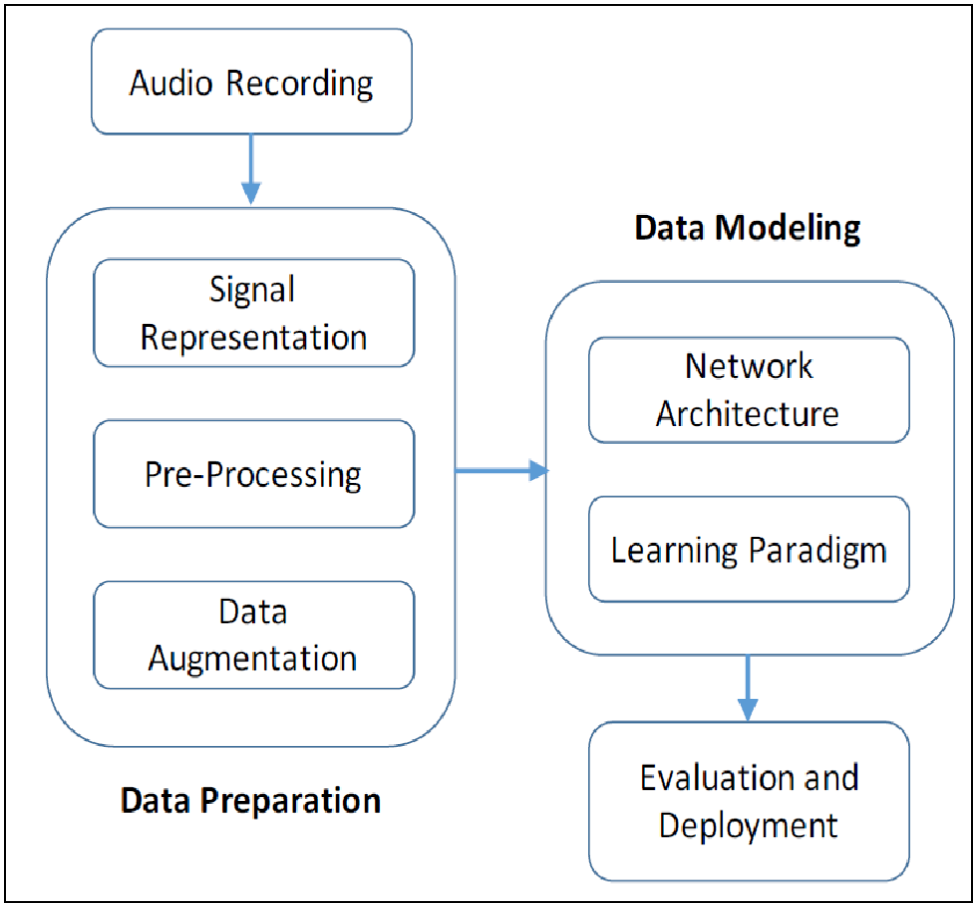


Fig – 2 Block diagram of Acoustic scene classification system.

A Convolutional Neural Network model will be developed that will differentiate different Acoustic Scenes. The audio recordings will be taken from a given dataset and will be transformed to its mel-spectrogram representation. These pre-processed images will then be fed to the CNN model for classification.

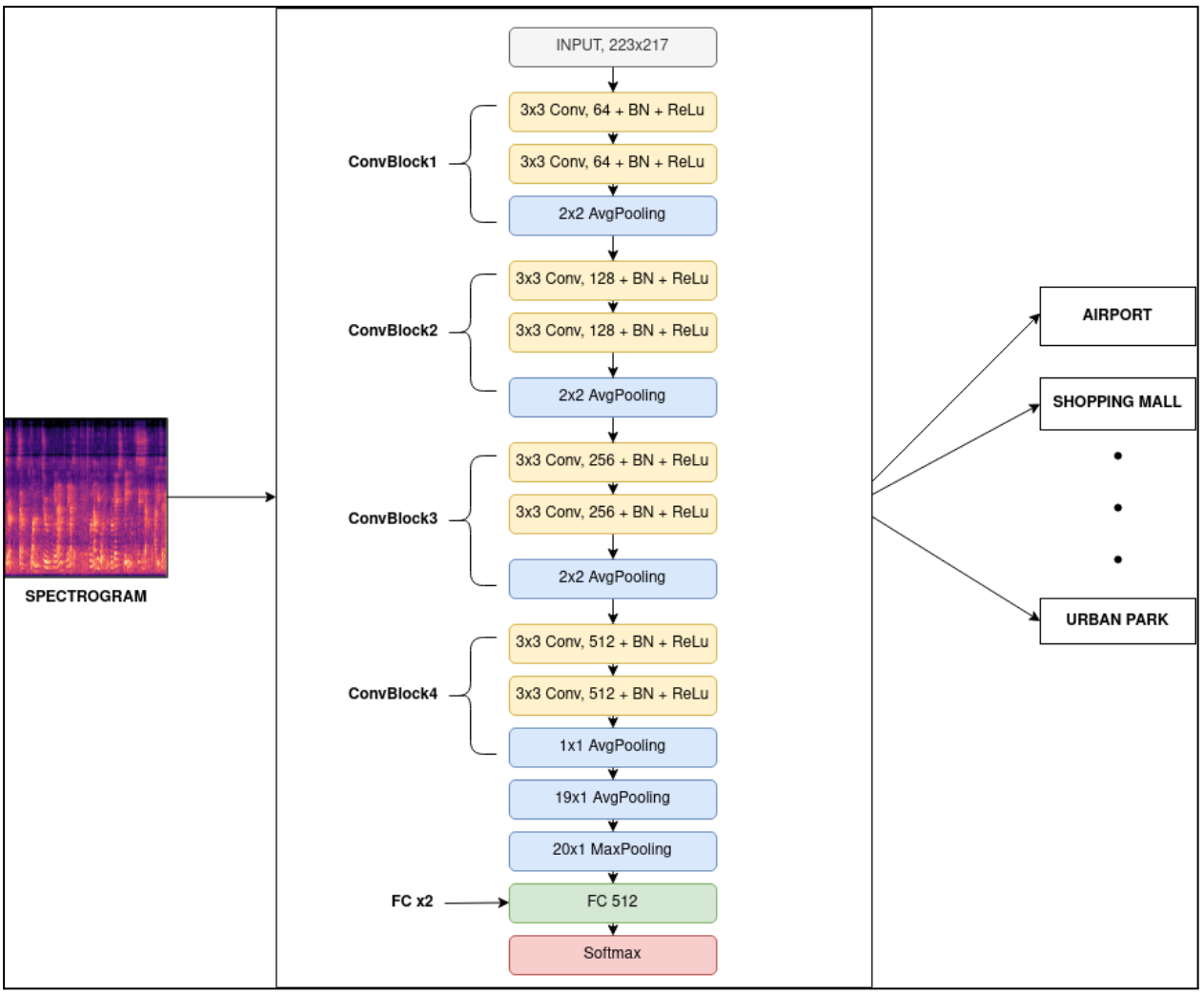


Fig - 3 modified CNN-9 architecture

This is the modified 9-layer CNN architecture similar to the one proposed for "Cross-Task Learning for Audio Tagging, Sound Event Detection and Spatial Localization DCASE 2019 Baseline Systems" with modified parameters.

The transformed spectrogram images are of dimension 223x217.

3. METHODOLOGY

3.1 Dataset

The dataset consists of 10 different acoustic scenes which are –

- Airport - airport
- Indoor shopping mall - shopping_mall
- Metro station - metro_station
- Pedestrian street - street_pedestrian
- Public square - public_square
- Street with medium level of traffic - street_traffic
- Travelling by a tram - tram
- Travelling by a bus - bus
- Travelling by an underground metro - metro
- Urban park – park

All of the acoustic scenes are recorded with 4 different audio recording devices, also the data is recorded from 12 different European cities.
The cities from which this audio is recorded are - Amsterdam, Barcelona, Helsinki, Lisbon, London, Lyon, Madrid, Milan, Prague, Paris, Stockholm and Vienna.
The audio recordings are of total 64 hours, device A recorded for 40 hours, device B and C both for 3 hours each, device C is simulated audio with 14 hours of recording. Audio in the dataset is provided in a single-channel 44.1kHz 24-bit format.
[3]The dataset was collected by Tampere University of Technology between 05/2018 - 11/2018. The data collection received funding from the European Research Council, grant agreement 637422 EVERY SOUND.

3.2 Audio Features

Audio features can be categorized into two major domains which are as follows –

- Time Domain
- Frequency Domain

Time Domain Features – Time domain features or temporal features are easy to implement, some of the time domain features are - energy of signal, zero crossing rate, maximum amplitude, minimum energy, etc. The easy implementation is an advantage of EMG signals but a major disadvantage of time domain features comes from a non-stationary property of the EMG signal, changing in statistical properties over time, but time domain features assume the data as a stationary signal[7].

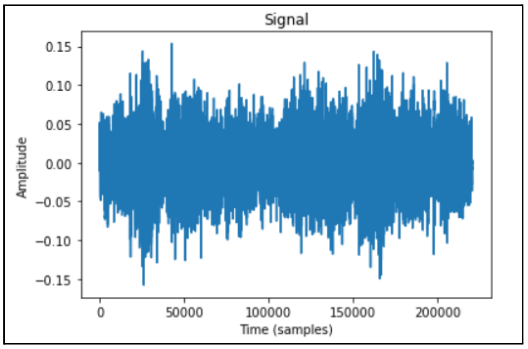


Fig-4 Amplitude

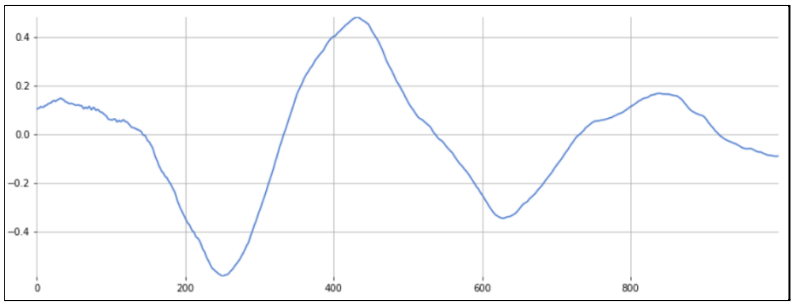


Fig-5 Zero crossing rate (5)

Frequency Domain Features – Frequency domain features or spectral features are harder to implement than Time domain features. The frequency lets us view signals in a different domain which is the frequency vs time domain, some of the examples of frequency domain are - spectral flux, spectral density, Mel-frequency cepstral coefficient (MFCC), etc.

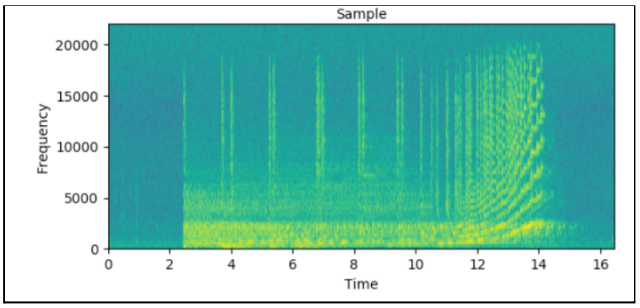


Fig-6 Spectrogram

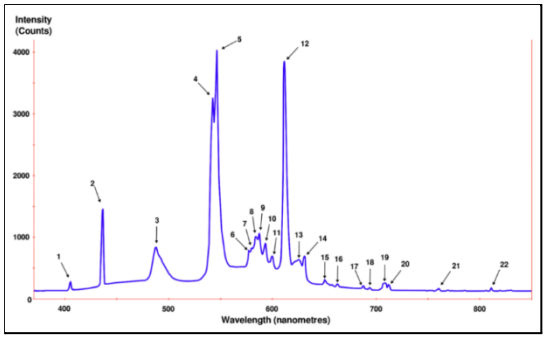


Fig-7 spectral density

Some of the features over which researches are carried out are - Mel-frequency cepstral coefficient (MFCC) which is a spectral feature, zero crossing rate which is a temporal feature and spectrum which is a spectral feature.

3.3 Librosa Library (Feature Extraction)

Librosa is a library specifically developed for handling audio files and extracting various spectral and temporal features. The core functionality of librosa [1] includes a range of commonly
used functions. Broadly, core functionality falls into four categories: audio and time-series operations, spectrogram calculation, time and frequency conversion, and pitch operations. We have used this library for spectrogram calculation and various audio and time-series operations.

Spectrogram: Spectrogram is a frequency domain feature which can be obtained from audio files by applying short time Fourier transformation on an overlapping windowed audio signal with a certain number of hops. Librosa provides functions to calculate and plot spectrograms of each audio file, or if the developer wants to plot it using STFT, it can do so.

Short-Time Fourier Transformation (STFT) : There are two types of short-time fourier transformation discrete and continuous.

- Continuous-time Short-Time Fourier Transformation -

Simply put, the function to be converted is multiplied by a window function that is nonzero for just a brief period of time in the continuous-time situation. As the window is moved down the time axis, the Fourier transform (a one-dimensional function) of the resultant signal is obtained, resulting in a two-dimensional representation of the signal. This is written mathematically as:

$$\mathbf{STFT}\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}\,dt$$

Fig-8 STFT Equation-1

Where w(t) is any window function, x(t) is the signal which needs to be transformed.

- Discrete-time Short-Time Fourier Transformation - The data to be changed in discrete time might be split up into chunks or frames (which usually overlap each other, to reduce artefacts at the boundary). Each chunk is Fourier converted, and the resulting complex is added to a matrix that stores magnitude and phase for each time and frequency point. This may be stated as follows:

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n}$$

Fig-9 STFT Equation-2

Mel-Scale: Mel-scale is a scale which is used so that two or more distant audio signals sound equally distant to the listener as well. The name mel comes from the melody to indicate that the scale is based on pitch.

The formula to convert the frequency into mel is –

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

Fig-10 Mel Formula

The unit after converting the frequency from mel-scale is ‘mel-bands’ or ‘bins’.

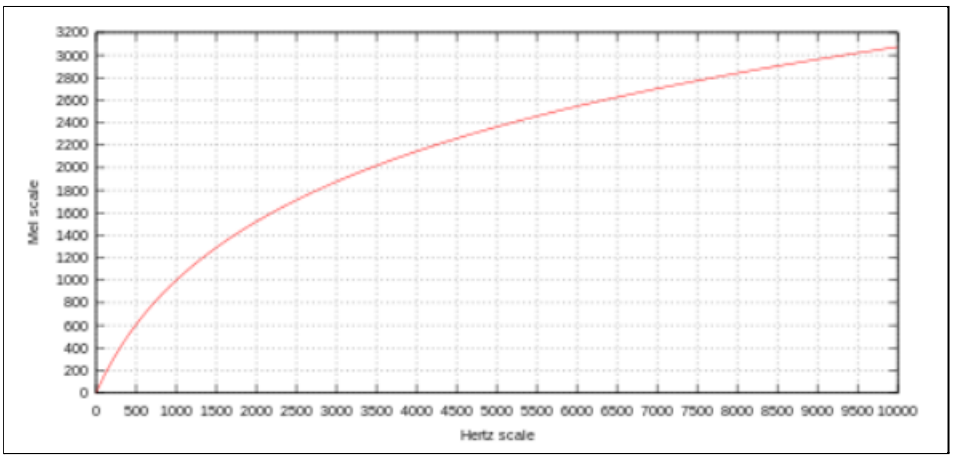


Fig-11 Mel-scale

Mel-spectrogram: Mel-spectrogram is a spectral feature which is obtained by applying short time Fourier transformation on an overlapping windowed of audio signal with a certain number of hops, and then passing it to through the mel-scale and finally converting them into logarithmic scale, gives us the log-mel spectrogram. We are using a window size of 2048ms with 512ms of hop and the number of mel-bands or bins is 128.

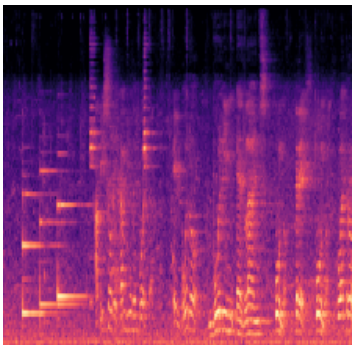


Fig -12 Log-mel spectrogram

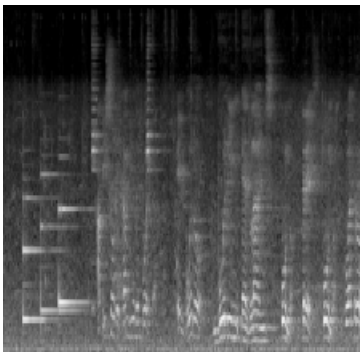


Fig -13 Single channel log-mel spectrogram

4. IMPLEMENTATION DETAILS

Phase-1

Algorithm : Convert audio files in the dataset to its corresponding mel-spectrogram

Input : Development Dataset(D1) = [A1, A2, … , An], where Ai = audio file & i -> 1 to 23040

Output : mel-spectrogram with dimensions 223x217.

Start

Step 1: For each file in dataset D1

Step 1.1: load audio file to a variable

Step 1.2: trim the loaded audio file

Step 1.3: generate mel-spectrogram

Step 1.4: convert mel-spectrogram to log scale

Step 2: Store images of size 223x217

Stop

4.1 Image Dataset

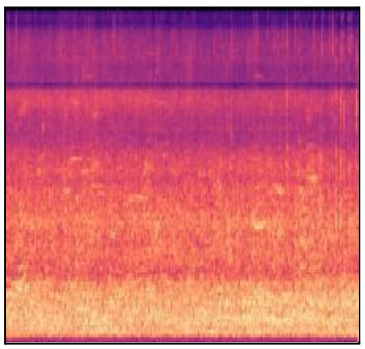


Fig-14 Airport

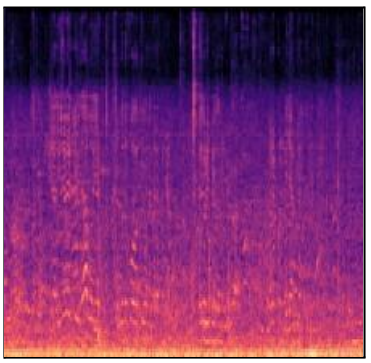


Fig-15 Indoor Shopping Mall

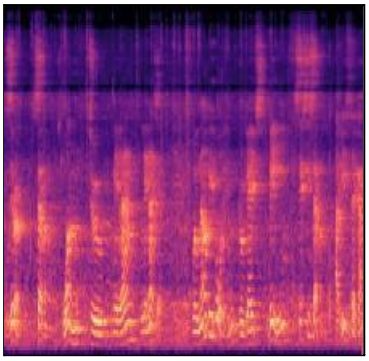


Fig-16 Metro Station

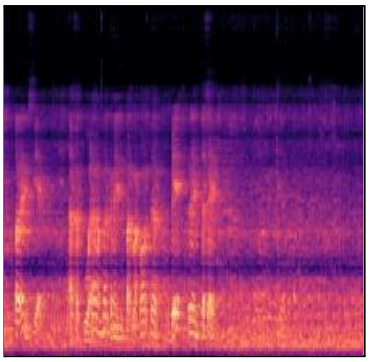


Fig-17 Public Square

The images above are generated by applying the short time fourier transformation and filtering. The dimensions of each image is 223 x 217.

4.2 Model Implementation

The model is implemented by keeping the CNN-9 architecture as its base, and tweaking some of the parameters according to our dataset requirements. The input layer of our model is 223x217x1 which is then followed by the 3x3 convolution layer with 64 filters, batch normalized layer and relu activation function, finally an average pooling layer of size 2x2. This is the first block of the model named Block-1. Similarly, there are in total 4 blocks each having slightly different configurations. Block-2 has 128 filters instead of 64. Block-3 has 256 filters, and block 4 has 512 filters with 1x1 average pooling layer. The 4 blocks are then followed by 20x1 average pooling layer and a 1x19 max pooling layer, the images are then flatten and passed through a dense layer with 512 filters and with linear activation function, and then to a dense layer with 10 neurons with softmax.

Phase-2

Input: dataset(D1)

Start

Step 1: Shuffle dataset

Step 2: Split into training and validation

Step 3: Reshape for CNN input

Step 4: One-Hot encoding for classes

Stop

Phase-3

Start

Step 1: Define the model

Stop

Phase-4 (Model training)

Start

Step 1: model compilation with Adam optimizer and categorical_crossentropy as loss function.

Step 2: starting training with epoch size=100 and batch_size=200.

Stop

Phase-5 (Model evaluation)

Start

Step 1: shuffle dataset D2

Step 2: Split into training and validation

Step 3: Reshape for CNN input

Step 4: One-Hot encoding for classes

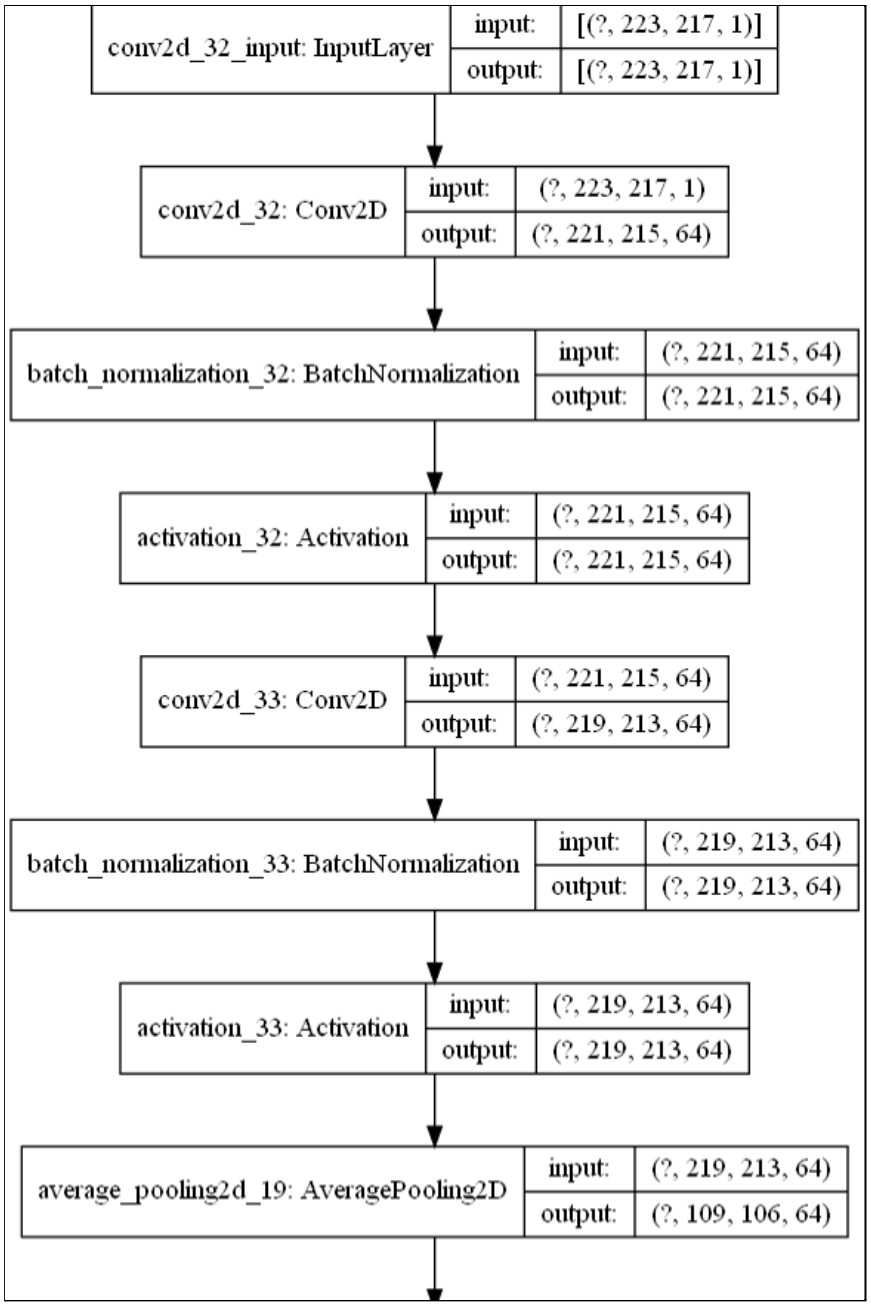
Step 5: Print evaluation score

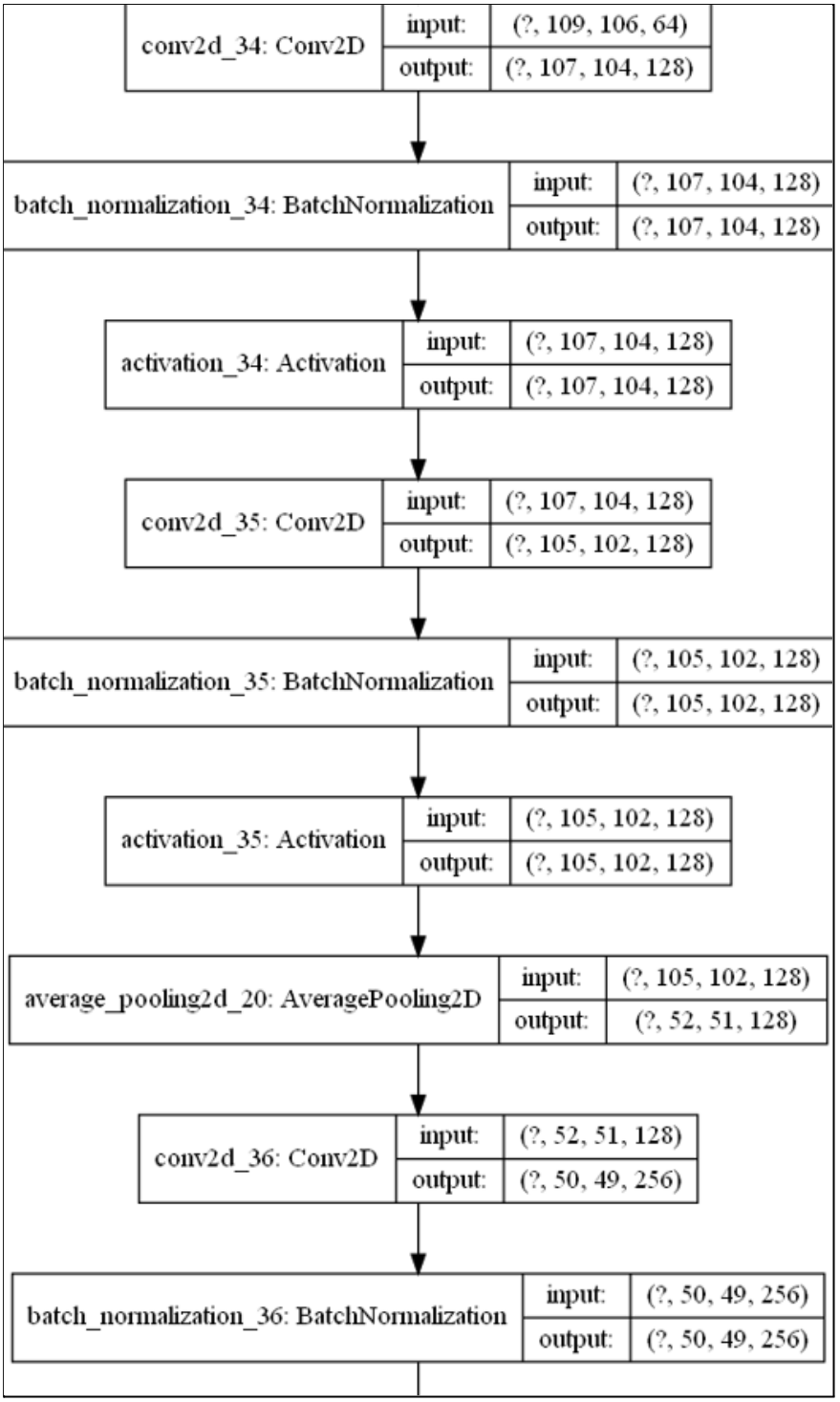
Stop

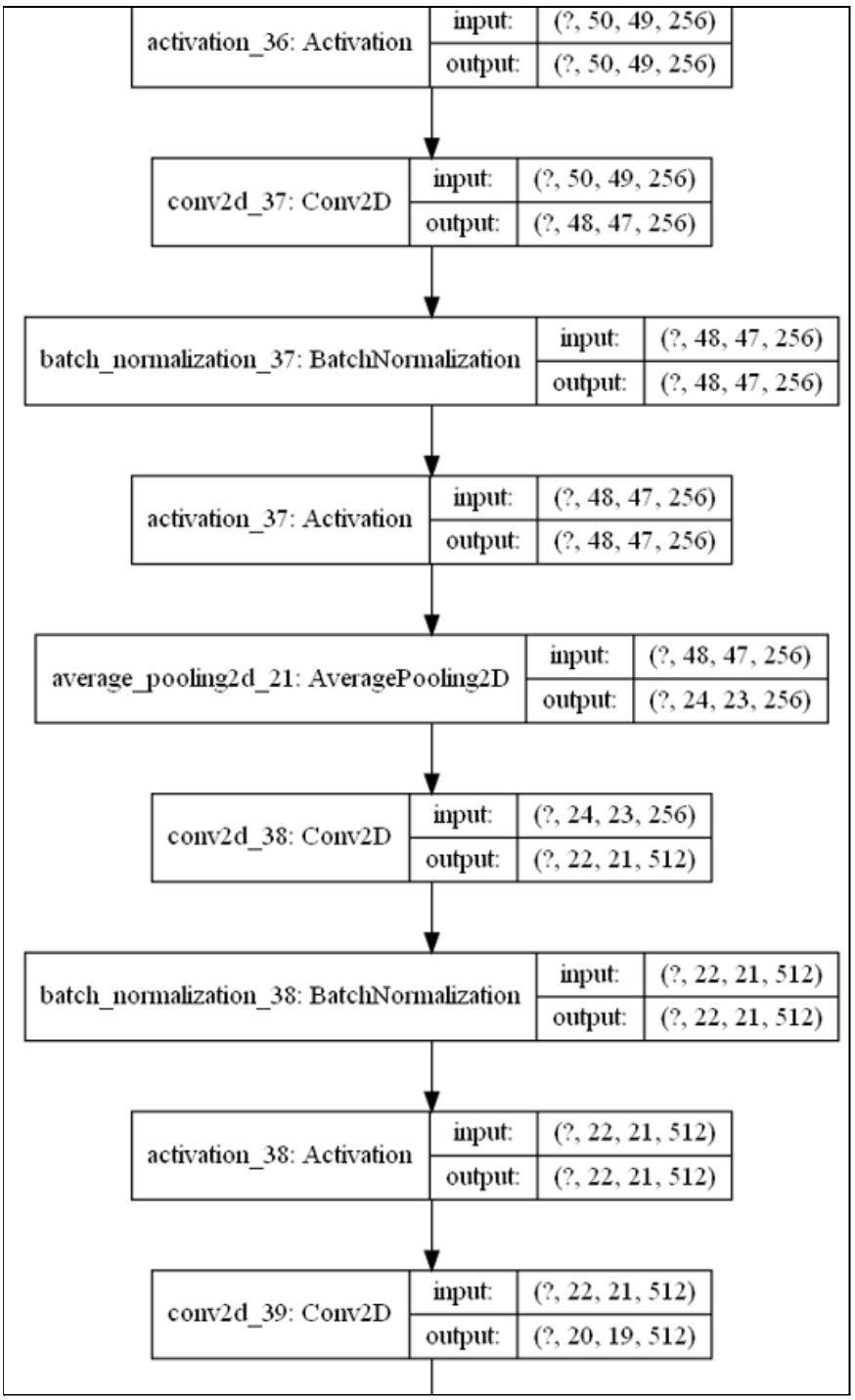
4.3 Model output

The dataset is split into 80:20 ratio for training and testing. To calculate validation accuracy an evaluation dataset from Dcase is used. The training accuracy is 62.62% and the validation accuracy from the 80:20 split is 61.26%. Whereas the validation accuracy from the evaluation dataset is 61.01%.

4.4 Model plot




conv2d_34: Conv2D
input: (?, 109, 106, 64)
output: (?, 107, 104, 128)
batch_normalization_34: BatchNormalization
input: (?, 107, 104, 128)
output: (?, 107, 104, 128)
activation_34: Activation
input: (?, 107, 104, 128)
output: (?, 107, 104, 128)
conv2d_35: Conv2D
input: (?, 107, 104, 128)
output: (?, 105, 102, 128)
batch_normalization_35: BatchNormalization
input: (?, 105, 102, 128)
output: (?, 105, 102, 128)
activation_35: Activation
input: (?, 105, 102, 128)
output: (?, 105, 102, 128)
average_pooling2d_20: AveragePooling2D
input: (?, 105, 102, 128)
output: (?, 52, 51, 128)
conv2d_36: Conv2D
input: (?, 52, 51, 128)
output: (?, 50, 49, 256)
batch_normalization_36: BatchNormalization
input: (?, 50, 49, 256)
output: (?, 50, 49, 256)


activation_36: Activation
input: (?, 50, 49, 256)
output: (?, 50, 49, 256)
conv2d_37: Conv2D
input: (?, 50, 49, 256)
output: (?, 48, 47, 256)
batch_normalization_37: BatchNormalization
input: (?, 48, 47, 256)
output: (?, 48, 47, 256)
activation_37: Activation
input: (?, 48, 47, 256)
output: (?, 48, 47, 256)
average_pooling2d_21: AveragePooling2D
input: (?, 48, 47, 256)
output: (?, 24, 23, 256)
conv2d_38: Conv2D
input: (?, 24, 23, 256)
output: (?, 22, 21, 512)
batch_normalization_38: BatchNormalization
input: (?, 22, 21, 512)
output: (?, 22, 21, 512)
activation_38: Activation
input: (?, 22, 21, 512)
output: (?, 22, 21, 512)
conv2d_39: Conv2D
input: (?, 22, 21, 512)
output: (?, 20, 19, 512)

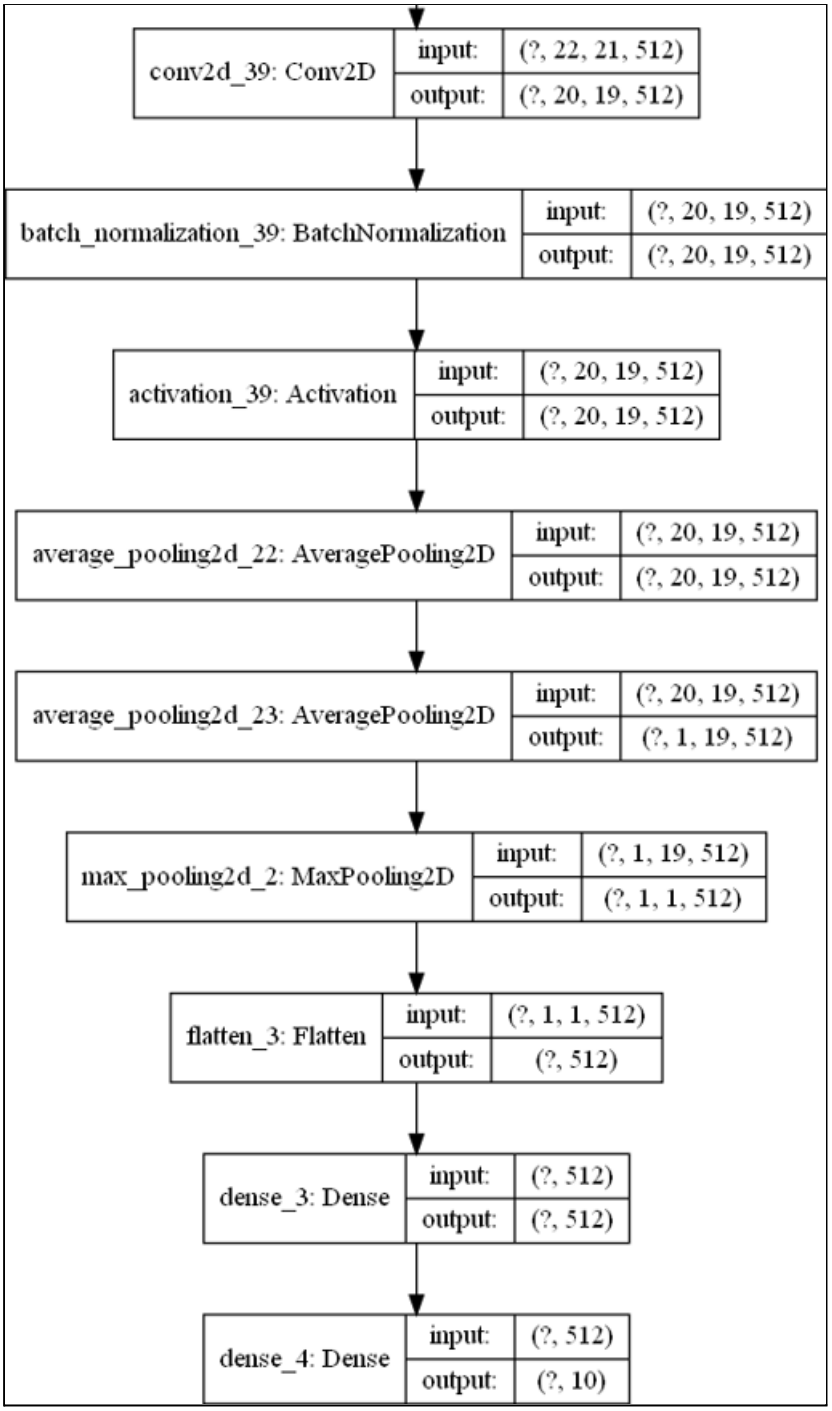


Fig-18 Model plot

4.5 Model summary

Layer (type)	Output Shape	Param #
conv2d_32 (Conv2D)	(None, 221, 215, 64)	640
batch_normalization_32 (Batc	(None, 221, 215, 64)	256
activation_32 (Activation)	(None, 221, 215, 64)	0
conv2d_33 (Conv2D)	(None, 219, 213, 64)	36928
batch_normalization_33 (Batc	(None, 219, 213, 64)	256
activation_33 (Activation)	(None, 219, 213, 64)	0
average_pooling2d_19 (Averag	(None, 109, 106, 64)	0
conv2d_34 (Conv2D)	(None, 107, 104, 128)	73856
batch_normalization_34 (Batc	(None, 107, 104, 128)	512
activation_34 (Activation)	(None, 107, 104, 128)	0
conv2d_35 (Conv2D)	(None, 105, 102, 128)	147584
batch_normalization_35 (Batc	(None, 105, 102, 128)	512
activation_35 (Activation)	(None, 105, 102, 128)	0
average_pooling2d_20 (Averag	(None, 52, 51, 128)	0
conv2d_36 (Conv2D)	(None, 50, 49, 256)	295168
batch_normalization_36 (Batc	(None, 50, 49, 256)	1024
activation_36 (Activation)	(None, 50, 49, 256)	0
conv2d_37 (Conv2D)	(None, 48, 47, 256)	590080
batch_normalization_37 (Batc	(None, 48, 47, 256)	1024
activation_37 (Activation)	(None, 48, 47, 256)	0
average_pooling2d_21 (Averag	(None, 24, 23, 256)	0
conv2d_38 (Conv2D)	(None, 22, 21, 512)	1180160
batch_normalization_38 (Batc	(None, 22, 21, 512)	2048

activation_38 (Activation)	(None, 22, 21, 512)	0
conv2d_39 (Conv2D)	(None, 20, 19, 512)	2359808
batch_normalization_39 (Batc	(None, 20, 19, 512)	2048
activation_39 (Activation)	(None, 20, 19, 512)	0
average_pooling2d_22 (Averag	(None, 20, 19, 512)	0
average_pooling2d_23 (Averag	(None, 1, 19, 512)	0
max_pooling2d_2 (MaxPooling2	(None, 1, 1, 512)	0
flatten_3 (Flatten)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
dense_4 (Dense)	(None, 10)	5130

Total params: 4,959,690

Trainable params: 4,955,850

Non-trainable params: 3,840

Table-1 Model Summary

5. RESULTS & DISCUSSION

```
Epoch 94/100
18432/18432 [==============================] - 960s 160ms/step - loss: 0.3716 - acc: 0.6128
- val_loss: 0.1002 - val_acc: 0.6005
Epoch 95/100
18432/18432 [==============================] - 960s 162ms/step - loss: 0.3458 - acc: 0.6144
- val_loss: 0.2858 - val_acc: 0.6050
Epoch 96/100
18432/18432 [==============================] - 1020s 163ms/step - loss: 0.3242 - acc: 0.6161
- val_loss: 0.2804 - val_acc: 0.6062
Epoch 97/100
18432/18432 [==============================] - 960s 158ms/step - loss: 0.3076 - acc: 0.6187
- val_loss: 0.2733 - val_acc: 0.6080
Epoch 98/100
18432/18432 [==============================] - 1020s 164ms/step - loss: 0.3981 - acc: 0.6198
- val_loss: 0.2705 - val_acc: 0.6081
Epoch 99/100
18432/18432 [==============================] - 960s 161ms/step - loss: 0.3886 - acc: 0.6223
- val_loss: 0.2647 - val_acc: 0.6117
Epoch 100/100
18432/18432 [==============================] - 1020s 163ms/step - loss: 0.3713 - acc: 0.6262
- val_loss: 0.2594 - val_acc: 0.6126
Test loss: 0.2594382493108958
Test accuracy: 0.6100646677113191
```

Fig-19 Result

The model achieved an accuracy of 61% after running for 100 epochs with a validation loss of 0.2594.

The accuracy achieved is 7.2% higher compared to the baseline model(53.8%).

Model Name	Features Used	Accuracy
Modified CNN-9	Log-mel energies	61%
DCASE2021 Task 1 Baseline, Subtask A	Log-mel energies	47.7 %
DCASE2020 Task 1 Baseline, Subtask A	Log-mel energies	54.1%
[7]Trident ResNet	Log-mel energies, deltas, delta-deltas	73.7%

6. SUMMARY AND CONCLUSION

6.1 Summary of achievements

In this project a lot of spectrograms were generated and a lot of time was spent on observing the generated spectrograms. The dataset contains a number of classes and when comparing two different classes, there were some acoustic events that were common for both classes, like for example let us consider two classes airport and metro-station. If we carefully listen to both the audios then we can hear similar audio events which take place both in the airport and metro-station such as announcement, people chatting and shattering of glass (a glass could shatter anywhere).

All of these, similar acoustic event, across all the classes generates similar patterns of mel-spectrogram which are again common across all the other spectrograms which has that acoustic event, this creates ambiguity, and ambiguity can cause a mel-spectrogram to be miss classified by the CNN which leads to lower accuracy.

A pre-trained model could further help a little to increase the accuracy but it would also suffer from ambiguity, one way to overcome this would be to train the model not just with audio but also with video input or pictures taken in certain intervals, this could help with certain ambiguity.

6.2 Limitations of the Project

Ambiguous Allocation between Sound Events and Scenes causing mis-classification. Acoustic scenes often comprise multiple sound events, which are not class-specific, but instead appear in a similar way in various scene classes.

Real-World Deployment: large variability in devices can cause lower precision while classifying an acoustic scene. This often requires a model compression step, where trained classification models are reduced in size and redundant components need to be identified.

6.3 Future Scope of Work

Data augmentation and use of pre-trained models can improve the classification accuracy.
The model can be run for a few more hundred epochs and additional parameter tuning can be done for further improving the accuracy of the model as well as decreasing the complexity of the model.

7. GANTT CHART

Activity	Time Frame				
	Feb 27-28	March 01-31	April 01-30	May 01-31	June 01-30
LITERATURE SURVEY					
PROBLEM IDENTIFICATION					
DESIGN					
IMPLEMENTATION					
TESTING					
DOCUMENTATION					

	Proposed Activity
	Ongoing Activity
	Activity Achieved

8. REFERENCES

[1] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015, July). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python in Science Conference* (Vol. 8, pp. 18-25).

[2] Chu, S., Narayanan, S., Kuo, C.-C. J., & Mataric, M. J. (2006). Where am I? Scene recognition for mobile robots using audio features. In *Proceedings IEEE International Conference on Multimedia and Expo*, pp. 885–888.

[3] Mesaros, A., Heittola, T., & Virtanen, T. (2018, November). A multi-device dataset for urban acoustic scene classification. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 9–13.

[4] IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events. In *Detection and Classification of Acoustic Scenes and Events 2021*. https://dcase.community/challenge2021/

[5] Geiger, J. T., Schuller B., & Rigoll, G. (2013). Large-scale audio feature extraction and SVM for acoustic scene classification. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.*
https://doi.org/10.1109/WASPAA.2013.6701857

[6] Kong, Q., Cao Y., Iqbal, T., Xu, Y., Wang, W., & Plumbley. M. (2019, April). Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems. In *Detection and Classification of Acoustic Scenes and Events 2019*.

[7] Phinyomark. A., Phukpattaranont. P., Limsakul. C. (2012, June). Feature reduction and selection for EMG signal classification. In *Expert Systems with Applications*. (Vol. 39, pp. 7420-7431)