

Sound Classification Using Deep Learning for Hard of Hearing and Deaf People

by

Md.Adnan Habib

18101551

Zarif Raiyan Arefeen

18101214

Arafat Hussain

18101093

S.M.Rownak Shahriyer

18101611

Tanzid Islam

18101673

A thesis paper made for the partial fulfillment of completing the Undergraduate of BRAC University for the department of Computer Science and Engineering(CSE)

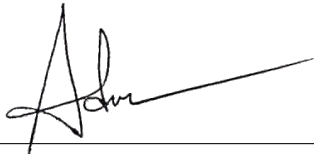
Department of Computer Science and Engineering
Brac University
January 2022

© 2022. Brac University
All rights reserved

Declaration

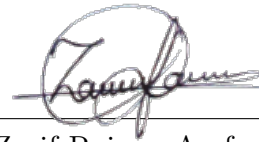
We, hereby, are glad to state that, the paper “Sound Classification Using Deep Learning for Hard of Hearing and Deaf People”, has been initiated while pursuing the thesis of under graduation under BRAC University, and the paper presented is a piece of our own original research. We can assure that our paper does not contain any information that has been presented or accepted by any other academic or other institution, except for where stated by reference. All sources that have been helpful to us, while completing the paper are properly cited and recognized.

Student’s Full Name & Signature:



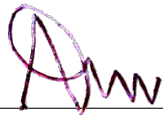
Md. Adnan Habib

18101551



Zarif Raiyan Arefeen

18101214



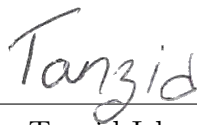
Arafat Hussain

18101193



S.M. Rownak Shahriyer

18101611



Tanzid Islam

18101673

Approval

The thesis paper entitled ‘Sound Classification Using Deep Learning for Hard of Hearing and Deaf People’ has been submitted by:

1. Md.Adnan Habib (18101551)
2. Zarif Raiyan Arefeen (18101214)
3. Arafat Hussain (18101093)
4. S.M.Rownak Shahriyer (18101611)
5. Tanzid Islam (18101673)

This is to clarify that our thesis paper has met the standard provided by our University and our originality is maintained, while pursuing to complete our Bachelor of Science in Computer Science and Engineering(CSE).

Examining Committee:

Supervisor:
(Member)

Dr. Mohammad Zavid Parvez
Assistant Professor
Department of Computer Science and Engineering
BRAC University



Co Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

Our study is for the betterment of the hearing impaired. The data and study provided can be fully relied upon. We did not conduct any illegal action while pursuing our research. The data-set, which was created, will be provided online for further research purposes. Optimistically, we expect our findings to be of a use in the future.

Abstract

Our paper mainly focuses on developing an audio classification for people, who cannot hear properly, using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). One of the many prevalent complaints from hearing aid users is excessive background noise. Hearing aids with background noise classification algorithms can modify the response based on the noisy environment. Speech, azan, and ambient noises are all examples of significant audio signals. Whenever a human hears a sound, they can easily identify the sound, however it's not the same for computers, and we have to feed the algorithm data-sets in order to make it distinguish between different sounds[1]. Hence, we came up with the idea to build a system for people who have problems to hear. We have successfully managed to achieve a total of 98.67%, and 97.01% accuracy after training the data on our CNN and RNN model and testing it respectively.

Keywords: RNN, CNN, melspectrogram, Audio feature extraction

Acknowledgement

First and most importantly, we would like to thank ALLAH, the Almighty, as he kept us safe in this Covid epidemic, even after some of us felt sick but by the blessing of the Almighty, we were not affected by the deadly virus. Due to the presence of the internet we were able to communicate with each other effectively and easily even after we were maintaining strict lockdown. Throughout the writing, we have received an enormous amount of support and help from our respected thesis faculty, Dr. Mohammad Zavid Parvez, sir provided us with help whenever we needed. Along with Dr. Mohammad Zavid Parvez Sir, our thesis co-supervisor, Rafeed Rahman Sir, was extremely helpful, without them it would have been impossible to complete our paper. Furthermore, our parents deserve a lot of credit for providing us with the constant support and care we needed. Thanks to their continual buttress, as we were comfortably able to complete our paper without any major hindrance, during our whole course of under graduation. In the end, we would like to portray our gratitude towards our university, BRAC University, which provided us with a platform and connection which not only allowed us to complete our thesis paper, but also helped us ameliorate throughout the course of four amazing and wholesome years.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iv
Abstract	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	xi
Nomenclature	xii
1 Introduction	1
1.1 Problem Statement	2
1.2 Research Objectives	2
1.3 Thesis Orientation	3
2 Literature Review	4
3 Background Analysis	9
3.1 CNN Algorithm	9
3.1.1 Conv 2D	12
3.2 RNN Algorithm	15
3.2.1 LSTM	18
4 Model Implementation and Optimization	21
4.1 Data-Set	21
4.2 Data Preprocessing	24
4.3 Features of Sound	24
4.4 Feature Extraction	24
4.5 Feature Selection and Engineering	26
4.5.1 Feature selection	26
4.5.2 Feature Engineering	26
4.6 Train-Test split	26

4.7	Models	27
4.7.1	CNN	27
4.7.2	RNN	28
5	Results and Discussion	29
	Conclusion	34
	Bibliography	37

List of Figures

3.1	CNN network	9
3.2	Convoluting features from input data	10
3.3	Multi layered features extracted from input data	10
3.4	Different pooling methods	11
3.5	Work flow of CNN	11
3.6	Conv 2D network with 2D input and 2D output	12
3.7	Applying Padding to the input	13
3.8	Applying Stride to the input	13
3.9	Collection of kernels	13
3.10	Kernel slides over respective input to produce processed version of input	14
3.11	Adding all the processed version of input to generate one output	14
3.12	Adding bias to the output	15
3.13	RNN network	15
3.14	RNN network	15
3.15	A complex RNN network	16
3.16	Backpropagation in RNN	16
3.17	Different types of RNN	17
3.18	Different Activation functions	17
3.19	LSTM network	18
3.20	LSTM classification	18
3.21	LSTM classification	19
3.22	Sigmoid graph (Red) and Tanh graph (Green)	19
3.23	Work flow of LSTM	20
4.1	Waveplot for acoustic guitar	23
4.2	Waveplot for Camera	23
4.3	Waveplot for chainsaw	23
4.4	Waveplot for dogbark	23
4.5	Waveplot for drums	23
4.6	Waveplot for goat	23
4.7	Waveplot for train	23
4.8	Melpectrogram of Air plane	25
4.9	Melpectrogram of Car horn	25
4.10	Melpectrogram of Chainsaw	25
4.11	Melpectrogram of Crow	25
4.12	Melpectrogram of Glass break	25
4.13	Melpectrogram of Dog	25

4.14	Input vs Output graph of ReLu	27
5.1	Accuracy comparison of RNN vs CNN	30
5.2	Confusion matrix of CNN	32
5.3	Confusion matrix of RNN	33

List of Tables

5.1	Classification Report for CNN and RNN	31
-----	---	----

Nomenclature

ANN Artificial Neural Network

CI Cochlear Implant

CNN Convolutional Neural Network

DCNNs Deep Convolutional Neural Networks

DNN Deep Neural Network

DTW Dynamic Time Warping

HMM Hidden Markov Model

MFCC Mel-frequency cepstral coefficients

RBM Results-Based Management

ReLU Rectified linear activation function

RNN Recurrent Neural Network

SE Simulated Evolution

SNR Signal to Noise Ratio

Chapter 1

Introduction

Ear, which is one of the most important sensing organs of our body. In our childhood we were taught that the ear is among the 5 most important organ in the human body, the others included are-tongue, skin, eyes and nose, our tongue helps us to taste, our eye lets us see the world, our skin makes us feel the temperature changes in our surrounding and help our body take action accordingly and our nose aids us to smell the environment around us.[2] Finally we have our ear, which doesn't only provide us with hearing but also it provides balance in our body.

Information from the inner sensory organs, from what we see and sensory receptors all through our body are extremely essential in-order to keep us balanced. The cerebellum and cerebral cortex of the brain process information that helps the body to cope with variations in head speed and direction.[3] Hence we can understand the importance of the ear in our body. With the absence of even one of the most valuable sensing organs, our body will not be performing as normal as any other normal human being.

Government has taken several initiatives towards this problem, however, there is still no legal protection for the language, and no measures have been taken to institutionalize it. As a result, sign language users do not have complete access to essential information and services, such as education, health care, and job prospects. As a result, this community strives to protect the rights of sign language users.[4]

Hearing impaired face a lot of difficulties dealing with everyday life. People, who can hear properly, can understand if somebody calls him from behind, but that's not the case for the hearing impaired. Whenever someone needs them they have to touch in order to make them understand that we are calling them, same goes if any car keeps honking behind the hearing impaired, it's almost impossible for some of them to hear barely anything. Research has shown that people, with hearing problems, tend to feel depressed or deserted due to their lack of communication. And, furthermore, the sign language used by people varies from country to country, even after using sign language the indication of what we are trying to say may not be clear every time.

People across the world use hearing aids in order to communicate, but this solution may not be viable, since the ear hearing aid may get clogged with the ear wax. For some people, it is really tiring to wear the hearing aid for a long time. People have to carry extra/backup batteries to use if the batteries being used get damaged or the charge finishes, due to high usage. Additionally, for a country like us, it is really tough for some people to buy a hearing aid and maintain it properly. Therefore, in this paper, we have decided to implement deep learning with our own created data-set, to create a system/software which can be used by people who suffer from hearing disability.

1.1 Problem Statement

Bangladesh is a country of approximately 163 million people, according to a study in 2004, done by M. Alauddin and A. Hasnat Joarder, stated that around 13 million people were suffering from different kinds of hearing loss at that time, out of which 3 million people suffered from severe hearing disabilities[5]. Hence came the importance of focusing on the topic of developing a certain kind of hearing system so that our disabled population can be utilized in order to build a better society. We should provide more awareness to the fact that the amount of hearing disabled humans are increasing day by day. One of the solutions to dealing with the hearing impaired is to provide them with hearing aids, but even after that a lot of people complain about the amplification of the sound that is passed in the hearing aid, the amplification of all the sounds makes it really hard for them to understand anything[6]. In our country, an action taken now will greatly help our future generation, with hearing problems, and help them provide to society as much as any other human being.

1.2 Research Objectives

Hearing-impaired people confront a number of difficulties, these include- Misunderstandings in sign language, difficulties of being in public, being depressed or having anxiety and many more[7]. They navigate the environment in a very different way than people who have excellent hearing. According to the World Health Organization the amount of people with some level of hearing problem will increase to 700 million people by 2050[8].

Our main aim of this research is to provide an audio detection system and will be able to assist people with hearing disability and provide them with enough information with the information of the sounds that surround them. Our research focuses on different types of sound properly identified by our system, so that the best output is given to our focused people.

1.3 Thesis Orientation

We have divided our thesis in 5 chapters, the description of the chapter are as follows:

In chapter 1, we have talked about the introduction to the Hearing Impaired people, the daily problems they face while dealing with daily life, we have talked about our problem statement and our Research objectives.

In chapter 2, we included all the related works we have found helpful.

In chapter 3, we have given a detailed description of the models we used to build our project.

In chapter 4, gives an elaboration of the data-set we used, we talked about the data preprocessing and the feature extraction of the data we collected.

In chapter 5, the classifier's output and our own observations of various classifiers and situations are contrasted in various tables and visualizations.

Chapter 2

Literature Review

The paper that was released by K.Karthikeyan and Dr.R.Mala[1], in 2018 stated that they have developed a system that can differentiate between audio clips. The system was able to differentiate between Sports, News, and Musical audio clips. The primary focus of the study was to build a system that can identify the mood of any music. Music is one of the most common and strong ways of expressing emotions. In music education and music psychology, emotion-based components or any subjective experiences or any physiological or behavioral responses have been recognized to have a strong association with music expressiveness. However, it is not easy to evaluate music mood based on musical audio as it is highly subjective but there is a strong connection between our mood and the music we listen to. The authors tried to study the correlation and they were able to create a system with which was able to connect this component into a point in their experiment. Although they have used a fairly small data-set and generic audio clips which are close to each other, with the system they have built, the result they found while experimenting was encouraging. The authors have extracted the audio into multiple features for this experiment. The audio clips were divided into features like Energy Features, Frequency Domain Features, Pitch Based Features, Time Domain Features and MFCC, and others. They used a multilayer feed-forward network with a supervised backpropagation learning technique to create an audio classifier. The audio classifier was implemented using ANN and MATLAB in two phases. The first phase was classified into two classes such as Sports Music and News and the later phase was classified into moods such as angry, sad and happy moods. The extracted audio of three different genres is normalized and divided into training and testing data. After the preparation of data sets, the data was fed into the network to create the multi-layer feed-forward network. Even though the training set was pretty small and consisted of only audio clips of different semantic structures, the result was something that can be looked forward to as the accuracy of the system was up to 80% and certainly had room for more improvement.

C. Freeman, R.D. Dony, and S.M. Areibi (2007)[9], stated in their paper that they have developed a background noise detection system where the system can successfully differentiate between different background noises. The problem the authors were trying to solve was the problem with the detection of background noise using

Hearing Aids. Hearing Aids generally amplify the background noise with all the other noises, making it difficult for the patients to listen to the background sounds clearly. They extended the work of Buchler et. al,[10] by specifically addressing the problem related to the detection of the background sound and classification. In their experiment, they have applied three different models, ANN, Windowed ANN, and HMM. Their goal was to put these three models to test and to find out which model would be a good choice in detecting background noises to reliable accuracy. The data consisted of different background sounds recorded from the environments which were collected from the Freesound database. The models were divided into four different classes of background noises and those were, speech babble, traffic noise, typing, and white noise. The authors have extracted the audio into multiple features for this experiment. The audio clips were divided into three different features, mean frequency, high and low frequency, and envelope modulation. Then the authors tested this on three different models as mentioned before, like ANN, Windowed ANN, and HMM, and came back with some promising results. According to the experiment Windowed ANN was found to be the most reliable model for background sound classification compared to the other two models which are ANN and HMM. ANN testing set had an average accuracy of 78.6% and the best reaching 89.1%, HMM averaging at 92.3% with the best run with 92.3% accuracy, and the Windowed ANN having the best accuracy out of all the model averaging at 94.2% and the best reaching 97.9%. In addition, Windowed ANN being the most accurate out of the models, windowed ANN is smaller and less time-consuming when it comes to training. The difference in size plays a significant part since the experiment was done for a hearing aid, which has limited space and computing capacity. Overall, utilizing an ANN with a windowed input for background categorization in a hearing air looks to be a good decision.

S. Hershey, S. Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, Rif A. Saurous, B. Seybold, M. Slaney, Ron J. Weiss, and Kevin Wilson[11], came up with a very different approach towards the audio classification than other researchers. Convolutional Neural Network (CNN) was proven to be very effective when it came to image classification. Some CNN models are ubiquitous in the image data space as they work very well on computer vision tasks, such as image classifications or object detections. Since Image classifications have improved greatly using Convolutional Neural Networks, the authors were curious to see how these Convolutional Neural Network models work for audio classification. The primary goal of this experiment was to investigate how popular Deep Neural Network architectures perform for audio classifications compared to image classifications and prove that state-of-art image networks are better at audio classification than simple fully connected networks.[12] The authors used the data-set YouTube-100M, a comparatively huge data-set of around 100M YouTube videos, which includes 70M training videos, 10M evaluation videos, and a pool of 20M videos that were used for validation. The training set contained 70M videos which are roughly 5.25 million hours and 30,871 machine-generated labels. The baseline model of this experiment is fully connected Deep Neural Network(DNN) which was compared with several other Neural Network models on successful image classifiers such as, AlexNet, VGG, Inception, and ResNet. The results from this experiment were very impressive as the state-of-art image networks were successfully

able to detect and classify audio better than the simple fully connected network or any earlier image-classification architectures. The authors were able to better accuracy by changing the size of both the training set and label vocabulary. From the graph and tables that were provided in the paper, it was seen that increasing the training data-set with larger label set vocabularies improved performance to a good extent[11]. Their system was successful in identifying the audios that were being played in the background, while they tested their system on segments of a video. Moreover, the system was also able to identify different sounds being played at the same time with great accuracy.

In their work, Hugo Meindo and Joao Neto (2005)[13], built a system for a low latency stream-based audio pre-processing system for the News that was broadcast on TV using a model-based technique. They were successfully able to build a system that was able to classify different types of audio from News videos. Their primary goal was to categorize audio into three different parts, speech/non-speech classification where the system tries to detect speech in the audio, another classification is gender classification where the system tries to detect the gender of the News host and the last classification is background sound detection where the system tries to differentiate background sound from speech. The authors used two different databases. One database was to train complex models and for speech recognition and the other database was for evaluation of speech segmentation and classification. The Portuguese-BN corpus[14] data-set was collected from a Portuguese public broadcast company and the data-set mostly contained all types of News programs and broadcasts. It contained 46 hours of total time and this data-set was used for training, development, and evaluation. Another data-set COST278-BN corpus[15] consisted of 30 hours of recorded news and this data-set was used for audio segmentation. The system that was built was composed of 5 modules, three for classification, one for speaker cluster, and one for acoustic change detections. The five modules were incorporated into a single model-based algorithm and to increase the model accuracy, the algorithm makes use of the Artificial Neural Network. All the Artificial Neural Networks that were used in this algorithm are feed-forward type fully connected Multi-Layer Perceptrons and a backpropagation algorithm was used to train the model. From the experiment, they were able to conclude that the system showed very good performance and was also able to keep very low latency for any stream-based operation. The recognition test that was carried out showed a very small distortion in performance when compared with hand-labeled audio segmentation under the same conditions.

The paper presented by Z. Kons and O. Toledo-Ronen (2021)[15], enlightened us with the insight of using Deep Neural Networks on Audio event classification. Usually, Deep Neural Network was not always the first choice when it came to audio classification but the authors in this experiment were successful in extracting impressive results with Deep Neural Network in Audio Classification. In this experiment, the authors at first built a system with Deep Neural Network classifiers and later compared the classifier results with SVM (support vector machine) classifiers and GMM (Gaussian Mixture Model) classifiers. The authors collected data from the FreeSound.org site to build their data-set. The data-sets mostly considered a wide range of acoustic event sounds and outdoor recordings of different kinds of live events. The samples were annotated manually including locating and labeling

segments with relevant audio classes. The Deep Neural Network classifier consists of a multilayer feed-forward perceptron network. Furthermore, they have suggested a new technique for enhancing the network's pre-training process by incorporating additional scaling factors during the RBM training reconstruction stage. With very minimal deterioration in classification performance, this approach appears to be beneficial for lowering the pre-training error rate and assisting back-propagation to converge faster. From the experiment, the conclusion was drawn that, for audio event classification Deep Neural Network (DNN) can be a very useful tool and also it performs slightly better than the SVM model. The authors also recommended, combining both the DNN and SVM model can yield overall better results and will improve the performance to a good extent.

To classify gunshot audio sounds, Settha Tangkawanit and Surachet Kanprachar (2018)[16], came up with a slightly different approach, they employed an ANN with some minor modifications such as the Artificial Neural Network model takes input with promising recognition accuracy. The data-set was made up of 6 different types of guns and from each gun, 10 gunshots were recorded using a high-quality microphone. From each gunshot, 10 other variants of this sound were produced by mixing some additional noises to make a diverse data-set. The noise injection approach was used to prepare the gunshot data for use with the Artificial Neural Network in the learning phase. In the frequency domain, the effect of having a varied number of feature vectors is also investigated. The average amplitudes of frequency bins are used to generate feature vectors for the ANN. With varying noise levels, that is, with different signal-to-noise ratios, the accuracy in identifying 6 gunshot noises is demonstrated (SNRs)[16]. This trained Artificial Neural Network model is tested with some gunfire sounds with a varying signal-to-noise ratio of the original 60 dB, from down 30dB to -10dB up and it was achieved by varying the noise power of the sounds. From this experiment, it has been discovered that using this suggested Artificial Neural Network model, the accuracy of greater than 92 percent may be attained for any number of bins evaluated for SNR greater than or equal to 5 dB.[17][16]

Khaleelur Rahiman PF, Jayanthi VS and Jayanthi AN (202)[18], released an article on speech enhancement methods using a deep learning approach. Their main goal was to use deep learning to enhance speech detection for hearing-impaired listeners. They have developed an algorithm that can decompose noisy distorted speeches into multiple frames as features using SE (search engine) algorithm and using deep convolutional neural networks to feed those noisy frames into the network to produce a frequency channel estimation. In their paper, the authors mentioned how many users were using conventional cochlear implants, also known as CI devices, and how CI users are getting negatively affected due to the presence of other talkers and environmental sounds that are around the device. To deal with these problems, the authors developed an improved CI device that can help the users detect and differentiate different background noises through combined deep convolutional neural networks (DCNNs) and with an SE algorithm. To test the model, the authors used multiple samples of the consonant of Tamil letters and vowels. At first, they used a clean speech, later in the second part they introduced the speech with some background noises (usually fan and music), again the speech was disrupted in the third part and the whole result was shown in the signal electrodiagram.[19] The re-

sults showed good improvement over the existing CI devices and it also showed SE problems in the CI users can be assessed using DCNNs.

The paper that was released by Mete Yaganoglu[3], in 2021 stated in his paper that he was able to build a system that can recognize speech and alert the deaf person in real-time. He also designed a device that will use a microphone to pick up the sound for the surroundings and process the information in a small device in real-time and give deaf people assistance accordingly. In his work, the authors specifically addressed the issues that deaf people face in their day to day life, so he built a wearable piece of technology that can help all users or any particular deaf user. He designed a wearable device that consists of four components, Raspberry PI, grove sensor, microphone, and vibration motor, and the coding was done in Python and the whole process was done in real-time. The author took 4000 audio blocks of men and women and mixed them with environmental sounds and then for each noise, the cluster was divided into four subgroups. This data-set was tested in both a computer environment and real-time. The DTW approach, which is a way of assessing the similarity between sequences of different durations in a time series analysis, was utilized for speech recognition.[3]The test was subsequently repeated three times in three distinct signal-to-noise situations, with 95% accuracy. with 97% specificity and 76% sensitivity.

Chapter 3

Background Analysis

3.1 CNN Algorithm

The Convolutional Neural Network (CNN) is one kind of Deep Learning algorithm which takes input, assigns priority to various features of the input by adding learnable weights and biases, and can distinguish one from the other.[20] It has some convolutional and pooling layer along with some other layers which have several filters/kernels that performs different operation.

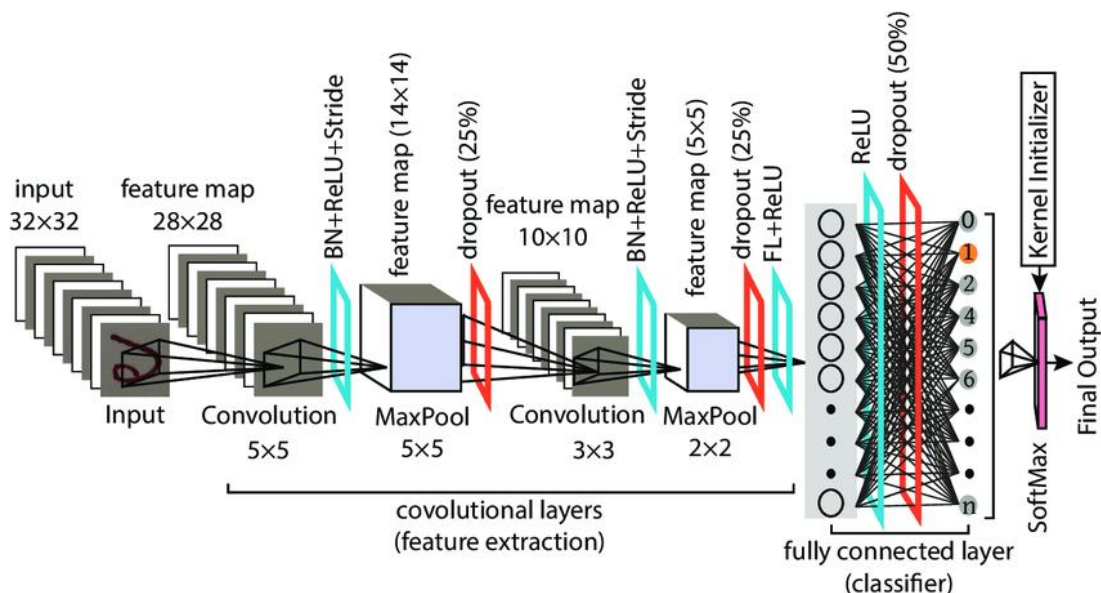


Figure 3.1: CNN network

The convolution layer extracts the features from the input for classification purposes by applying filter/kernel to the input. These convolutional layers are composed of multiple layers where each layer takes the weighted sum of multiple input, generates several activation functions, outputs the activation value, and forward the output to the following layer.

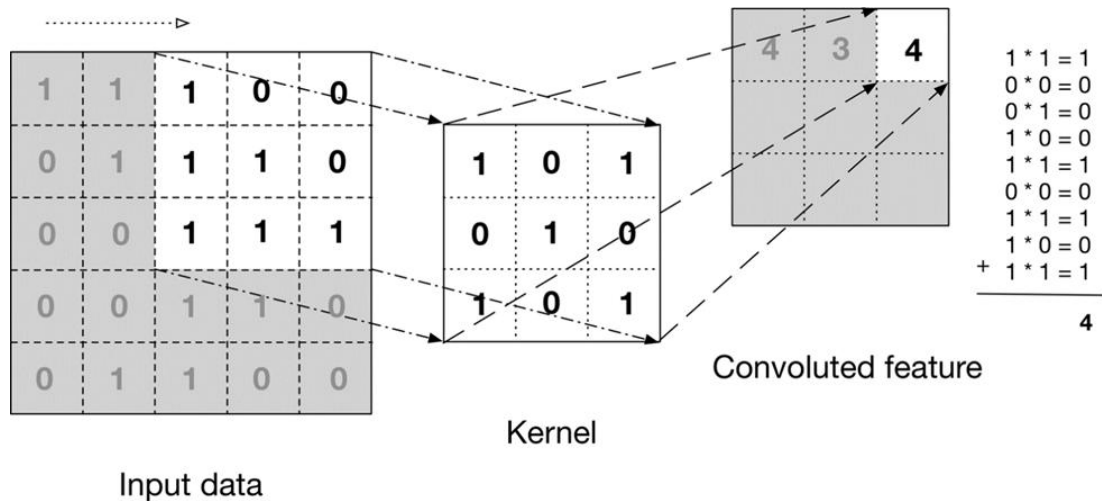


Figure 3.2: Convoluting features from input data

The initial layer extracts basic features and passes the output to the next layer which detects more complex features and the complexity increases along with the number of increasing layers.

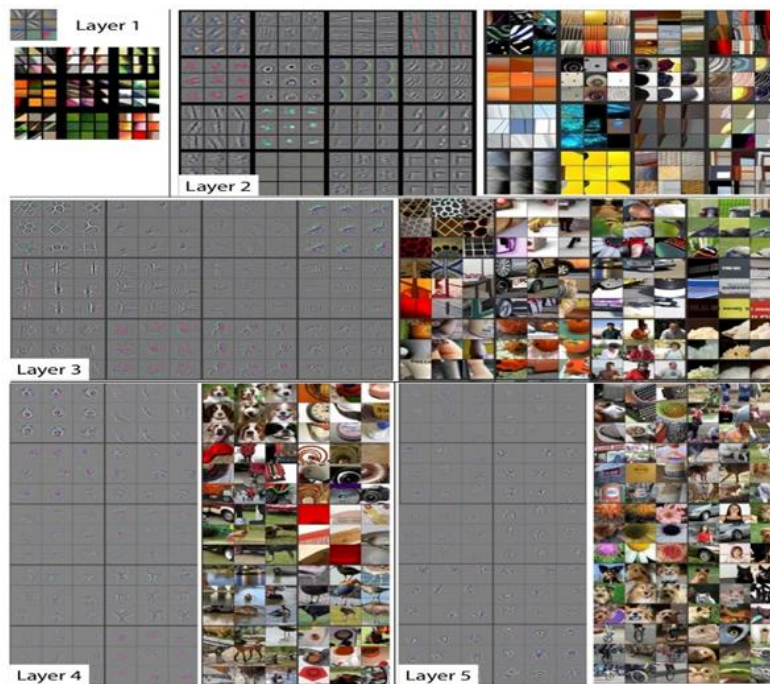


Figure 3.3: Multi layered features extracted from input data

After the convolution layer, the result is sent to the pooling layer which is used to decrease the computational power for data processing by reducing dimensions of the inputs with the help of filters/kernels. Average pooling and max pooling are the two types of pooling methods wherein max pool, the maximum value is chosen from a portion of the input that is covered by the dimension of the kernel.

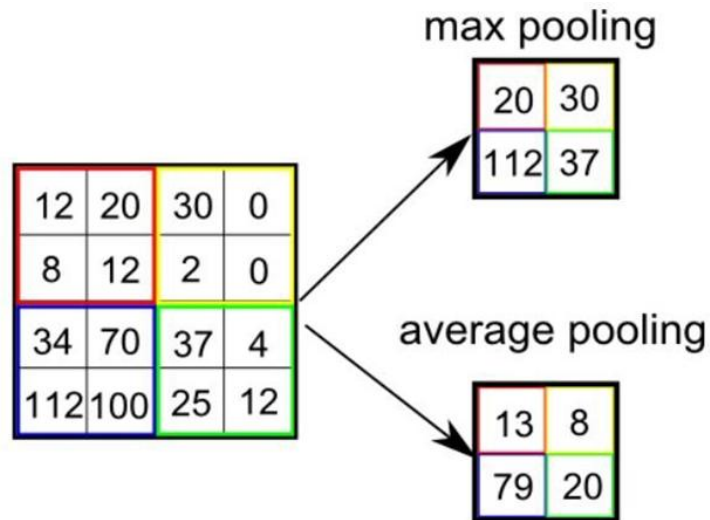


Figure 3.4: Different pooling methods

Max pooling can be described as noise limiter because it can discard the noisy activations, perform de-noising along with decreasing the dimension of the input.[21] Average pool returns the mean value of all the values from a specific subset of the input that is covered by filter's dimension. For noise reduction, average pooling can only decrease the dimension of the input which makes Max pooling a better pooling method in comparison with Average pooling.

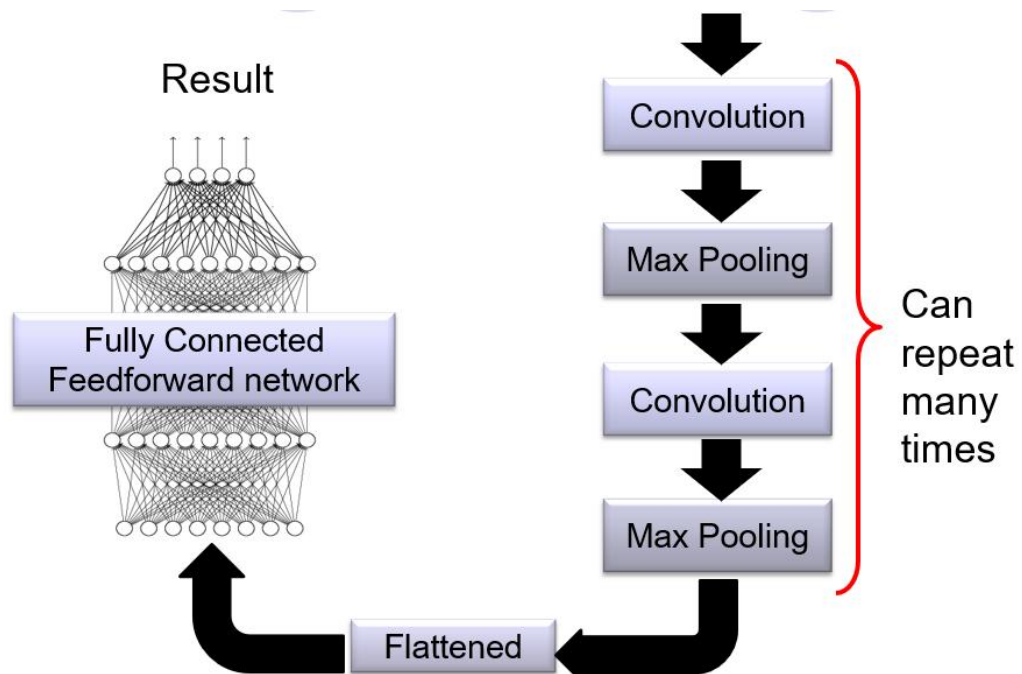


Figure 3.5: Work flow of CNN

The convolution and pooling layer can take place multiple times to flatten out the input which means after processing, the input only holds the necessary information that is required for classification purposes and is being passed through a fully connected feedforward network which classifies the input and generates output accordingly.

CNN can discover essential properties without the need for human intervention.[22] Also, unlike any other classification algorithms, CNN requires very little amount of preprocessing. Along with image detection, CNN can also be applicable in many other sectors like speech recognition, text, and sound classification.

3.1.1 Conv 2D

1D convolution, or just convolution, is a convolution technique that involves a one-dimensional signal. 2D convolution is the process of performing convolution between two signals that spans along with mutual perpendicular dimension.[23] Convolution 2D is a convolution process where a multidimensional (2D) convolution process occurs. It is accomplished by the multiplication and accumulation of the momentary values of overlapping samples that corresponds to the 2D inputs. Here, one of the inputs is inverted for two times, and then 2D convolution kernels are applied to forecast the segmentation map for a specific region/subset of the input. Convolution 2D is one of the most common types of convolution layer.

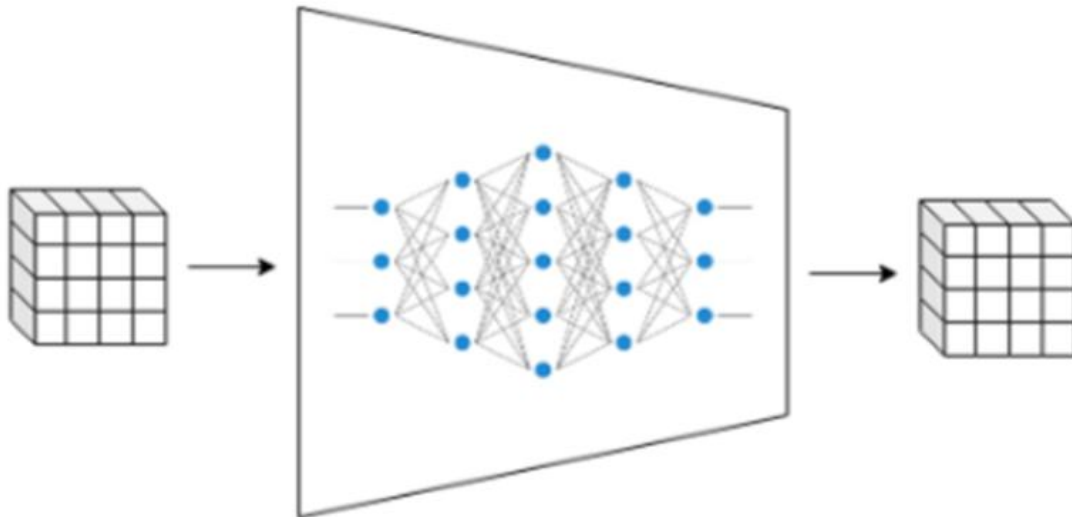


Figure 3.6: Conv 2D network with 2D input and 2D output

The kernel can be different based on the filter that is being applied to the input in the convolutional layer.[24] The kernel slides in 2 directions to extract features from the input. The output features the weighted sums of the features that are extracted from the inputs. The size of the kernel determines the number of input features that get combined together to produce a new output feature.[25]

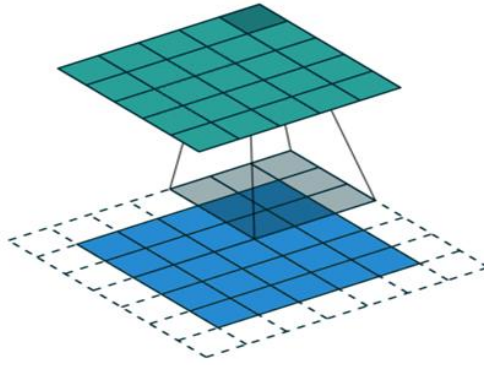


Figure 3.7: Applying Padding to the input

Techniques like padding and stride are applied here, where padding adds extra values to the inputs to allow the kernel to gather the original edge values to be in the center while the padded values are extended beyond the edge to generate an output with the same dimensions as the input.

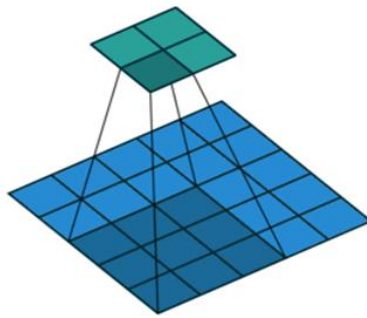


Figure 3.8: Applying Stride to the input

Striding shrinks outputs size smaller than the input. Here, some of the slide portions of the kernel are skipped so that every single slide acts as a standard convolution process which will downsize the output.

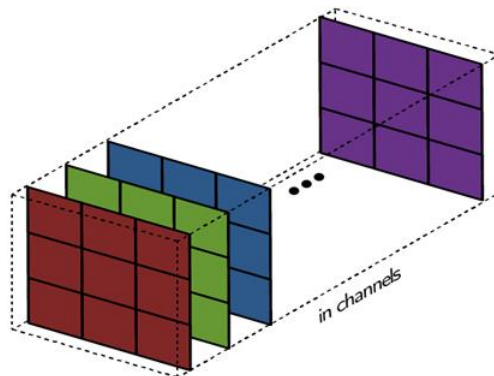


Figure 3.9: Collection of kernels

Also, for multi channel inputs, filters with a collection of kernels are applied where there is one kernel for every single input channel to the layer and each kernel is unique from one another.

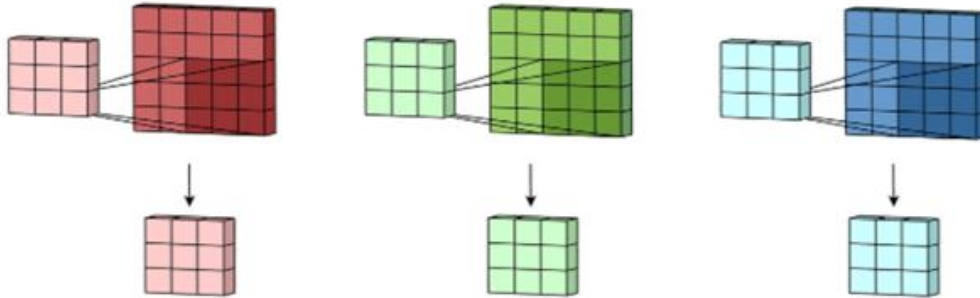


Figure 3.10: Kernel slides over respective input to produce processed version of input

Each of these filters of the kernels slides over their respective channel to produce a processed version of each input and later each of the versions is combined together to form one single overall output channel.

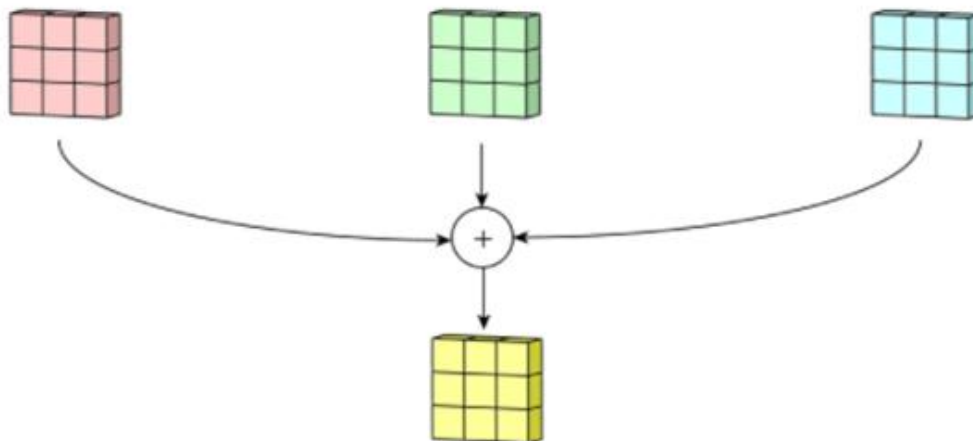


Figure 3.11: Adding all the processed version of input to generate one output

As Inputs are processed by each filter with a different set of kernels and each output filter has one bias, the bias is added to the output channel and then merged to produce the final output channel.

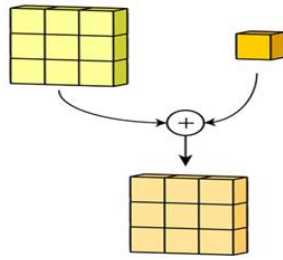


Figure 3.12: Adding bias to the output

Convolution 2D can be used in digital image classification, image processing, object detection, and many other processes like video and audio classification.

3.2 RNN Algorithm

A recurrent neural network (RNN) is one kind of a neural network which works best with time series data or sequential data.[26] Here, the former phase's output is forwarded as input to the following step. As a result, the current output is based on the previous input which allows the neurons to understand and predict the sequence of data.

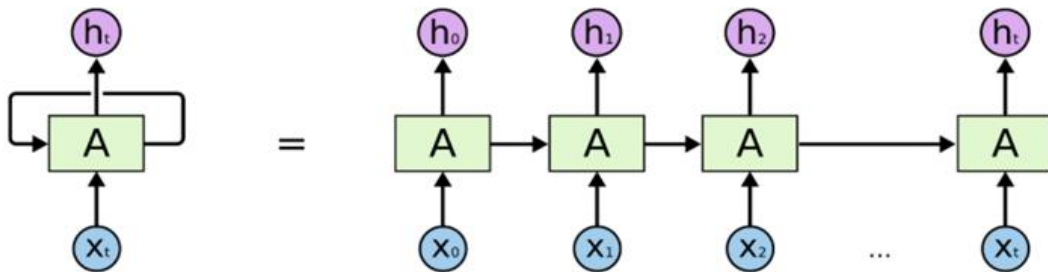


Figure 3.13: RNN network

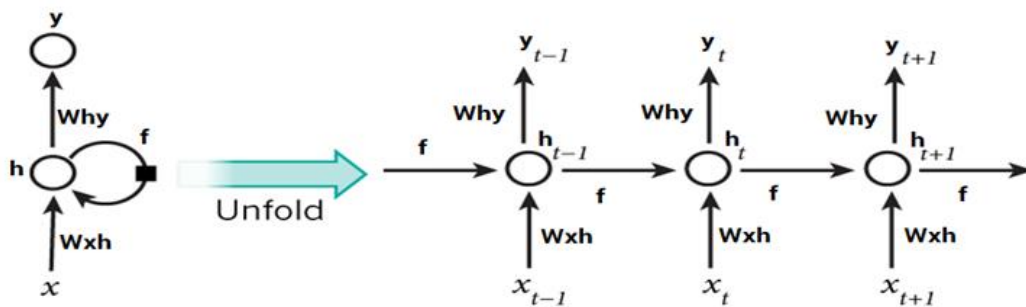


Figure 3.14: RNN network

An RNN has input and output layers along with many hidden layers based on its architecture. Each layer has its activation functions, weights, bias, and these layers take inputs, process them and pass them to the next layer.[27] It also has current and previous state inputs. The previous state input contains crucial information about the current state that assists the network to memories and generate the sequential data output.

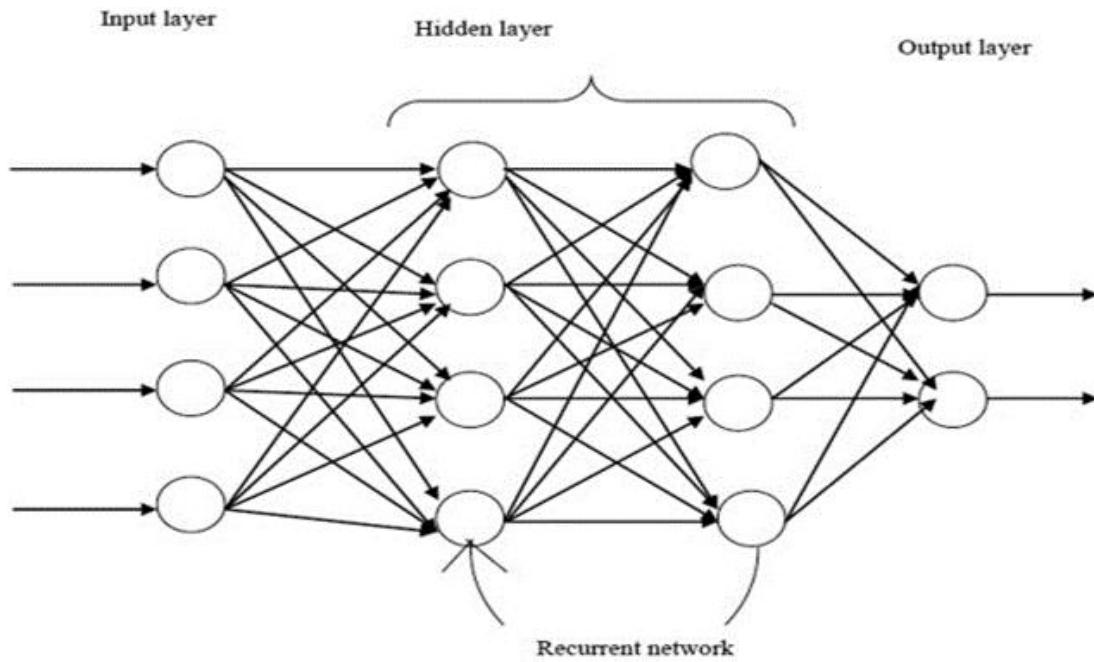


Figure 3.15: A complex RNN network

Like any other deep learning algorithm, RNN applies weight matrix to the current and previous input and updates the weight using different approaches like backpropagation through time (BPTT) and gradient descent.

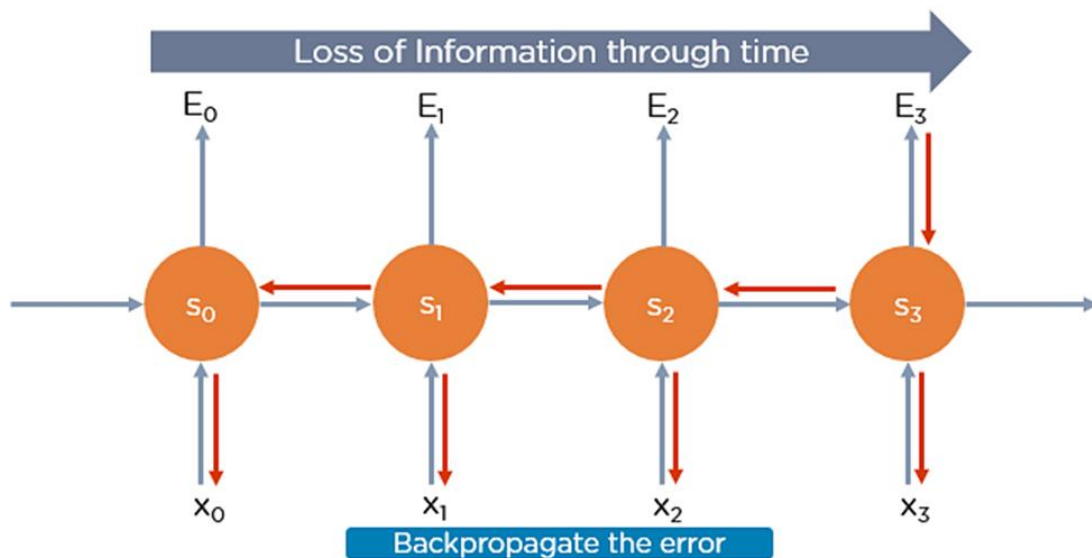


Figure 3.16: Backpropagation in RNN

Here, partial derivatives of the error concerning the weights are found via backpropagation. These derivatives are used by gradient descent algorithms to minimize the given function and adjust the weights up or down accordingly to decrease the error.[28] This backpropagation for an unrolled RNN is known as BPTT. Here, the error of a particular timestep is dependent on the preceding timestep, and the error is backpropagated to the first timestep from the latest timestep within this backpropagation through time (BPTT). While propagating the error, the network calculates the error for each timestep and updates the weights accordingly.

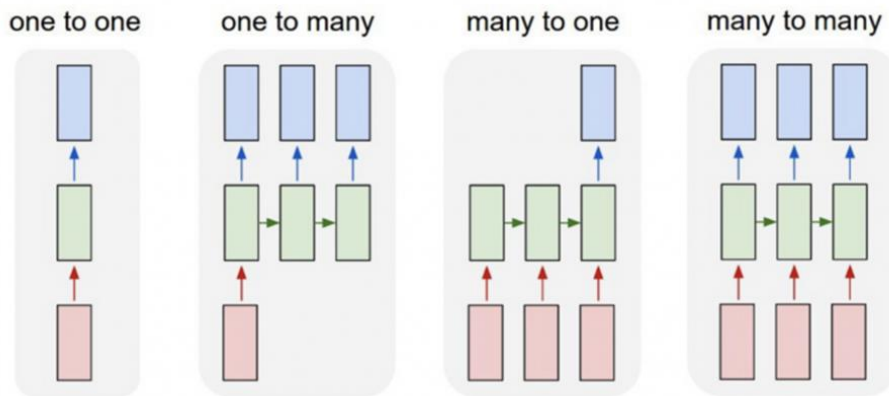


Figure 3.17: Different types of RNN

There are different types of RNN which are:

- One to One: One single input produces a single output.
- One to Many: One single input produces multiple outputs.
- Many to One: One single output is generated from multiple inputs.
- Many to Many: Multiple outputs are generated from multiple inputs.

Also, different activation functions like Sigmoid, Tanh, Relu can be used in RNN.

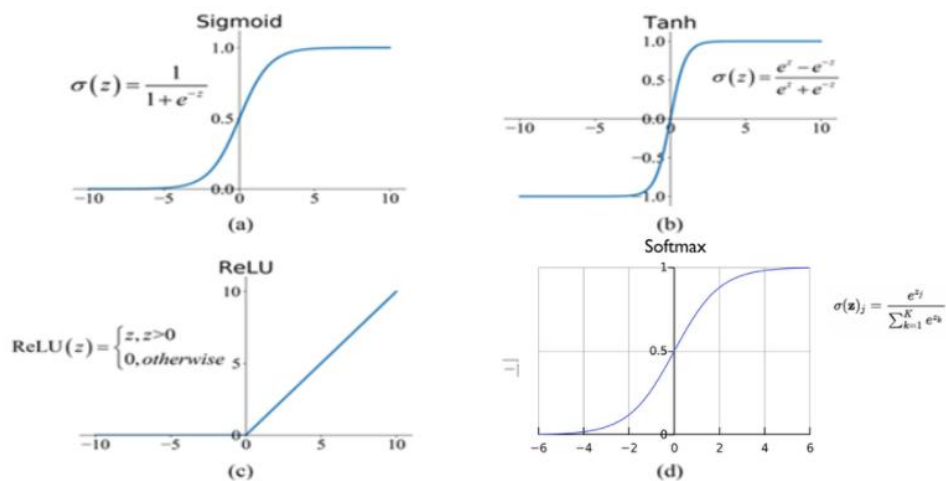


Figure 3.18: Different Activation functions

RNN is used for data that involves sequence like speech and voice recognition, text prediction, sentiment, time series data, image caption generation, part of speech tagging, machine translation, and transliteration. It can handle varying length input and can be very precise in predicting the sequence of data. However, as the network has to perform forward propagation, estimate error, and then backward propagation, the whole computation process can be very slow. Also, RNN suffers from the Exploding and Vanishing Gradient problem where, in vanishing gradients, values of a gradient gradually become too small to ignite the neurons. As a result, the model stops learning and tends to forget old information. On the other hand, in exploding gradients, the RNN algorithm assigns high importance to the weights which can create a very unstable network that is incapable of effective learning.

3.2.1 LSTM

Long short term memory (LSTM) is basically an extension of recurrent neural network (RNN) to overcome the limitations of RNN by having long short term memory.[29] As RNN suffers from short-term memory, it will perform poorly in terms of carrying older sequences to the newer ones. In the LSTM network, LSTM units are utilized as foundations for RNN layers. This enables RNN to have a long-term memory to remember inputs for a long time.

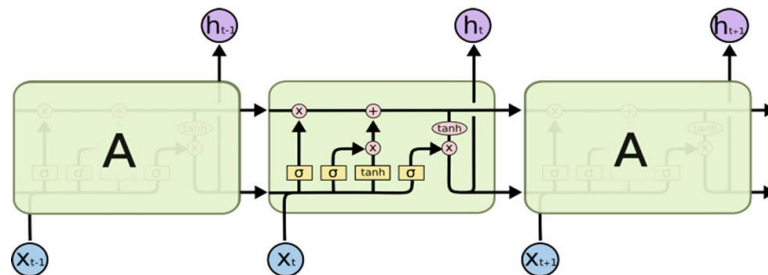


Figure 3.19: LSTM network

The core idea behind the long-term memory of LSTM is the cell state which changes slowly with minor linear interactions.[30] This cell state carries and transfers important information for the data sequence. Information is added or removed to the cell state over time and thus allows the LSTM network to remember old information for a longer period.[31]

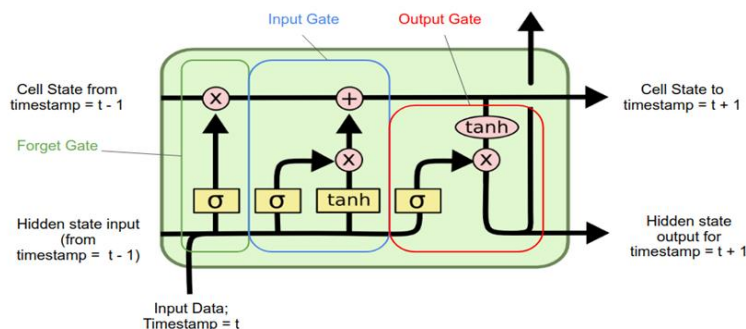


Figure 3.20: LSTM classification

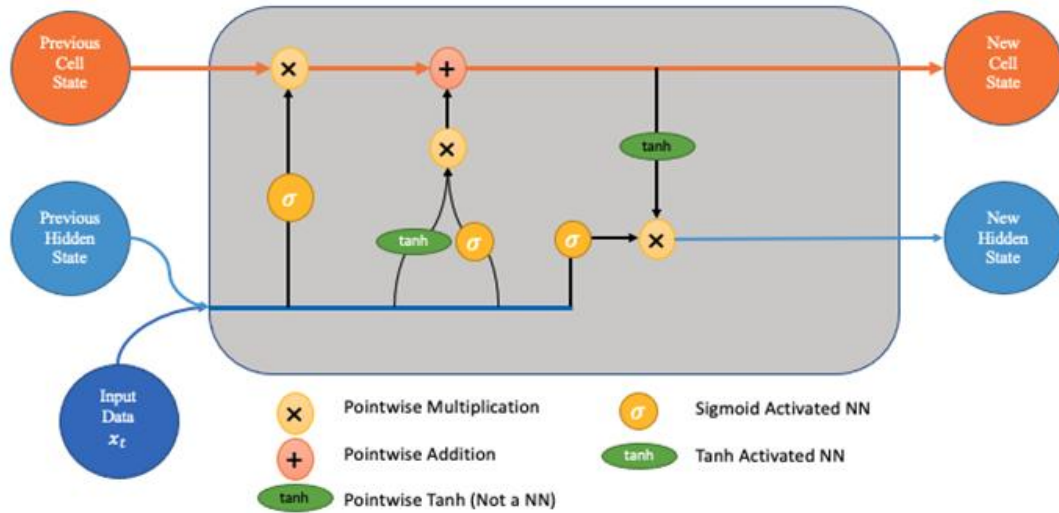


Figure 3.21: LSTM classification

LSTM has three gates where the forget gate determines how much data from the past information must be erased which are not important and what past information goes through the current state. The input gate controls what information should be added to memory (cell state) with present inputs. Lastly, the output gate controls what goes into output. The gates use sigmoid as activation function as the sigmoid gate can be used as a switch (0/1). Also, Tanh activation is used to regulate the values throughout the network.

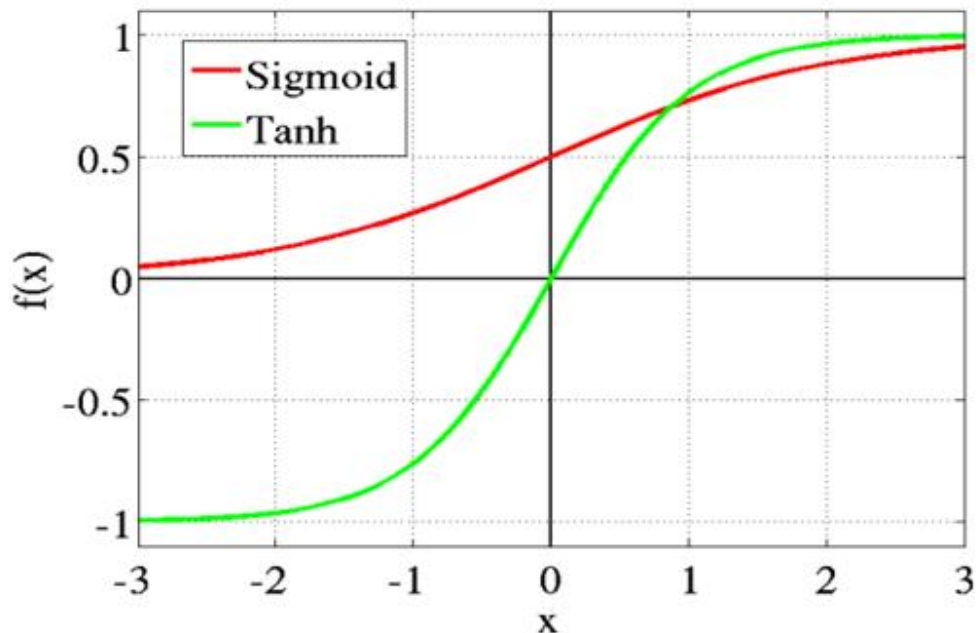


Figure 3.22: Sigmoid graph (Red) and Tanh graph (Green)

LSTM controls information sequences by using these three series of gates. The output of an LSTM of a particular timestamp is dependent on the cell state, output from the preceding concealed layer, and the current states input. At first, the for-

get gate controls the long-term memory according to output from the preceding concealed layer along with the input of the current layer. Then, new updated information is added to the memory (cell state) by the input gate given the output from the preceding concealed layer and the input of the current layer. Lastly, output gates decide output of the new concealed state according to the recently updated cell state, output from the preceding concealed layer, and the new input data.

The workflow of an LSTM network can be visible from the diagram below. Here, Z^f controls the forget gate, Z^i controls the input gate, Z updates information, and Z^o controls the output gate.

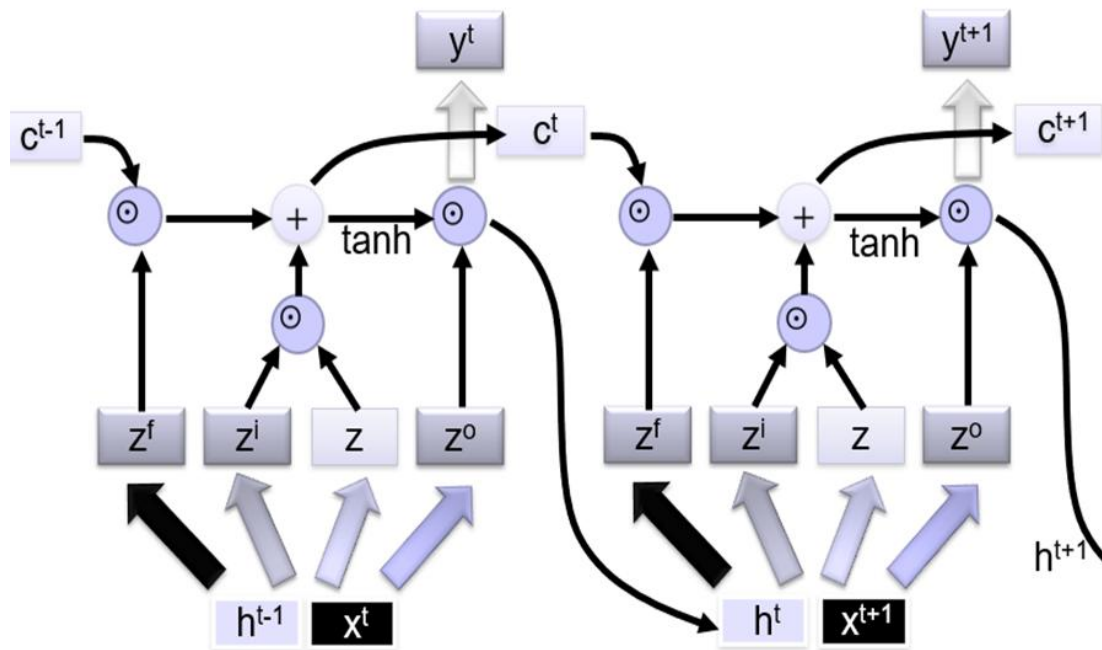


Figure 3.23: Work flow of LSTM

The problem that arises with RNN like vanishing gradients can be solved using LSTM. As the LSTM network has long short-term memory, it keeps the gradient step and retains old information. As a result, the network takes less time to train and gives a result with high accuracy. Also, LSTM can be very useful to improve the performance of the RNN network and can be used in any applications where long short-term memory is needed like grammar learning, speech recognition, handwriting recognition, time series prediction, along with music, and audio classification and recognition.

Chapter 4

Model Implementation and Optimization

In this chapter, we will go over our overall system workflow and all of the approaches we utilized to perform our study in depth. We will be describing all the actions we took to investigate and research for our paper. There are several ways a data can be preprocessed, based on the preprocessing, our accuracy varies. We aim to use our model in such a way that people, with hearing problems, can communicate with the rest of us freely.

Our device will be really easy to use, users will be easily be able to carry the device and use it whenever it will be necessary.

Sound recognition, in our paper, works in such a way that means it is the ability of our model to identify the surrounding environmental sound easily and view it to the users accordingly.

When any sound is received by our model, it will preprocess the sound and extract the feature later, to compare the sound with our database to find out the sound appropriately.

4.1 Data-Set

We intended to develop our own data-set for our article, but due to the rigorous lockdown and house quarantine we are maintaining, it was nearly impossible for us to access primary data. However, we did our best to find audio recordings online, and we acquired some primary data, such as the sounds of dogs, cats, and other basic raw audio files.

Previously, when we searched online for a data-set, we couldn't find the data-set we were looking for. For our thesis we needed a data-set, which will contain varieties of sound clips stored in it, after extraction. There were a lot of data-set which contained speech, but we didn't need that for our project, currently, as in the future we may dive into converting speech to text, so that people, who can barely listen without hearing aid, may communicate with others using our software.

We have not used any secondary data-set. We have identified 50 categories of sounds from our regular life, and started recording and collecting data from various sources, for our data-set. We have separated the categories among ourselves and gathered the sound clip in one place for preprocessing. For the sound clips, we have selected various categories, including- sounds of acoustic guitar, airplane, bicycle bell, cat, cheering, cow, crow, dog, duck, engine, fire alarm, flute, glass breaking and others.

We collected our sound files in mp3 format. Then we converted our sound files from mp3 to wav format. The conversation was done as we need the frequency and amplitude waves to process in our sound recognition, and the audio format, wav, is a waveform. After this conversion, we split the audio files into segments of 3 seconds with the exact sound, and we cut out the unnecessary sound from the audio files. Pydub, a library that can be imported in the Python programming language, is used to edit our sound files. We have used AudioSegment of pydub to convert our files from mp3 to wav files. We have used the extract feature of AudioSegment to convert the files to wav format.

We have used audio of length, which is equal to 1 hour or more, in our algorithm, the audio files which were more than 1 hour were trimmed down to 1 hour, for the simplicity of our calculation. AudioSegment from pydub have been used to convert mp3 audio format to wav format. Then the audio in wav format is read using the librosa package in the python programming language. We have used librosa as a package from python to analyze audio, the sample rate from the audio data obtained in the downsampling follows the default sample rate of the librosa package. Later pydub was further used to split the sound clips into frames of 3 seconds. After splitting, we extracted the features and then we stored the features in our data-set. Our data-set was balanced, all the categories covered around the same amount of storage in our CSV file. Which was good, as our data-set was not biased. data-set is extremely valuable for a machine learning project as we use data-set to train and test our model. Errors, faults, or biases in the data-set results in major problems for the whole project, as the calculated accuracy may not be entirely correct, causing the model to work wrongly.

Waveplot for some of the sound categories are given below -

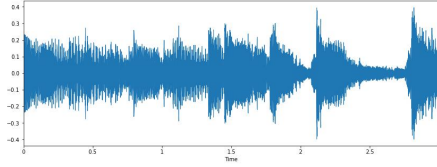


Figure 4.1: Waveplot for acoustic guitar

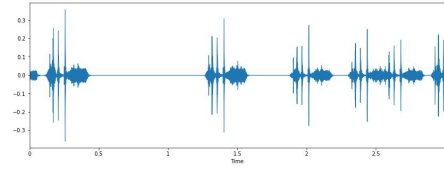


Figure 4.2: Waveplot for Camera

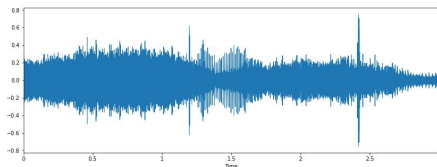


Figure 4.3: Waveplot for chain-saw

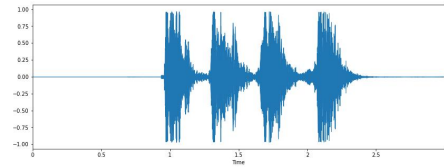


Figure 4.4: Waveplot for dogbark

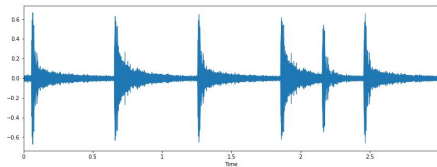


Figure 4.5: Waveplot for drums

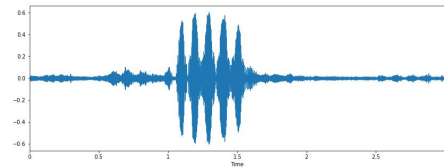


Figure 4.6: Waveplot for goat

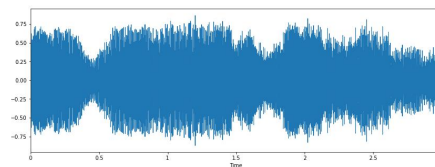


Figure 4.7: Waveplot for train

4.2 Data Preprocessing

Data preprocessing is an important part before implementing the data-set. Data preprocessing is the step of changing an original data set into a clean data-set. Data preprocessing is done in order to remove unnecessary data columns from the data-set, there might be some data containing any valid information at all, these data may cause difficulties to deal with in future, hence the importance of removing these data or fixing them before using them.

As we have created our own data-set, we had to collect different background noise. We included sound clips of birds, helicopters, dogs, cats, glass breaking and others. After collecting all our sound clips, we extracted every feature using melspectrogram. We then kept all our audio clips in a list, then from the list we have taken the data into a CSV file. Each data was assigned in its corresponding label.

Librosa is a Python package, which is used to analyze music and audio files. Basic purpose of librosa includes music generation and automatic speech recognition. `librosa.load` delivers a time series that is described as follows in the librosa glossary: "time series". We have used `librosa.load` with a duration of 3. Note that each of our audio files has a sample rate of 22050 Hz.

4.3 Features of Sound

Features a sound distinguishes the sound's category from other classes that are presented in our data-set. Each class has a certain feature that is used to identify them from other classes.

For our case we extracted our features using melspectrogram and then the data was stored in the data-set for the further process to take place.

4.4 Feature Extraction

The mel spectrogram converts hertz data to mel scale values. The mel scale is a set of tones that human hearing perceives as being equally spaced apart. The interval in hertz between mel scale values (or simply mels) grows as frequency rises. At lower frequencies, humans are better at perceiving differences than at higher frequencies.[32]

Few melspectrograms of our class are as follows -

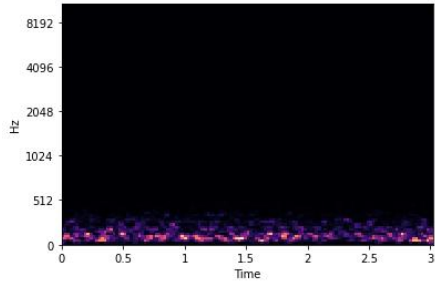


Figure 4.8: Melpectrogram of Air plane

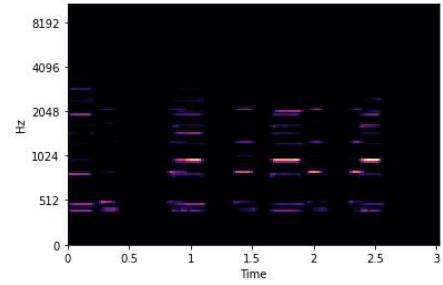


Figure 4.9: Melpectrogram of Car horn

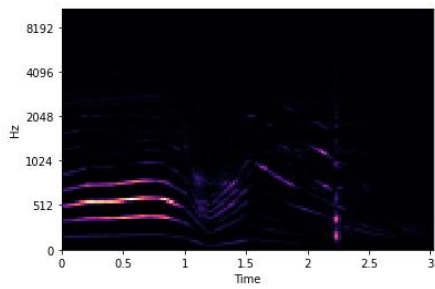


Figure 4.10: Melpectrogram of Chainsaw

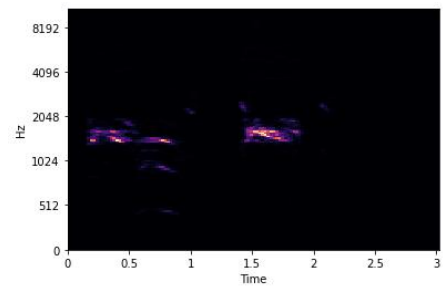


Figure 4.11: Melpectrogram of Crow

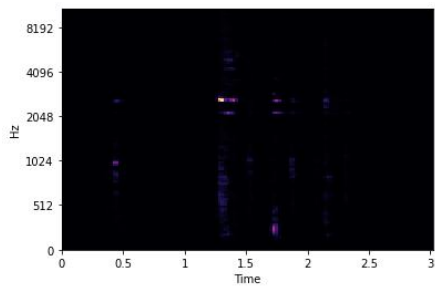


Figure 4.12: Melpectrogram of Glass break

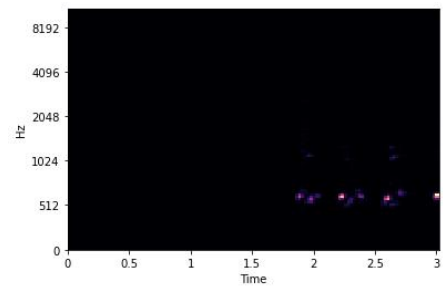


Figure 4.13: Melpectrogram of Dog

4.5 Feature Selection and Engineering

4.5.1 Feature selection

In order to optimize our model's performance and reduce the computational time and cost we use feature selection. In feature selection we mainly decrease the amount of input variables, so that our model can easily compute the data. Feature selection may help us with interpretation and visualization of data, and while overcoming the problem of various dimensionality for the enhancement of our model's performance, it can also decrease the duration of utilization, training time and need of storage.

In our model we have selected the feature as Audio file data and the rest of the corresponding classes are kept in label.

4.5.2 Feature Engineering

All machine learning algorithms require some input data, in order to generate outputs. Feature engineering, the act of developing new input features for machine learning, is an efficient approach to enhance prediction models. One of the primary aims of predictive modeling is to discover an effective and accurate predictive connection between a collection of accessible data and an outcome, such as the possibility of a customer doing a particular action. When utilizing machine learning to create a predictive model, feature engineering is the process of choosing and manipulating variables. It includes extracting essential information, emphasizing trends, and bringing in someone with domain experience, and it's an excellent approach to improve prediction models.

4.6 Train-Test split

The main purpose of train-test split is to find out how well our algorithm performs on a particular data-set. What we mainly do is we split our data-set into two parts, the first part is used for training. Usually we put more than half of the rows of a data set in the training portion, as it is the most vital part of a system. If our model is trained with more and correct data, our model will be able to distinguish our target more easily. The rest of the rows will be used to test our model, this is how we actually find out how well our model performs. As a result we can say that we will be splitting our data-set into two parts-

- i Training dataset.
- ii Testing dataset.

We have used scikit-learn's `train_test_split()` in order to split our data-set, we followed the standard ratio of 4:1 to split our data. Both of our features and labels will be split according to the ratio provided, which in our case is 4:1.

4.7 Models

We have used two models, which are- CNN and RNN. We have changed the hyper parameters for both the models individually, the detailed explanation of the models used are given below.

4.7.1 CNN

Conv2D of the CNN was used in our project.

The data-set was separated into features and label accordingly, using python programming language. Label was then encoded, in order for the algorithm to properly execute the model.

In order to test our model, we have split our data-set into 4:1. Our model contained multiple hidden layers, and we used the activation function of ReLU. ReLU, which stands for rectified linear unit, is a piecewise linear function that gives the output as input, whenever the input is positive and otherwise gives an output of zero. As ReLU disregards negative values, the output is zero whenever the input is negative, which results in information loss. The main reason of us using ReLU was that ReLU does not contain any vanishing gradient, furthermore ReLU is more computationally efficient than the Sigmoid activation function.

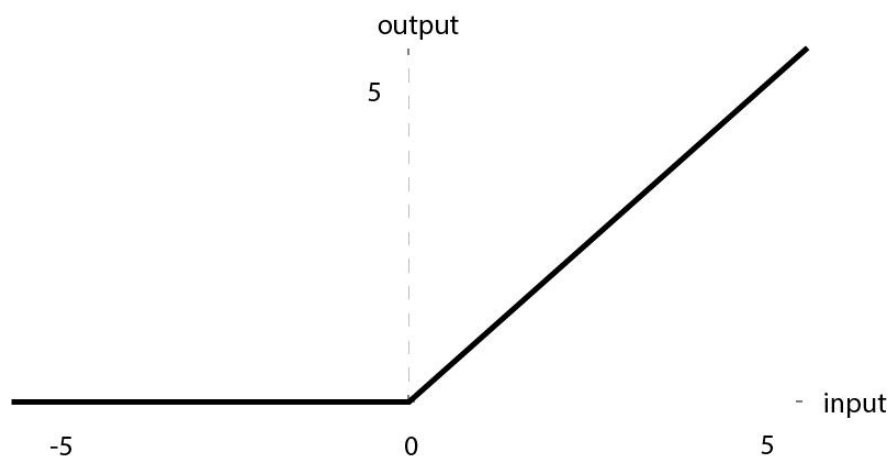


Figure 4.14: Input vs Output graph of ReLu

Output activation function of softmax was used, since softmax works better with multiclass classification problems.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (4.1)$$

Equation 4.2: Softmax Function

Optimizer is a function or algorithm, which is used to modify the attributes of a neural network. We have included Adam optimizer in our model. Adam optimizer uses adaptive learning. Among the adaptive learning optimizers Adam performs the best and it is fast too, which is the reason for us using Adam optimizer.

Learning rate of 0.001 was included.

Categorical crossentropy was included in our model as a loss function. In multi-class classification problems, categorical crossentropy is a loss function. After fitting the training data-set into our model we compiled it and ran it for 50 epochs.

4.7.2 RNN

Similar to the CNN model, RNN was used in the same environment. We have used LSTM of RNN, to test our model. We have split the data-set into 3:1 for the train and test, we have included 75% for training and the remaining data for testing.

Here we have used multiple hidden layers, as well. We have not used any activation function to the process, but we have used softmax in the output. The main advantage of softmax, over other activation functions, includes the output possibilities range. The sum of all possibilities is equal to 1 and the range of possibilities varies from 0 to 1. The reason why softmax is used is that the activation function helps us derive the probabilities of each class keeping the probability of the target class high. Dropout is generally introduced in a model to prevent that model from overfitting. Dropout for our RNN was 0.3.

Chapter 5

Results and Discussion

It is really tough for people to live day to day life without hearing properly. People with hearing problems need newer and better technology to use, in order to make their life easier. In this study we have primarily focused on creating a model, which can be implemented in software applications, in order to notify people with the sound that occurs surrounding them.

We have used Google Colaboratory to train all the models that have been used in this research. Google Colaboratory is a cloud-based Jupyter notebook environment that entirely runs in the clouds, and the whole notebook is integrated with Google Drive. We ran both CNN and RNN models for this experiment and both the models came up with almost the same accuracy values. CNN had an accuracy of 98.67% and RNN had an accuracy of 97.01%, we can see that from the bar chart (Figure 5.1). Both the models can be further compared in the classification report (Figure 5.1), as we have precision values, recall values, f1-score values, and support values from each of the 50 categories in this experiment. Even though CNN has slightly higher accuracy than RNN, we can see CNN has much higher computational time than RNN. RNN takes almost 1.5 hours to train the whole model, whereas CNN takes almost approximately 3 times more to train.

Initially when we first tested our program, we got an accuracy of 83.05%, but we didn't get our accuracy as we expected due to some clips which had no data in it at all, then we started to pre-process our data again in order to improve our accuracy. We ran each audio file to find the number of empty sounds it had, and resized the audio files accordingly, the files which had no audio at all were detected and we took them out of our data-set, we then added new audio files in replacement to the ones we deleted and then after test our code again we successfully managed to improve our accuracy to 98.67% using CNN and 97.01% using RNN.

According to the previous studies that were done for the audio classification, using the RNN model was not always the first choice due to poor accuracy values compared to the other models like ANN and CNN. From the results that were obtained from the experiment and research, we concluded that for the audio classification, RNN can be a very good choice. The result from our studies clearly shows that very high accuracy can be achieved with the models like CNN and RNN for audio classification,

98.67%, and 97.01% respectively, figure-x, making these models a reliable choice for further implementation. Our research objective was to provide an audio detecting system that can assist people with hearing disabilities and our system can detect different types of sounds in our environment to a great accuracy making the system a dependable option for further application and help the people in need.

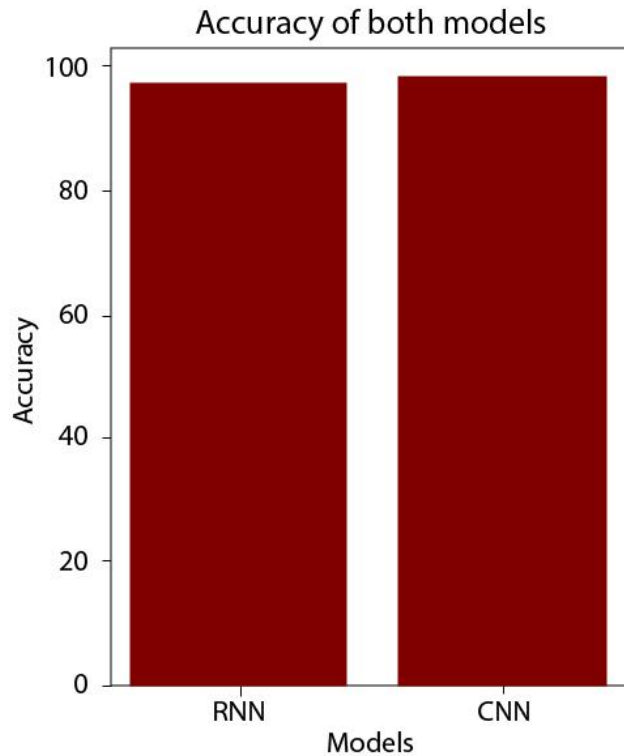


Figure 5.1: Accuracy comparison of RNN vs CNN

For CNN and RNN we have added the Classification report tables, table-1 and table-2. The classification reports contain precision, recall, f1-score and support values. Precision is the ability of the algorithm to find out the number of total data that is actually figured out properly by the model. It is generally shown as the ratio of total number of data figured and the total number of data that is present in the data-set. The ability of a model to accurately detect it's all positive cases is known as recall. The harmonic mean of accuracy and recall is calculated using the f1-score. The scores for each class indicate how accurate the classifier was in categorizing the data points in that class in comparison to all other classes. The number of samples of the genuine answer that fall into that class is the support. Accuracy refers to the number of correct guesses made by our model. Macro avg represents the mean average of all the values we have found. Because the weighted average takes into account how many of each class were used in the computation, less of one class implies its either precision or recall or F1 score has a smaller influence on the weighted average for each of those items.

CNN					RNN				
Classification Report					Classification Report				
Class	Precision	Recall	f1-Score	Support	Class	Precision	Recall	f1-Score	Support
Acoustic Guitar	0.92	0.94	0.93	317	Acoustic Guitar	0.87	0.88	0.87	317
Airplane	0.99	1.00	1.00	299	Airplane	1.00	1.00	1.00	299
Bicycle Bell	1.00	0.98	0.99	316	Bicycle Bell	1.00	1.00	1.00	316
Birds	0.99	1.00	0.99	280	Birds	1.00	0.99	0.99	280
Camera	1.00	1.00	1.00	305	Camera	1.00	1.00	1.00	305
Car Horn	1.00	1.00	1.00	322	Car Horn	1.00	0.98	0.99	322
Cat	1.00	0.93	0.96	284	Cat	0.99	0.92	0.95	284
Cello	0.97	0.83	0.90	313	Cello	0.87	0.85	0.86	313
Chainsaw	1.00	1.00	1.00	298	Chainsaw	0.97	1.00	0.99	298
Cheering	1.00	1.00	1.00	290	Cheering	1.00	1.00	1.00	290
Construction	1.00	1.00	1.00	290	Construction	0.99	1.00	0.99	290
Coughing	0.82	0.97	0.89	297	Coughing	0.83	0.90	0.86	297
Cow	0.99	0.99	0.99	294	Cow	0.98	0.96	0.97	294
Crow	1.00	1.00	1.00	297	Crow	0.99	0.98	0.98	297
Crowd	1.00	1.00	1.00	284	Crowd	1.00	1.00	1.00	284
Crying Baby	1.00	1.00	1.00	289	Crying Baby	0.99	1.00	1.00	289
Dog	0.99	0.98	0.99	298	Dog	1.00	0.91	0.95	298
Door Creeks	1.00	1.00	1.00	297	Door Creeks	0.95	0.97	0.96	297
Drums	1.00	0.99	1.00	295	Drums	0.99	0.99	0.99	295
Duck	1.00	0.95	0.98	283	Duck	0.97	0.92	0.95	283
Electric Guitar	0.98	0.98	0.98	293	Electric Guitar	0.96	0.97	0.96	293
Engine	1.00	1.00	1.00	295	Engine	1.00	1.00	1.00	295
Fire Alarm	1.00	1.00	1.00	315	Fire Alarm	1.00	1.00	1.00	315
FireTruck	1.00	1.00	1.00	305	FireTruck	0.99	1.00	1.00	305
Fireworks	1.00	1.00	1.00	309	Fireworks	0.99	0.97	0.98	309
Flute	0.95	0.98	0.96	311	Flute	0.97	0.95	0.96	311
Glass Breaking	1.00	1.00	1.00	321	Glass Breaking	0.99	0.98	0.98	321
Goat	0.99	0.98	0.99	308	Goat	0.96	0.94	0.95	308
Helicopter	1.00	1.00	1.00	315	Helicopter	1.00	1.00	1.00	315
KeyBoard	1.00	0.99	0.99	292	KeyBoard	0.98	0.95	0.97	292
Knocking	0.99	0.99	0.99	308	Knocking	0.99	0.95	0.97	308
Laughing	1.00	1.00	1.00	302	Laughing	1.00	1.00	1.00	302
Laughing Baby	0.99	1.00	1.00	291	Laughing Baby	1.00	0.98	0.99	291
Market	1.00	1.00	1.00	309	Market	0.99	1.00	1.00	309
Monkey	0.98	1.00	0.99	294	Monkey	0.99	0.97	0.98	294
Owl	1.00	1.00	1.00	291	Owl	0.99	0.99	0.99	291
Piano	0.95	0.95	0.95	325	Piano	0.96	0.96	0.96	325
Pigeon	1.00	1.00	1.00	296	Pigeon	0.98	1.00	0.99	296
Police Siren	1.00	1.00	1.00	285	Police Siren	0.94	0.99	0.96	285
Rain	1.00	1.00	1.00	311	Rain	1.00	1.00	1.00	311
Rooster	0.99	0.98	0.98	293	Rooster	0.84	0.95	0.89	293
Running Footsteps	1.00	1.00	1.00	266	Running Footsteps	1.00	1.00	1.00	266
Saxophone	0.94	0.93	0.93	302	Saxophone	0.95	0.80	0.87	302
Sheep	1.00	0.99	1.00	296	Sheep	0.99	0.99	0.99	296
Siren	0.99	1.00	1.00	329	Siren	0.97	1.00	0.98	329
Snoring	1.00	1.00	1.00	285	Snoring	1.00	1.00	1.00	285
Train	1.00	0.99	1.00	303	Train	0.96	0.98	0.97	303
Violin	0.97	1.00	0.99	269	Violin	0.88	1.00	0.94	269
Walking Footsteps	1.00	1.00	1.00	311	Walking Footsteps	0.99	1.00	1.00	311
Washing Machine	0.97	0.99	0.98	306	Washing Machine	0.91	0.98	0.94	306

Table 5.1: Classification Report for CNN and RNN

CNN					CNN				
Classification Report					Classification Report				
Class	Precision	Recall	f1-Score	Support	Class	Precision	Recall	f1-Score	Support
accuracy	-	-	0.99	14984	accuracy	-	-	0.97	14984
macro avg	0.99	0.99	0.99	14984	macro avg	0.97	0.97	0.97	14984
weighted avg	0.99	0.99	0.99	14984	weighted avg	0.97	0.97	0.97	14984

We can see the accuracy of using CNN from Table 5.1, which represents the accuracy to be approximately around 100%, we can see that 29 categories out of 50 categories give a precision of 1, which means 100%. Over all the accuracies of recall, f1-score and support was satisfactory, which resulted in the average values, macro avg and weighted avg, to be high.

A confusion matrix summarizes the total outcomes of the model, it represent the ability of the model to guess the output proper and shows how much of a class is categorized into other class, the confusion matrix of CNN is given below, in Figure 5.2 -

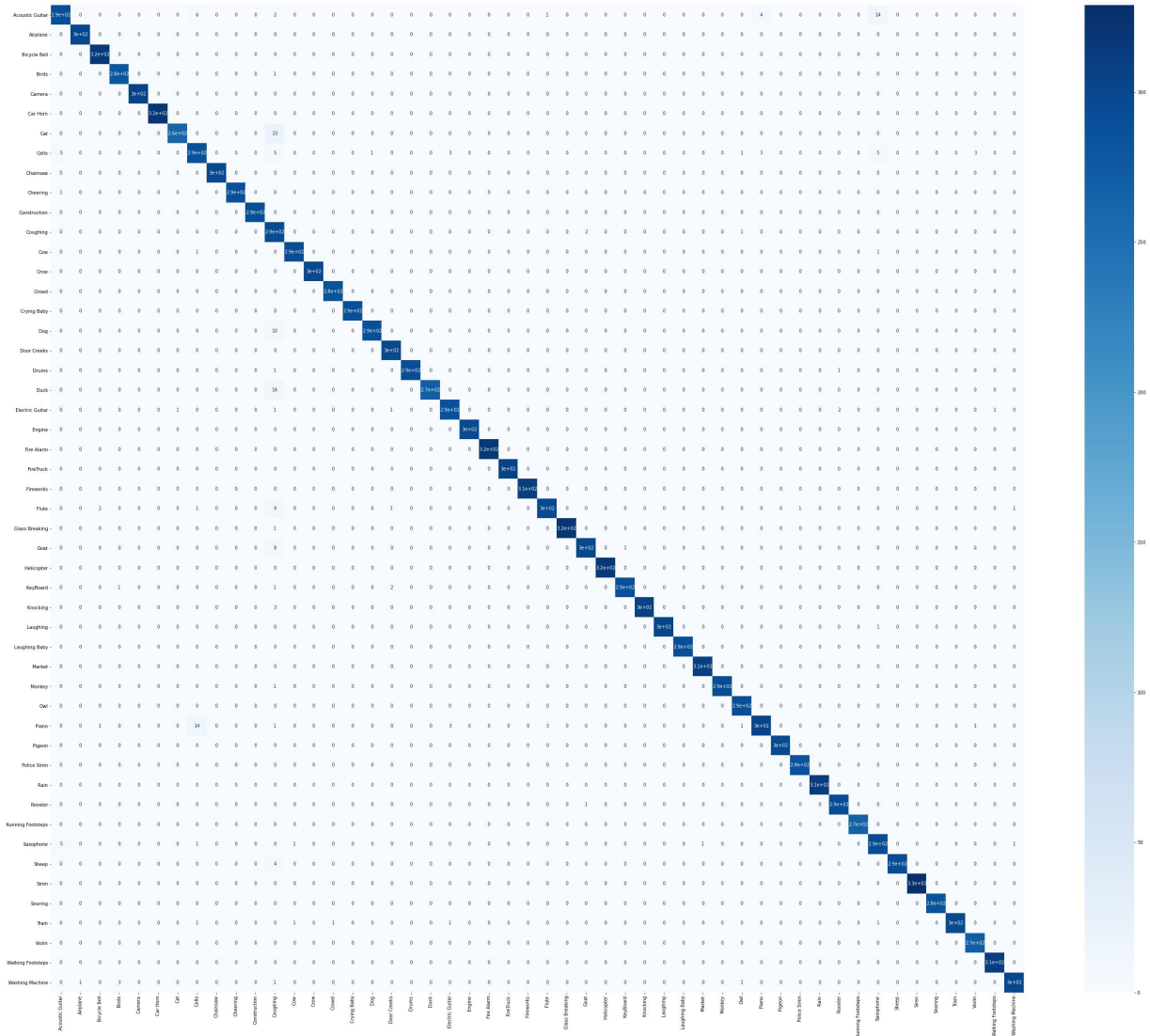


Figure 5.2: Confusion matrix of CNN

RNN, which took less time to train compared with the other model, CNN, comparatively gave lower accuracies. We can see that, a total of 16 categories gives the accuracy of 100% in RNN, which is less than 50 % of the categories we have used in our data-set, across the table the probabilities calculated were not as high as that of CNN. The overall avagares, macro avg and weighted avg, were 2% less than that of CNN. RNN is less powerful than CNN as RNN contains less feature compatibility compared to CNN.

Confusion matrix of RNN is given below in Figure 5.3 -

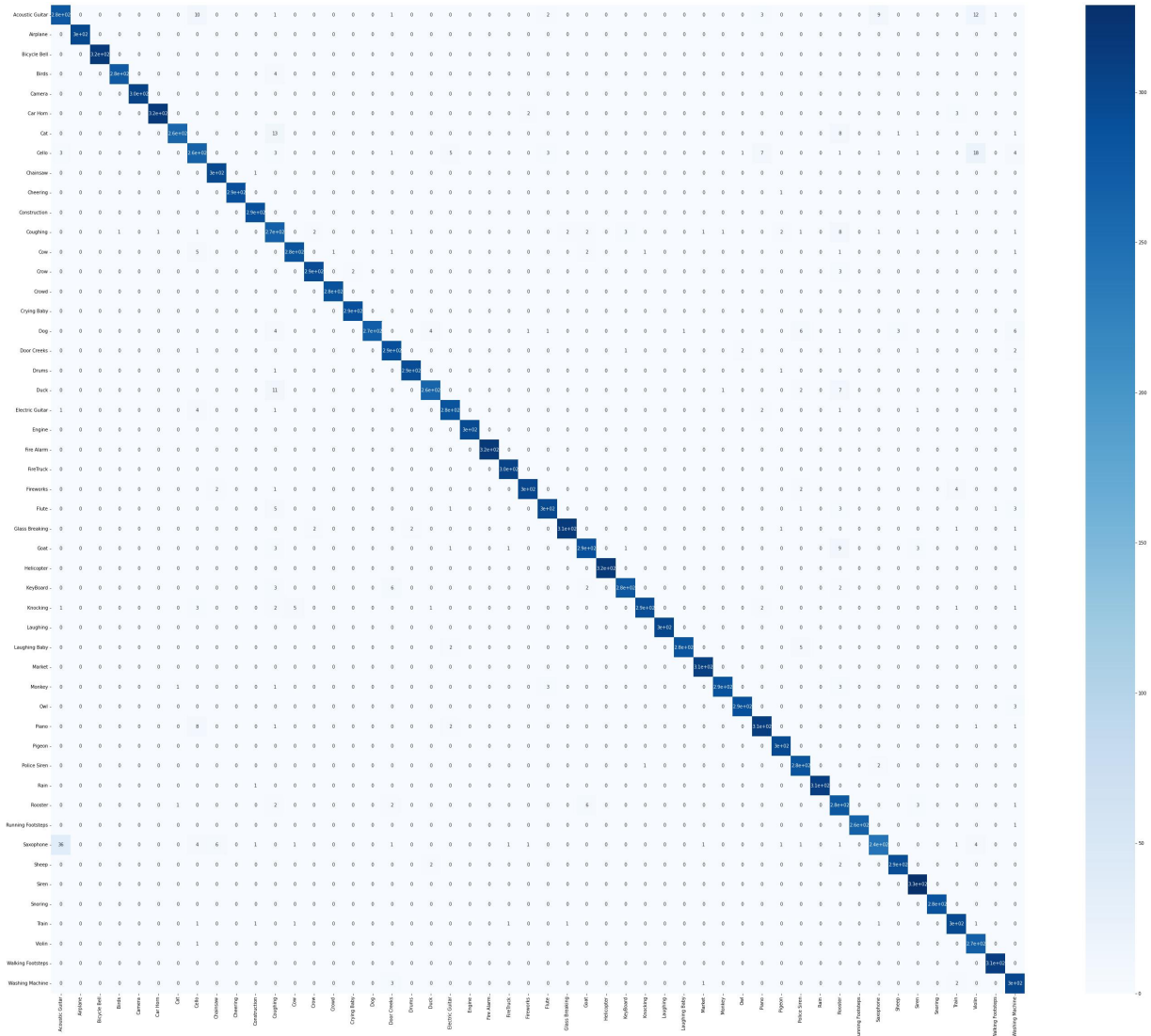


Figure 5.3: Confusion matrix of RNN

Conclusion

The main purpose of our research was to provide the hearing impaired people with a model, which they will be able to use in their everyday life to distinguish between sounds properly and being able to get notified whenever any sound occurs around them. The model will help the people to differentiate between sounds, which will eventually help them to take proper decisions and again be a fully functioning members of our society. Our model is designed to provide the deaf people with better understanding of their surroundings and the environment they live in. We have created our own data-set, for feeding to the algorithm we used, we have written our own code in Python. We have used audiossegment of Pydub, a library of Python programming language, to trim and split our audio files, AudioSegment from pydub was used to process our files primarily. Later after creating our data-set, we have extracted our features using melspectrogram of librosa, which is a Python package. Then we used the RNN and CNN algorithm to utilize our project. And finally, with our model we have successfully been able to get an accuracy of 98.67%, and 97.01% using CNN and RNN accordingly.

Being a developing country Bangladesh has seen a good margin of GDP growth in the past decade, even in 2019 Bangladesh successfully achieved a total GDP growth of 8.15%, which was the highest GDP growth Bangladesh ever gained[33]. Back in 2014, amount of people, with hearing disabilities was approximated to be around 13 million,[34] which may have already crossed above 20 million, utilizing this people, making them as productive as any other normal human being will benefit us a lot, and we can also ensure a better future for our coming generation. If they are helped properly, these 20 million people will have a huge positive effect on the GDP of bangladesh. It needs to be developed with a powerful user interface in a developed system for individuals with disabilities. If our model is implemented properly, it will really help the hearing impaired contribute to society equally as others.

In the future, we would like to execute our project in real life. Our plan is to create a software which people will be able to use on their smartphone or smartwatch. Our application will be able to detect sounds and vibrate the phone, if any similar sound clips are found. For deaf people, whenever the phone vibrates, we will be portraying an image for the sound that occurred in the surroundings of the person. Users will be able to choose the sounds, for which they want to get notified. For example, if the user chooses to get notified whenever any glass breaks around him, or any car honks, they will be able to select the sounds and will get notified accordingly. Furthermore, our future plan is to implement our project in such a way that will help the people with hearing disability, to make a conversation with any person talking with them, without the use of an expensive hearing aid machine.

Bibliography

- [1] K.Karthikeyan and D. R.Mala, “Content based audio classifier feature extraction using ann techniques,” *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 5, 04 2018. DOI: 10.26562/IJIRAE.2018.APAE10081.
- [2] G. Wu, “Metabolism and functions of amino acids in sense organs,” *Adv Exp Med Biol.* 2020, p. 1265, 2020. DOI: 10.1007/978-3-030-45328-2_12.
- [3] M. Yağanoğlu, “Real time wearable speech recognition system for deaf persons,” *Computers & Electrical Engineering*, vol. 91, p. 107 026, 2021.
- [4] F. Z. Siddiqua, “The silent conversation,” *Computers & Electrical Engineering*, 2014.
- [5] M. Alauddin and A. H. Joarder, “Deafness in bangladesh,” *Hearing Impairment*, 2004. DOI: 10.3329/bjo.v26i2.50605.
- [6] M. M. Othman O. Khalifa and M. Baharom, “Hearing aids system for impaired peoples,” *International Journal of Computing and Information Sciences*, vol. 2, pp. 23–26, Jan. 2004.
- [7] R. A. Dobie and S. B. V. Hemel, “Hearing loss: Determining eligibility for social security benefits.,” *Impact of Hearing Loss on Daily Life and the Workplace.*, vol. 6, 2004.
- [8] *Deafness and hearing loss*, en. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> (visited on 01/19/2022).
- [9] C. Freeman, R. Dony, and S. Areibi, “Audio environment classification for hearing aids using artificial neural networks with windowed input,” pp. 183–188, 2007. DOI: 10.1109/CIISP.2007.369314.
- [10] S. L. Büchler M. Allegro, “Sound classification in hearing aids inspired by auditory scene,” p. 387 845, 2005. DOI: <https://doi.org/10.1155/ASP.2005.2991>.
- [11] S. Hershey, S. Chaudhuri, D. P. Ellis, *et al.*, “Cnn architectures for large-scale audio classification,” in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, IEEE, 2017, pp. 131–135.
- [12] E. R. Nascimento, R. Bajcsy, M. Gregor, I. Huang, I. Villegas, and G. Kurillo, “On the development of an acoustic-driven method to improve driver’s comfort based on deep reinforcement learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2923–2932, 2021. DOI: 10.1109/TITS.2020.2977983.

- [13] H. Meinedo and J. Neto, “A stream-based audio segmentation, classification and clustering pre-processing system for broadcast news using ann models,” in *Ninth European Conference on Speech Communication and Technology*, Cite-seer, 2005.
- [14] H. Meinedo, D. Caseiro, J. Neto, and I. Trancoso, “Audimus. media: A broadcast news speech recognition system for the european portuguese language,” in *International Workshop on Computational Processing of the Portuguese Language*, Springer, 2003, pp. 9–17.
- [15] Z. Kons, O. Toledo-Ronen, and M. Carmel, “Audio event classification using deep neural networks.,” in *Interspeech*, 2013, pp. 1482–1486.
- [16] S. Tangkawanit and S. Kanprachar, “Spectral vector design for gunfire sound classification system with a smartphone using ann,” in *2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, IEEE, 2018, pp. 421–426.
- [17] E. Rençberoglu, “Fundamental techniques of feature engineering for machine learning,” *Towards Data Sci*, 2019.
- [18] P. Khaleelur Rahiman, V. Jayanthi, and A. Jayanthi, “Retracted: Speech enhancement method using deep learning approach for hearing-impaired listeners,” *Health informatics journal*, vol. 27, no. 1, p. 1 460 458 219 893 850, 2021.
- [19] P. Rahiman, J. Vs, and J. A.N., “Speech enhancement method using deep learning approach for hearing-impaired listeners,” *Health Informatics Journal*, vol. 27, p. 146 045 821 989 385, Jan. 2020. DOI: 10.1177/1460458219893850.
- [20] S. Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, en, Dec. 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [21] *CNN for Deep Learning — Convolutional Neural Networks*, en, May 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>.
- [22] A. Dertat, *Applied Deep Learning - Part 4: Convolutional Neural Networks*, en, Nov. 2017. [Online]. Available: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>.
- [23] H. Sneha, *2d convolution in image processing*, en, 2018. [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/two-dimensional-convolution-in-image-processing/>.
- [24] NamyaLG, *What is 2-Dimensional Convolution?* en, Apr. 2021. [Online]. Available: <https://medium.com/theleanprogrammer/2-dimensional-convolution-189abb174d92>.
- [25] I. Shafkat, *Intuitively Understanding Convolutions for Deep Learning*, en, Jun. 2018. [Online]. Available: <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>.
- [26] M. Saeed, *An Introduction To Recurrent Neural Networks And The Math That Powers Them*, en-US, Sep. 2021. [Online]. Available: <https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/>.

- [27] A. Biswal, *Recurrent Neural Network (RNN) Tutorial: Types and Examples [Updated]* — *Simplilearn*, en-US.
- [28] T. P. Lillicrap and A. Santoro, “Backpropagation through time and the brain,” en, *Current Opinion in Neurobiology*, Machine Learning, Big Data, and Neuroscience, vol. 55, pp. 82–89, Apr. 2019, ISSN: 0959-4388. DOI: 10.1016/j.conb.2019.01.011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959438818302009>.
- [29] N. Donges, *Recurrent Neural Networks (RNN): What It Is & How It Works — Built In*, en. [Online]. Available: <https://builtin.com/data-science/recurrent-neural-networks-and-lstm>.
- [30] C. Olah, *Understanding LSTM Networks – colah’s blog*. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [31] M. Phi, *Illustrated Guide to LSTM’s and GRU’s: A step by step explanation*, en, Jun. 2020. [Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
- [32] E. A. Lopez-Poveda, “Development of Fundamental Aspects of Human Auditory Perception,” en, in *Development of Auditory and Vestibular Systems*, Elsevier, 2014, pp. 287–314, ISBN: 9780124080881. DOI: 10.1016/B978-0-12-408088-1.00010-5. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/B9780124080881000105>.
- [33] *Bangladesh GDP Growth Rate — 2022 Data — 2023 Forecast — 1994-2021 Historical — Chart*. [Online]. Available: <https://tradingeconomics.com/bangladesh/gdp-growth>.
- [34] *Statistical Yearbook for Asia and the Pacific 2014*, EN. [Online]. Available: <https://www.unescap.org/publications/statistical-yearbook-asia-and-pacific-2014>.