2020

Classification of Animal Sound Using Convolutional Neural Network

Neha Singh
Technological University Dublin

Recommended Citation

Singh, N. (2020). *Classificatin of animal sound using convolutional neural network. Masters Dissertation. Technological University Dublin. DOI:10.21427/7pb8-9409*

Classification of Animal Sound using Convolutional Neural Network

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

Neha Singh

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Analytics)

04 January 2020

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: NEHA SINGH

Date: 04 January 2020

Abstract

Recently, labeling of acoustic events has emerged as an active topic covering a wide range of applications. High-level semantic inference can be conducted based on main audio effects to facilitate various content-based applications for analysis, efficient recovery and content management. This paper proposes a flexible Convolutional neural network-based framework for animal audio classification. The work takes inspiration from various deep neural network developed for multimedia classification recently. The model is driven by the ideology of identifying the animal sound in the audio files by forcing the network to pay attention to core audio effects present in the audio to generate Mel-spectrogram. The designed framework achieves an accuracy of 98% while classifying the animal audio on weekly labelled dataset. The state-of-the-art in this research is to build a framework which could even run on the basic machine and do not necessarily require high end devices to run the classification.

Keywords: CNN, VGG

Acknowledgments

First of all, I would like to thank my Supervisor Dr.Bojan Bozic for providing valuable feedback throughout this study. It was a honor to work and study under his supervision.

I would also like to thank Dr. Luca Longo, M.Sc. theses coordinator, for his useful inputs in the formulation without which this dissertation would not have been conceivable.

I'm deeply indebted to all the TUD staff at the Computer College for their constant support and guidance in completing the thesis.

My special thanks to all my classmates and friends in providing relevant assistance and help in completing the thesis.

Lastly, Love and regards to my family for their constant support and inspiration.Special thanks to everyone who believed in me!

Contents

Declaration **I**

Abstract **II**

Acknowledgments **III**

Contents **IV**

List of Figures **VII**

List of Tables **X**

List of Acronyms **XI**

1 Introduction **1**

1.1 Research Focus 2

1.2 Background 2

1.3 Research Problem 3

1.4 Research Objectives 7

1.5 Research Methodologies 8

1.6 Scope and Limitations 9

1.7 Document Outline 10

2 Review of existing literature **12**

2.1 Identification of the content using the information present in the picture 13

2.2 Identification of the content using the Multimodal- picture, and text . . 14

2.3 Audio event recognition . . . 15
2.4 Machine Learning methods of classification . . . 17
2.4.1 Support vector machine . . . 18
2.4.2 Hidden Marvol . . . 20
2.4.3 Gaussian mixture models . . . 21
2.5 Deep Learning classification model . . . 22
2.5.1 Classifications using the Audio content in the Video . . . 24
2.5.2 Summary of Previous work . . . 26
2.5.3 Gaps . . . 28
2.5.4 State of the art to solve the problem . . . 28
3 Experiment Design and Methodology **32**
3.1 Project approach . . . 32
3.2 Design aspects . . . 33
3.3 Detailed Design and Methodology . . . 34
3.4 Data Understanding . . . 36
3.4.1 Data Collection . . . 36
3.4.2 Data Description . . . 37
3.4.3 Data Preprocessing . . . 39
3.4.4 Data Split . . . 41
3.4.5 Feature Extraction . . . 41
3.4.6 Data Exploration . . . 42
3.4.7 Modeling . . . 44
3.4.8 Evaluation . . . 50
4 Implementation and Results **52**
4.1 Software . . . 52
4.2 Data Modeling . . . 52
4.2.1 Model 1: Convolution neural network . . . 53
4.2.2 Comparative Models . . . 55
4.3 VGG pre-trained model on AudioSet . . . 60

4.3.1 Implementation 60

4.3.2 Summary of Task 2 62

4.4 Results 62

5 Evaluation and Analysis 64

5.1 Introduction 64

5.2 Evaluation of the result 65

5.3 Evaluating the model for Animal Audio classification 71

5.4 Strength and Limitation of Result 75

6 Discussion and Conclusion 76

6.1 Research Overview 76

6.2 Problem Definition 77

6.3 Experimentation, Evaluation and Results 77

6.4 Contributions and Impact 78

6.5 Future Work and Recommendations 79

References 80

Appendices 85

.1 Data Exploration and Model Implementation 86

List of Figures

1.1 Audio Waveform Sample 5

1.2 Two Dimensional Image 5

1.3 Time Frequency representation 6

2.1 SVM Classification 19

2.2 Gaussian Mixture Equation 21

2.3 Deep Learning Architecture 22

2.4 Gated CNN Architecture 24

2.5 Spectrogram 31

3.1 Work Flow Diagram 33

3.2 CRISP-DM Cycle 34

3.3 Distribution of Classes in Dataset 38

3.4 Audioset Description 39

3.5 Data Description 39

3.6 Data Label 40

3.7 Bar Plot Analysis 41

3.8 Mel-Spectrogram-Bat 43

3.9 Mel-Spectrogram-Bird 43

3.10 Mel-Spectrogram-Lion 43

3.11 Audioset Description 44

3.12 Data Description 44

3.13 Sample Convolutional Architecture 45

3.14 CNN Image frame 46
3.15 CNN Image frame selection flow 46
3.16 CNN Image process flow 47
3.17 Activation Parameter value 48
3.18 Flatten Layer 49
3.19 VGG-Transfer Learning Process 49
3.20 Evaluation Metrics 51
3.21 Confusion Matrix 51

4.1 CNN Architecture 53
4.2 Model 1 Architecture 54
4.3 Model 1 Summary 56
4.4 Model 2 CNN Architecture 57
4.5 Model 2 Summary 58
4.6 Model 2 Architecture 58
4.7 Model 3 SVM 59
4.8 Model 3 Architecture 60
4.9 Model 3 XGBoost 60
4.10 Pre Required Files 61
4.11 Smoke Test 61

5.1 Model 1 Performance 65
5.2 Model 1 Classification Report 66
5.3 Model 2 CNN Performance 66
5.4 Model 2 Classification Report 68
5.5 Model 3- Classification report for Testing 70
5.6 Confusion matrix-Model 1 72
5.7 Confusion matrix-Model 2 73
5.8 Confusion matrix-Model 3 73

1 Tiger Amplitude plot 86
2 Sea Lion Amplitude plot 86

3 Camel Amplitude plot 87
4 Bat Amplitude plot 87
5 Fox and Coon Amplitude plot 88
6 Panther Amplitude plot 88
7 Bees Amplitude plot 89
8 Canary Animals Amplitude plot 89
9 Wolf Amplitude plot 90
10 Bear Amplitude plot 90
11 Donkey Amplitude plot 91
12 Seal Amplitude plot 91
13 Pig Amplitude plot 92
14 Dolphin Amplitude plot 92
15 Raccoon Amplitude plot 93
16 Whale Amplitude plot 93
17 Family Sciuridae Amplitude plot 94
18 Horse Amplitude plot 94
19 Frog Amplitude plot 95
20 Monkey Amplitude plot 95
21 Lion Amplitude plot 96
22 Cattle Animals Amplitude plot 96
23 Dog Amplitude plot 97
24 Cat Amplitude plot 97
25 Classification Report Training - Model 2 98
26 Classification Report Validation - Model 2 99
27 Classification Report Training - Model 3 100
28 Classification Report Validation - Model 3 101

List of Tables

4.1 Experiment Result 62

5.1 SVM as a classifier using the output of CNN 67

5.2 CNN-SVM Evaluation 67

5.3 Model 3 Evaluation 69

5.4 Comparative Summary 74

List of Acronyms

CNN	Convolutional neural network
SVM	support vector machine
VGG	Visual geometry group
ReLU	Rectified Linear Unit Activation Function
AudioSet	Sound Vocabulary Dataset
CRISP-DM	Cross-Industry Standard Process for Data Mining

Chapter 1

Introduction

In recent years, the ubiquity of multimedia services along with proliferation of mobile devices are not only changing life experiences but has drastically changed the Internet trends. Over past few years the world of multimedia has seen a drastic increase in the number of videos and audio uploaded every minute. You Tube, World's first largest multimedia search gets an hourly upload of around 300 hours of videos. This rapid increase of information imposes new demands of content management. Artificial intelligence and machine learning are amongst the most significant technological developments in recent history, touching virtually every aspect of business operations of major companies. Machine learning applications in five elements of multimedia Text, image, audio, video and animation has been remarkable with advancement in machine learning. Some of the real lives example of machine learning ranges from improving user experience on Yelp with the algorithms compiling, categorizing and labelling images more effectively. Machine learning algorithm also evaluates every tweet and messages on social networking sites, twitter and Facebook messenger making it easier to filter out spams and providing curative feeds. It provides algorithm to curate content for Pinterest on Image Data. One of the ongoing applications of machine learning, Baidu- The Future of voice search, a deep neural network that can generate entirely synthetic human voices that are very difficult to distinguish from genuine human speech. Although machine learning has shown promise in various applications such as speech and object recognition, it has not yet met the expectations for other fields such as audio concept classification Ravanelli, Elizalde, Ni, and Friedland (2014). Managing and processing content using audio files is still an unexplored part of machine learning. With enormous number of multimedia files uploaded online, content management using audio is a key area of machine learning application.

1.1 Research Focus

The focus of the project is two-fold. It sets out with building research theory by processing and analyzing animal audio sounds of various classes followed by classification of audio and labelling. The research concludes with evaluation of classification performance using various machine learning techniques to classify audio into various animal class.

1.2 Background

With the pervasion of cellphones in daily life and the popularity of video-sharing websites such as YouTube, Facebook in recent years have seen an explosive increase in the number of consumer-produced videos. The means of retrieving videos in such large collections are still mostly limited to text-based search in human-generated video metadata or voice captions, rather than the actual content of a video Hassanzadeh and Wang (2016). Given the exponential growth of videos published Mertens, Lei, Gottlieb, Friedland, and Divakaran (2011a) and uploaded, multimedia content analysis for indexing and retrieval is perhaps a key factor in understanding the content. Demands for automatic management and retrieval Lu, Ge, Zhao, and Yan (2010) have become a major research area. Most of video classification are based on picture frames of video which fails if the video quality is not good. While the visual content of the clips often contains most useful information for event detection Hassanzadeh and Wang (2016). Audio can provide important cues to the semantics of data Hershey et al. (2017a). Several recent works have paid attention to exploring more concrete key audio effects with high-level semantics Cai, Lu, Hanjalic, Zhang, and Cai (2006) to detect event.

Extracting audio from the video is a simplified process of converting .mp4 format file in .wav format. Our key idea is to consciously identify animal sound Chou, Jang, and Yang (2017) by forcing the network to pay attention to acoustic details in the clip and categorize the contents by audio tagging.

There has been a strong convergence towards applying computer vision approaches to audio classification. By virtue, if we are able to represent audio signals as images, we will be able to take advantages of well-known image classification techniques to audio classification problems. Nearly 200 years ago, French mathematician Fourier proved that certain function can be represented as infinite sum of harmonics or approximate it with an arbitrary precision by using a finite sum. On the basis of Fourier's work, we can transform the representation of an audio signal as a function of time into a representation of frequencies and vice versa. This relatively trivial stated fact has proved notably to be a major contribution to various scientific achievements ever since Fourier first studied it. On similar lines, we are able to represent audio as an image by using Fourier transform algorithms.

1.3 Research Problem

Ever since the birth of the digital era and the availability of web-scale data exchanges, researchers in these fields have been working hard to design more and more sophisticated algorithms to index, retrieve, organize and annotate multimedia data Gemmeke et al. (2017a). Audio event recognition, human-like ability to identify and relate sounds from audio, is a nascent problem in machine learning Deng et al. (2009). Another problem in audio classification is unlike image classification where the objects are centered and occupy dominant part of the image, audio events may only occur in a short span of audio recording Boreczky and Wilcox (1998). Accurate audio tagging thus relies on the amount of labelled audio data, including clip level labelled audio data and event level labelled audio data. However, creating such labelled audio dataset is very difficult and expensive. Recently, there have been astonishing results in comparable problem with object tagging in image detection from comprehensive dataset-ImageNet. Google thus launched an AudioSet having weakly labelled sound events from all domains. Unlike previous work, data from all domains including animals can be used to create a system that recognizes and categorizes animal and their activities in the video. Utilizing the google dataset requires huge computational power to process and convert the audio embeddings provided by Google. Many researchers have extracted the audio features from the embeddings and have made it available publicly.

Many audio classication methods are based on the bag of frames Gemmeke et al. (2017a) assumption, where an audio recording is cut into segments and each segment inherits the labels of the audio recording. However, this assumption is incorrect because some audio events only happen for a short time in an audio clip. Prior works in acoustic detection propose a fully supervised training model built on a training dataset that contains the annotation of the temporal position of the acoustic events Mesaros, Heittola, Eronen, and Virtanen (2010a). Strongly labelled data works best with fully supervised training model but collecting the data is its biggest limitations. In comparison, the weekly labelled data only requires annotations of the occurrence of the acoustic events and therefore is easier to amass and can be used to train weekly supervised learning model to classify sounds.

Recently state-of-the-art recommends transforming audio files as Images. However, representing audio files as images is change of dimension cardinality with Images being represented in two-dimension measure and audio files being single time dimension. Fourier transformation algorithm can be used to transform the waveform to the time-frequency(T-F) representation. It works on associated principle of representing the signal depending on sample rate, which is number of samples per second of audio. If a 3 second audio clip has a sample rate of 44,100Hz, that means it is made up of 3*44,100 =132,300 consecutive numbers representing changes in air pressure. Figure 1.1

The representation provides very little information about the frequencies present in the

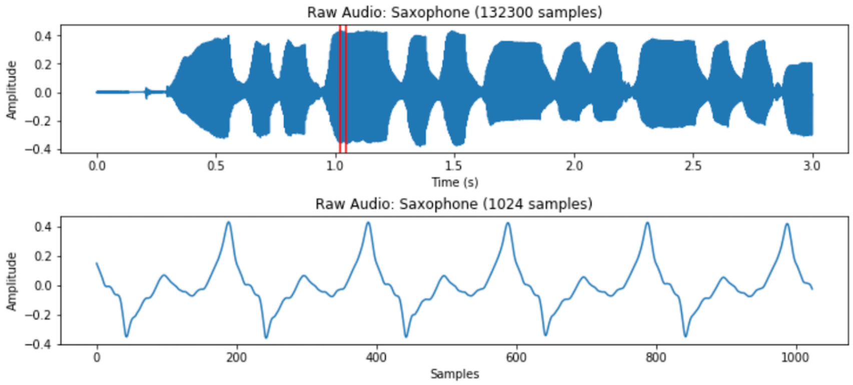


Figure 1.1: Audio Waveform Sample

signal. It can be solved by running FFT (Fast Fourier transform) on small overlapping chunks of the signal. The result can be converted to polar coordinates, providing magnitudes and phases of different frequencies.

Taking an FFT of size 1024 will result in a frequency spectrum with 1024 bins. Due to overlapping of signal, the second half of the spectrum is redundant and hence, (N/2) +1 bins are useful which is 513.Information about whole file can be generated by using FFT of 1024 sample window and sliding it with 512 samples.259 frequency spectrums available to be viewed as 2-dimensional image is generated in this case of three-second file. Example is shown in Figure 1.2

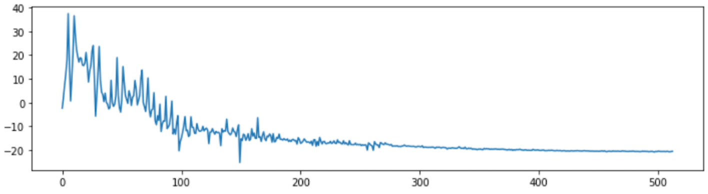


Figure 1.2: Two Dimensional Image

As part of this research we will be using Python library ‘Librosa’ to manipulate audio files. Librosa transforms regular spectrogram into melspectogram, frequency bins to mel scale. This helps us define the number of bins and minimum or maximum frequency that we want for the experiment. This greatly reduces the size of the spectrograms avoiding wastage of bins and enhances the quality of the spectrogram by providing customized frequency scale.

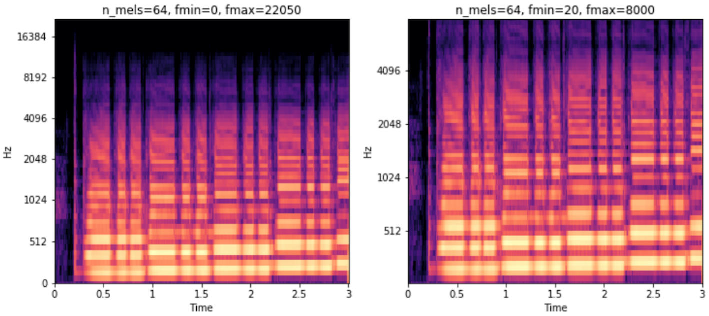


Figure 1.3: Time Frequency representation

Then the T-F representation is treated as an image which is fed into CNNs. However, unlike image classification where the objects are usually centered and occupy a dominant part of the image, audio events may only occur in a short part in an audio recording. To solve this problem, some attention models Ravanelli et al. (2014) for audio classification are applied to attend to the audio events and ignore the irrelevant features.

The focus of this work can be formalized by the research question: “Can precision, accuracy and F1-score for animal sound detection and classification in the audio files be better achieved using deep convolutional neural network model trained on frame per second and spectrogram of the audio as compared to hybrid models built using CNN with traditional machine learning classifier SVM or XGBoost?” Designed experiment can be used to investigate three sub-questions derived as part of the Research question previously stated:

Sub-question A - What mechanism can be used to transform audio file into spectrogram?

Sub-question B - Does the inclusion of audio features as image spectrogram impacts the accuracy of performance of the classifier?

Sub-question C - Which classifier performs best in terms of precision, accuracy and F1-score for classifying animal audio?

1.4 Research Objectives

With the development of multimedia and web technology, video have become ubiquitous on internet. Audios directly reveal the event and help in categorizing and context identification. Using machine learning algorithms to detect and classify the audio, understand the semantics is the most evolving research area.

Though, fully-connected neural network and single attention deep neural network have been used in building audio classifiers in the past, animal audio detection and classification model built on CNN class of deep neural network, trained frame per second and spectrogram of audio from weakly labelled files provides improvement in precision, accuracy and F1-score.

The aim of this work can be outlined from the hypothesis:

Null hypothesis:

Classification and detection of animal audio using deep neural network (CNN) model built on frame per second and spectrogram of audio files does not provide improvement in precision, accuracy and F1-score more than hybrid models CNN-SVM, CNN-XGBoost.

Alternate hypothesis:

Classification and detection of animal audio using deep neural network (CNN) model built on frame per second and spectrogram of audio files provides improvement in precision, accuracy and F1-score more than hybrid models CNN-SVM, CNN-XGBoost.

The principal objective is to conduct experiments that seek to reject the Null hypothesis.

The Research objectives corresponding to the three research sub-question are as follows:

Research objective A - Using python library, Librosa on the audio files to convert into mel-spectogram.

Research objective B - Measure and analyze the changes in performance of the classifier trained with spectrogram.

Research Objective C - Compare the performance of the classifier by measuring the accuracy, precision and F1-score.

Following Experimental steps are performed in conducting the research:

- Document and investigate the state of art in audio classification and the current method used to classify audio.
- Create the dataset for ontology, animal of different sub-class. Extract wavelength audio files from mp3 files if required.
- Convert the wavelength files to images and train the CNN architecture to classify the audio.
- Evaluate the model for its ability in accurately classifying the animal sound.

1.5 Research Methodologies

The research conducted in this project is primary as it involves collecting animal audio data from various online portals and organizing it to be used for training the model. The methodology used are quantitative as the results of the classification from the model will be quantified and evaluated. Research by form is constructive as the work is part of developing a better solution for existing audio classification on the weekly labelled dataset, results for which can be compared with the existing models already designed for audio classifiers.

The work can be classified as an inductive research as it involves building research theory and classification method based on animal audio detected in the files. Similar classification mechanisms can then be applied to generic audio files to detect the animal sound. This research theory can be used in future to design classifier model for other class in all ontology.

The design of this research work is broadly driven by Cross Industry standard process for Data Mining (CRISP-DM) model. Various phases of CRISP-DM cycle are broadly achieved as part of the research work. CRISP DM's first phase of Business Understanding is analogous to Literature Review covered in Chapter 2 of research work. Data understanding and preparation phase is covered in sub-section 3.3 and 3.4, Detailed design and methodology and Data Understanding in Chapter 3. In same chapter, section 3.5 and 3.6 are analogous along with chapter 4 cites details on Data modeling phase of CRISP-DM. Model evaluation and analysis are covered as part of Chapter 5. The end phase of CRISP-DM cycle, Deployment phase corresponds to chapter 6, Discussion and Conclusion.

1.6 Scope and Limitations

The scope of the research work is limited to measuring the changes in performance of animal audio classifier of several state-of-the-art machine learning classifiers using audio dataset. The model is driven by the ideology of identifying the animal sound in the audio files by forcing the network to pay attention to the acoustic details in the audio clip and categorize the content for audio tagging. Additionally, the purpose of this work is to improve the performance of the traditional machine learning classifier.

Though, the objective of this research can best be achieved by using a fully supervised model built on labelled dataset for all the animal ontology class into consideration. Building a dataset with audio examples for all animal class is a time consuming and expensive task. AudioSet, dataset designed by google has labelled audio features extracted from their famous YouTube library. Biggest drawback in utilizing the dataset is computational power required in processing the audio embeddings and filtering it with respect to scope of this project. Preprocessing of the dataset is another limitation as the work is primarily based on building model on mel-spectrograms generated from audio files, converting the audio files captured in different format to wavelength format of 16bit resolution and 44.1KHZ sampling rate is an additional task.

1.7 Document Outline

There are five parts remaining in this report. The following is exhibited a framework of the substance canvassed in every section requested by the chapter number:
Chapter 2 – Review of Existing Literature: This Chapter explores previous research work in animal audio classification, audio event detection, audio feature extraction and machine learning. It discusses the application of machine learning algorithm in multimedia content management and retrieval. As classifying animal audio on similar line of image classification is relatively new and unexplored part of application of machine learning, this chapter provides a comprehensive coverage on most of the work done till date. Additionally, the chapter also highlights and provide brief on strength and gaps in the previous work. The Chapter also points the way forward by summarizing the state-of-the-art techniques in audio classification.
Chapter 3 – Experiment Design and Methodology: This section outlines the details of the project approach regarding design, methodology, experimental set-up, orderly information of work process and data preparing stages. It provides details on all the significant steps taken that structure the premise of this investigation and their precise execution. Specifically, it covers the data collection, description, preprocessing investigation and feature engineering. It likewise calls attention to important information quality issues that can restrain the presentation of machine learning algorithm utilized along these lines. By and large, this section centers around design aspect of the project work and its execution.
Chapter 4 – Implementation and Results: This chapter gives an inside and out explanation of the experiments performed as part of this research work. It centers around the implementation of the model including details on model training, tuning and performance. This chapter briefly also describes the details of the comparative model

built on specification from previous research work. All the more absolutely, the implementation of the machine learning algorithms, comparison between the models and conversion of audio files to images is exhibited in this section.

Chapter 5 – Evaluation and Analysis: This part of the paper covers the performance testing and assessment of the methodologies utilized by analyzing the results of the experiments conducted to classify using different machine learning model trained on same datasets. It reasons that the work done by this research work is able to classify audio as intended and the performance of the classifier can be measured in terms of various performance metrics, accuracy, precision, f1-score, confusion matrix. Average scores were also calculated for precision, accuracy and F1-score to estimate the overall performance of the model. The model giving the best precision is considered and proposed to be applied in future to globally classify sound of all types. The section finishes up with an exchange of qualities and constraints of the Research work.

Chapter 6 – Conclusion: This chapter covers the general accomplishments of the research work and highlights the future work that could be developed later on. The section also gives a conclusion and review the experiment conducted in this research work. The section moreover outlines the recommendations for heading of future work.

Chapter 2

Review of existing literature

Classification of video and audio content has been a challenging task for researches for years as huge number of video and audio are captured and uploaded on multimedia sharing websites like YouTube, Facebook, etc. every day. Work of many researches has yield significant results for managing images, videos with Image recognition by CompuVision. Most research focuses on removing the Image features from the video, with Inage being a prime factor for classifying, categorizing or identifying the picture as a definite solution. The video can for the most part comprise of the three unique kinds of content, it generally consists of the audio, an arrangement of picture frames and the inscribed content. To find the solution in categorizing videos, the researchers consequently have isolated the substance by breaking down any of the three data. Utilizing the accessible picture in the video at 1 picture per frame casing the researchers have prepared the models which could recognize the content over the time length of the video. The major identification is done so that analysts can recognize what is the content of the data provided in the audio or video. In the examination done by the Brezeale and Cook (2008) a basic and clear investigation is directed on the content management by segregating the contents of the videos. Issue in utilizing the effectively structured model on Image for video classification/tagging is the nature of Images. Image resolution can't be guaranteed in every one of the recordings accessible on the Internet and consequently, labeling them just based on Image include is a troublesome undertaking. In this regard, we might need to deal with a multi-class portrayal of our sound or video records by recognizing the classifications of sound events which occur in multimedia file. For example, one may want to tag workout video while being on the 'Park' with kids and pet dog just before the storm. These are different level of annotations present in the video, while being in park during the storm will result in lot of wind, sound from trees, sounds of other kids playing in park, other pet sounds which should be explicitly recognized to understand and tag the video properly. Considering just image per frame will not expressly perceive the surrounding and the weather condition. It is thus necessary to consider the audio as well while designing a model to classify Videos. The goal of this paper is to present a animal audio event detection and classification model which can be used as a generic classifier model in future by training it with audio feature extracted from videos or any audio type.

2.1 Identification of the content using the information present in the picture

A research conducted by Yue-Hei Ng et al. (2015) explains how model can be trained by using data from the video extracted by stabilizing the video content on 1 picture per scale. The researcher also suggests way of increasing the accuracy of the model further by using multiple epochs. The paper also mentions details on identifying the content of the video using simpler models like LSTM with a future work probability of training the model on Convoluted neural network to achieve higher accuracy. Independent feature pooling networks was used in the paper which allows the model to get trained using the picture frame data available in the file. Following this, Yue-Hei Ng et al. (2015) proposes that the output of CNN can be converged with the LSTM network, increasing the accuracy by learning from the output of both the models. Technically the researchers have used virtual assistant Alexa and GoogleNet which are type of CNN.

2.2 Identification of the content using the Multimodal-picture, and text

Another novel approach adopted by the researchYue-Hei Ng et al. (2015) discloses how to train the model using the videos which are 2 minutes long. The model was trained using 120 pictures generated by breaking 2-minute-long video to 1 picture per second. In addition, they took the LSTM output and Feature pooling output which is merged further to increase the precision and accuracy of the model. The accuracy of the model was increased to 73% from the baseline accuracy of 61% proving it to be an efficient model in comparison to previous models designed on same dataset. While this model's accuracy is good, it requires a lot of hardware resources, high CPU usage, high graphical use and is therefore an expensive model to train. Besides being costly if the model has a couple of blurred images, the model wouldn't be able to provide the estimated accuracy.

Lin and Hauptmann (2002) in their research work tries to classify the video which consist of news by using a multi-modality model combining image processing with text processing. Using the SVM, the output was combined and apart from that, the researchers used the probabilistic approach to combine output from both models. Creating a classifier with very high accuracy requires a lot of effort, so the researchers choose a different method and used the combination of different models. This is one of the easiest ways to increase accuracy by integrating the performance of various models and merging them using probability or other approaches to increase the accuracy significantly. It is definitely a better method because it becomes constant after a certain point in order to improve the accuracy of a single classifier and even if the accuracy is decreased it increases at a very low rate.

Performance results from this research show that the classifier built on the image extracted from the video is more reliable and accurate compared to the classifier built on analyzing text, but even better when both classifiers combined the performance of the classifier. Using a multi modalities model reduces the noise and provide better result in noisy environment. This model is equivalently great however the main issue with this model is that it is confined to News which is a little skyline of the down to earth world. The methodology can be widened, and the model can be used by training some real-life scenario.

2.3 Audio event recognition

In this paper Cai et al. (2006), a flexible framework is proposed for key audio effect detection in a continuous audio stream, as well as for the semantic inference of an auditory context. In the proposed framework, key audio effects and the background sounds are comprehensively modeled with hidden Markov models, and a Grammar Network is proposed to connect various models to fully explore the transitions among them. Moreover, a set of new spectral features are employed to improve the representation of each audio effect and the discrimination among various effects. The framework is convenient to add or remove target audio effects in various applications. Based on the obtained key effect sequence, a Bayesian network-based approach is proposed to further discover the high-level semantics of an auditory context by integrating prior knowledge and statistical learning.

On comparative lines, another research by Zhou et al. (2007) interest in detecting and classifying sound in meeting room environment to describe the human and social activity in the meeting room by proposing a new front-end feature analysis and selection framework for Acoustic event detection. The research uses different mechanisms at different stages to attain maximum accuracy in detecting and classifying the audio. The features are characterized by quantifying their relative discriminant capabilities using Kullback-Leibler Distance (KLD). Most discriminant feature set are selected from large feature pool by using Adaboost based algorithm. HMM based framework is used to detect and identify the audio. The experiments show that the discriminant feature set extracted using KLD and Adaboost outperforms the model trained on MFCC features, proving audio feature components having different discriminant capability for audio event detection.

Sound event detection (SED) aims to detect what sound events happen in an audio recording and when they occur. Kong, Xu, Sobieraj, Wang, and Plumbley (2019) the research Propose a time-frequency (T-F) segmentation framework trained on weakly labelled data to tackle the sound event detection. Mixed audio clips of various classes with 10-second recording are trained using 'VGG-like' convolutional neural network with 8 convolutional blocks on the input of mel-spectrogram. F1 score and mAP of 'VGG-like' convolutional neural network and fully connected neural network for Event-wise sound event detection, Frame-wise sound event detection and Time-Frequency segmentation is compared. The T-F segmentation was trained using the T-F segmentation masks which then detects the sound events. Though the experiments show that the global weighted ranking pool (GWRP) outperforms the global max pooling, global average pooling in T-F segmentation and event detection. The only limitation of this approach is T-F segmentation masks doesn't perfectly match the ideal ratio masks (IRP) of sound events. The model is trained for urban sound on similar state of the art approaches as our research work.

Kumar, Dighe, Singh, Chaudhuri, and Raj (2012) introduces unsupervised mechanism for detecting acoustic unit descriptors (AUD) which are based on identifying patterns of occurrences of automatic units of sound. In this work, they examine the AUD patterns within individual audio events. In general, such patterns can be defined as 'grammars', the composition of audio events organized in terms of atomic sound units but using a simplified unigram-based definition that uses only the relative occurrences of AUD as signature of the events. By using a simple discriminative classifier to search for any audio recording for occurrences of AUD, many varieties of the sound can be accurately detected. Another study by Zhuang, Zhou, Huang, and Hasegawa-Johnson (2008) on feature analysis and selection for acoustic event detection suggests quantifying the discriminative potential of each feature variable according to estimated Bayesian accuracy and deriving a discriminative feature set for the detection of acoustic events. It also claims that compared to MFCC, features sets derived from the proposed methods achieve an increase in acoustic event detection of about 30 percent relative accuracy.

The paper Joint Detection and Classification Convolutional Neural Network on Weakly Labelled Bird Audio Detection by Kong, Xu, and Plumbley (2017) works on detecting and classifying bird audio in weekly labelled audio file. The research works on applying VGG like convolutional neural network on the spectrogram as baseline followed by applying joint detection and classification model to detect and classify at same time. In this research, VGG works as a classifier and CNN as a detector. Though the classifier shows high probability in indicating whether a frame contains a bird or not it also has false alarms, that is the detector may attend to non-bird events wrongly. Similar research by Pleva, Vozáriková, Ondáš, Juhár, and Čižmár (2010) focuses on designing audio event classification and detection in urban sounds. The objective of the research was primarily driven in identifying the threats in the audio signal and hence, the work was to create a simple audio event detection system trained on the urban recording and check the used approach based on acoustic information representation based on HMM. The research work was limited in designing a framework that identifies the threats in the signal and ignores the precision parameters.

2.4 Machine Learning methods of classification

A machine learning algorithm, also known as model, is a mathematical expression that represents data, often a business problem, in the scope of a problem. Machine learning algorithms are broadly classified into two general categories of machine learning: Supervised and Unsupervised. When we have a piece of data that we want to forecast or describe, we apply supervised machine learning techniques. We do this by using previous input and output data to predict a new data-based output. Unsupervised Machine learning, on the other hand, looks at way to connect and group data points without using a fixed goal variable. In other word, it examines data in terms of characteristics and uses the characteristics to construct clusters of similar items.

Classification is a supervised learning approach in which the computer program learns from the data input it receives and classifies new observations using this learning. Logistic Regression, Naïve Bayes, Support vector machine, decision tree, random forest are all different models of supervised learning classifier. Brief description about the classifiers used in previous work is mentioned below.

2.4.1 Support vector machine

SVM, a supervised learning algorithm for regression and classification. Yet SVM has been mainly used for classification in recent times. It is a discriminative classifier that operates by separating hyperplane. As such, given the labelled training data (Supervised learning), an optimal hyperplane is generated by the algorithm that categorizes new examples. This hyperplane is a line dividing a plane onto two components in two dimensional spaces where it lies on either side in each section. A margin is a distance between the hyperplane and each class ' support vectors. The data point nearest to the hyperplane is a support vector. The one in which this margin is maximum is a suitable hyperplane. Using SVM in linearly separable datasets is also particularly easy. Since this will be a limitation to use SVM, a method to use SVM's for non-linearly separable datasets was also found by using kernels. The kernels would convert non-linearly separable datasets into linearly separable data and perform the classification. This is achieved by the kernel by adding another dimension so that in a higher dimension a non-linearly separable data can be linearly separable. Upon marking, the modified decision boundaries can be translated back into the original dimensions using mathematical transformations. SMN proves to be a robust classifier to Outliers. Thanks to the volatile nature of the data until transformed by the kernels, SVM is also referred to as the black box. When designing the model, it is very important to carefully decide both C and Gamma parameters.

A research by Lu et al. (2010) proposes a framework based on SVM to address the issue of detecting audio events through audio processing. Throughout our framework when going from start to the end, a sliding window is first used to pre-segment the audio stream into short segments. Throughout our framework when going from start to the end, a sliding window is first used to pre-segment the audio stream into short segments. Support vector machine is then used as a classifier to detect the audio. In this study, Support vector machine is used for audio event detection and the output for performance metrics is compared to other common Gaussian Mixture Model (GMM) classifiers. SVM classifier is used to classify each segment formed by the sliding window into the specified audio class.

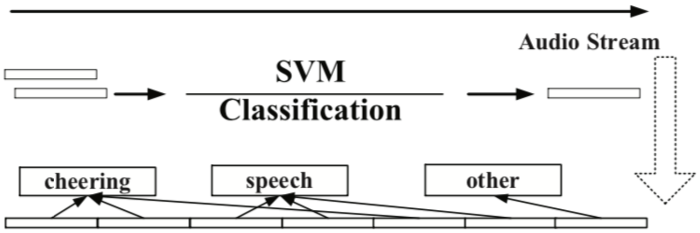


Figure 2.1: SVM Classification

On comparison with GMM based model built by using same training data and same feature set, it was seen that the detail performance of GMM model was much lower than SVM making SVM to be more powerful classifier with an average F value of 79.71% on 8-hour TV dataset. Some of SVM's benefits are that it is highly efficient in memory. When the number of measurements is larger, it also performs fairly well. Because of these benefits, SVM is primarily used for the classification of pictures. Nonetheless, a SVM con is that if the dataset is large, it takes a lot of time to practice. It becomes very important to parameterize the SVM properly when implementing the SVM algorithm. Kernel function and penalty factor are the two most important parameters. When there are very few data points, SVM also suffers from the issue of overfitting. The proper selection of the penalty factor will overcome this problem.

2.4.2 Hidden Marvol

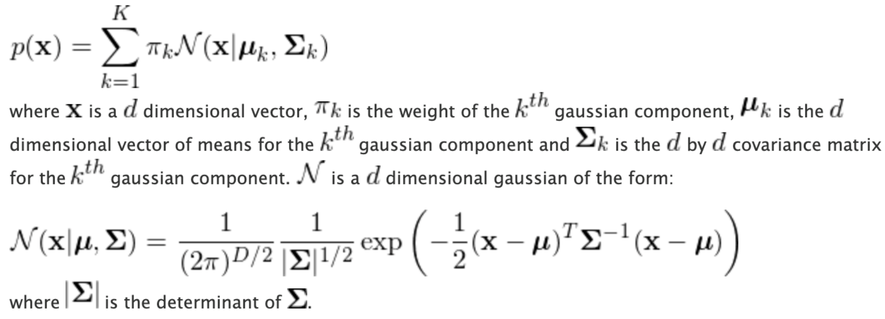
Hidden Markov models (HMMs) are introduced and studied in the early 1970s, named after the Russian mathematician Andrey Andreyevich Markov, who developed a great deal of relevant statistical theory. They were first used in speech recognition and since the late 1980s have been applied successfully to lot of machine learning techniques. Hidden Markov models are probabilistic frameworks that model the observed data as a series of outputs developed by one of several internal states. Inference algorithm are then used by the model to estimate the probability of each state along the observed data.

Boreczky and Wilcox (1998) proposes Hidden Markov model (HMM) for video segmentation. The video is segmented into regions defined by shots, shot boundaries and camera movement within shots. Audio and motion features are used to improve shot boundary detection. Within the HMM framework, the segmentation technique allows features to be combined. For a video labeled with images, transition forms, and motion, the parameters of the HMM are learned using training data in the form of frame-to-frame distances. The state of the Markov hidden model (HMM) has an associated distribution of probability that models the state-conditioned picture, audio, and motion features. Using the image difference function alone, the HMM segmentation algorithm gives higher precision for identical recall values. Additional features make it possible to switch between precision and accuracy in classification.

The paper Zhang, Gatica-Perez, Bengio, and McCowan (2005) addresses the issue of temporal unusual event detection which are characterized by a number of features and propose a framework. The approach for this research is driven by the fact that while obtaining a large training dataset for unusual events is impossible, it is conversely possible for usual events to do so, allowing a well-estimated model for usual events. Bayesian adaption techniques is used to adapt a usual event model to produce a number of unusual event models. Supervised adapted Hidden Markov Model framework was used to run the data and precision metrics was measured for comparison. In future work, we will investigate the use of some criterion for optimizing the number of iterations, as well as improved feature selection.

2.4.3 Gaussian mixture models

Gaussian mixture models are based on unsupervised machine learning algorithm in which the training samples are not labelled for their category membership instead the classifier is trained by estimating the underlying probability density functions of the observations. Majority of the breakthrough in machine learning space is happening through unsupervised learning as it is more flexible in terms of experimenting with the data. GMM have been vastly used in many fields for speech and music recognition. Though GMM is a probability distribution model it comes with an added advantage of modeling distributions with many peaks which is achieved by adding several Gaussian together. A mixture of Gaussian can be written as:



$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $\mathbf{x}$ is a d dimensional vector, π_k is the weight of the k^{th} gaussian component, $\boldsymbol{\mu}_k$ is the d dimensional vector of means for the k^{th} gaussian component and $\boldsymbol{\Sigma}_k$ is the d by d covariance matrix for the k^{th} gaussian component. $\mathcal{N}$ is a d dimensional gaussian of the form:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$.

Figure 2.2: Gaussian Mixture Equation

In the research work, Elizalde et al. (2016) a framework for Acoustic sound classification in real life recordings was proposed and an accuracy of 78.9% was achieved compared to the baseline accuracy of 72.6%. The goal of the work was to classify the audio into predefined classes that categorizes the environment. The goal was achieved in various step, first step was to use an appropriate method to describe the audio segments' acoustic scenes. Bag-of-audio-words-based feature representation is an effective approach to characterizing audio events, typically built on low-level features such as MFCCs. However, acoustic scenes are more complex mixtures of various audio events and it needs a more reliable representation. Training a GMM on MFCC vectors of the training data is the first step in achieving high-level fixed-dimensional feature representation for audio segments. Using the 30ms window and 50% overlap, they extracted 20 dimensional MFCC features. MFCCs are improved with their characteristics of delta and acceleration. They experimented with 4 different values of the GMM variable size M, 64, 128, 256 and 512 for final feature representation. By setting the relevance factor r for to 20 the baseline accuracy was outperformed by an absolute accuracy of 5%.

2.5 Deep Learning classification model

Unlike linear and logistic regressions called linear models, the purpose of neural networks is to capture non-linear patterns in data by adding parameter layers to the model. The neural network structure is robust enough to construct our well-known linear regression and logistics. The word Deep learning originates from a neural network with many hidden layers and encapsulates a wide architecture range.

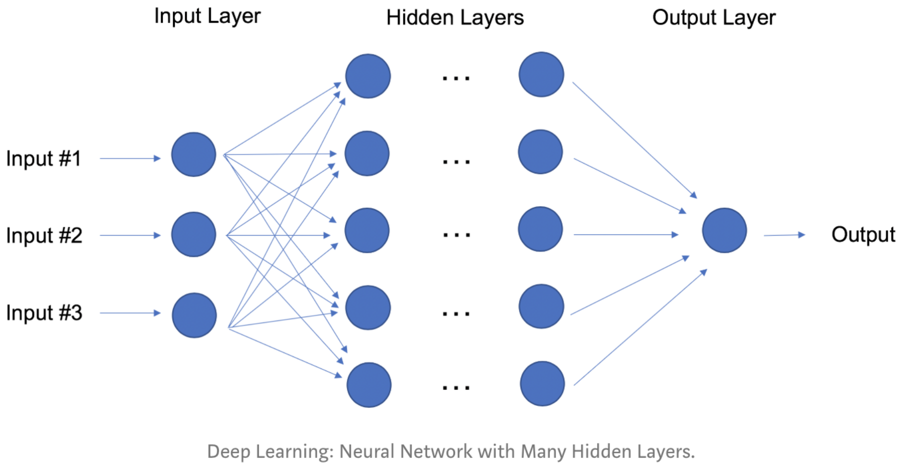


Figure 2.3: Deep Learning Architecture

Deep learning techniques require a lot of data for the best performance and a lot of computational power as the system is self-tuning several parameters within massive architectures. It soon becomes apparent that deep learning practitioners need very powerful computers equipped with GPUs. Deep learning techniques have been extremely successful in the areas of vision (Image classification), text, audio and video.

Deep neural network, well known as Convolutional neural network (CNN), offered means to overcome the limitations of video classification on the basis of just image processing. CNN offers the ability to utilize the audio features from the video and utilize the same to train the model. In the research paper Xu, Kong, Wang, and Plumbley (2018), authors present a gated convolutionary neural network function and a temporary attention- based classification system for audio classification. The proposed framework for classification won 1st prize in the DCASE challenge for large-scale weakly supervised sound event detection task. Two sub-tasks are defined using this weakly labeled data, including audio tagging and sound event detection. A convolutionary recurrent neural network (CRNN) with the non-linearity of learning gated linear units (GLUs) applied to the log Mel spectrogram was proposed along with a temporal attention model along the frames to predict the positions of each audio event in a chunk of weakly labelled data. For the proposed CRNN method initially, the audio waveforms are converted into T-F representations like log mel spectrogram. In order to extract high-level characteristics, convolutionary layers are then added to the T-F representations. A bi-directional recurrent neural network (Bi-RNN) is then adopted to capture the information of the temporal sense, followed by a neural feed-forward network (FNN) to predict the posteriors of each audio class at each frame Xu et al. (2018). Eventually, by comparing the posteriors of all the frames, the expected likelihood of each audio tag is obtained. The conventional ReLU is replaced by gated linear units as activation functions thereby introducing an attention model to all the layers of neural network. The GLU can monitor a T-F unit flow to the next layer of information. If the value of a GLU gate is close to 1, then attend the corresponding T-F cell otherwise when it is 0, the corresponding T-F element will be ignored.

The framework designed by the authors is as above for audio classification. To produce gating outputs and linear outputs, a pair of convolutionary networks are used. Such GLUs can reduce the problem of gradient vanishing for deep networks by providing a linear path for the propagation of gradients while preserving non-linear

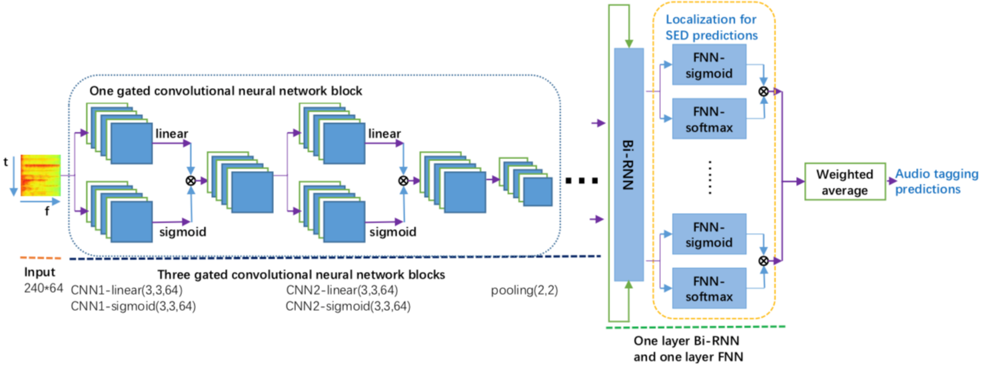


Figure 2.4: Gated CNN Architecture

capabilities via sigmoid operation. A two-level fusion strategy was adopted by the authors to ensure robustness of the model.

As the gradient-based optimization algorithm trains neural networks with a fixed or dynamically changing learning rate, the performance along the epochs will be gradually improved but fluctuating. Therefore, in the same system, our first fusion strategy is carried out among the epochs, improving the stability of the systems. The technique of second fusion is to combine posteriors of different configurations from different systems. An error rate of 0.66 with an F1 score of 55.5% was achieved with a unified method of audio tagging.

2.5.1 Classifications using the Audio content in the Video

The stepping stone for this form of audio classification has been held as early as this study by Lu et al. (2010). In this study, the author explains and demonstrates how to extract the details from the video's audio. The building blocks for research like this generally work on the basis of whether the audio present in the video is packed with elements of speech or no-speech. This high level of distinction is achieved using the famous LSP VQ (Linear Spectral Pair Vector Quantization) and KNN (K Nearest Neighbour) and after differentiating the primary sounds the next step is to understand the sound whether it is background noise or music or any kind of sound.

In addition, another function is to understand whether or not there is a silence and to understand this aspect, the researchers used the zero-crossing rate and short-term energy and evaluated it if it is below the threshold or not, and if it is below the threshold, it is called silence. Now, the remaining part is the non-silence and in order to understand this last important part, the researchers used spectrum J and X (SF) and band periodicity (BP), noise frame ratio (NFR) to differentiate between whether it is music playing or environment sound, and this is how the authors were able to distinguish between noise, ambient sound, music sounds and silence in this study. With this method, the percentage of accuracy was very high, and it reached up to 98% for the segregation of music and voice, as well as the accuracy for all the different classes that are sounds of the world, silence, music, the speech was scored up to 96%. This work is definitely incredibly accurate, but the variety is missing and therefore more classes are needed to run it in the real-world scenario.

The natural sound study was conducted by Aytar, Vondrick, and Torralba (2016), this work is more advanced than previous research, as it took into consideration 2 million unlabeled videos to examine wildlife. They have developed a new sound network they call SoundNet, and in the research world this network has received quite a lot of attention. A CNN model was developed by research Lee, Pham, Largman, and Ng (2009) in which the characteristic is obtained from the audio and convolutionary-restricted-Boltzmann-machines (CRBMs) is used for convolutionary-deep-belief-networks (CDBNs) which is an advanced way for RBM. In addition, PCA is used to cope with the model and spectrogram's higher dimensionality. Also, different music genre audio classification is conducted and built on top of this model, and it has been observed that the accuracy certainly increases after using more classification method and Li and Guo (2000) used SVM to perform the same classification. The authors have conducted and created a new network of 7 analyzes in research done by Takahashi, Gygli, Pfister, and Van Gool (2016), which they called as AENet, and this is used extensively to analyze the audio features.

2.5.2 Summary of Previous work

Audio classification has attracted many attentions in recent years. Various representatives' challenges including DCASE, AudioSet Gemmeke et al. (2017a) have been published with an objective to produce audio event recognizer that can label sounds as humans. Mertens, Lei, Gottlieb, Friedland, and Divakaran (2011b) proposes Hidden Markov model (HMM) for video segmentation. The video is segmented into regions defined by shots, shot boundaries and camera movement within shots. Audio and motion features are used to improve shot boundary detection. Lu et al. (2010) suggested using SVM for audio analysis of general multimedia data and com- pared the performance of the model with Gaussian mixture model (GMM). A sliding window is used first to pre-segment the audio stream into short segments by moving from start to the end followed by extraction of features to represent sound in each segment. Kumar et al. (2012) introduces unsupervised mechanism for automatically discovering the automatic units of sound from unlabeled data known as Acoustic unit descriptors (AUD). Phan, Maaß, Mazur, and Mertins (2014) proposes regression to overcome the problem of classification and segmentation of speech recognition and consider random forest regression by decomposing the audio signals into multiple interleaved super frames, annotated with the corresponding event class labels and their displacement to event temporals. Kong et al. (2017)in his paper suggested joint-detection-classification (JDC) model to detect and classify audio clip. JDC model can attend to informative and ignore uninformative informative reducing the error rate in comparison to bag of frames model. Kong, Xu, et al. (2019) in his paper highlights problem of audio tagging in weekly labelled dataset where the onset and offset annotations of the sound is not present. He proposes a time- frequency segmentation framework (segmentation mapping on log mel spectrogram of audio clip) trained on weakly labelled data to tackle the sound event detection and separation problem.

By reading and reviewing the previous work, it can be seen that a great deal of effort has been made to understand video and audio content and that the researchers have mainly distributed the entire understanding of the content on the basis of three pillars that they call modalities, and the modalities are video, audio and text where video is nothing but the analysis of a series of pictures at different level. Researchers in Agnihotri and Dimitrova (1999) and Lin and Hauptmann (2002) research demonstrated a long-standing technique where they compared the pixel-level contrast of the image and attempted to draw the pixel-level image to understand the artifacts in the picture. One of the finest techniques to identify the objects in the video was this research. Some researchers like Lin and Hauptmann (2002) presented the classification using an SVM-built classifier while others used a different approach and used the HMM to build the classifier. Nevertheless, using any image-based classifier or technique has its drawbacks, such as when it will be dark, the pixels will not be able to detect and when the image is blurred, the pixels will fail to identify the content again, however, when high-efficiency devices such as high-performance GPU and powerful CPU are required to analyze these data. Even, in the case of bad pixels or blurred images, text depending on the layout defines the text by using the image in the video and which is again becoming troublesome.

Another research by Li and Guo (2000) in which the classification work is carried out on the basis of audio. For content-based audio classification and retrieval, a system based on support vector machine is suggested. The SVM minimizes the structural risk, that is, the probability of misclassifying still-to-be-seen patterns for a data distribution of fixed but unknown probability. The experiments conducted involved comparing the SVM and Nearest feature line model with Neural networks as the baseline. A dataset of 409 sounds from Musclefish, classified into 16 classes was used for the experiments. Low error rates were obtained using SVM and Nearest feature line kernel. Another research by Aytar et al. (2016) presents a deep convolutional neural network which learns on audio waveform, trained through the transfer of knowledge from vision to sound. A large amount of unlabeled sound data was trained making it feasible to train the model with overfitting, suggesting deeper models perform better. The synchronous nature of videos (sound + vision) helps us to perform such a transformation that for natural sounds has resulted in semi-rich audio representations. After evaluating all the available methods, it can be found that the accuracy has always been increased whenever there is involvement of more than one classifier and the best way to build a classifier is to increase the accuracy not only by perfecting the same classifier, but also by adding another classifier and by increasing the efficiency with the aid of both classifier.

2.5.3 Gaps

Compared to image data domains, there is relatively little work on applying CNNs to video classification Karpathy et al. (2014). Event detection was completely dependent on audio context recognition Mesaros, Heittola, Eronen, and Virtanen (2010b). Training of the model is done only as per the weak annotations of sound per frame by applying CNN Kumar et al. (2012). Acoustic event detection is done by using CNN by directly modeling on several seconds long signal Takahashi et al. (2016). Joint detection classification model gave an error rate of 17.3% against 16.3% from the baseline model Xu et al. (2018). Takahashi et al. (2016) considered Acoustic event detection (AED) on a dataset with video-level labels as a Multiple instance learning problem, but scaling the approach remains open problem. Deep neural network displays strong power Hassanzadeh and Wang (2016) in audio classification and detection, the networks generally used is feed forward structure without temporal recurrence leading to loss of information. Investigation of accuracy scores for detecting and classifying animal sounds in video files is performed only once employing CNN in contrast with traditional machine learning algorithm such as SVM and Ada Boost.

2.5.4 State of the art to solve the problem

Temko et al. (2006) in his research uses acoustic events in meeting room to detect and describe human and social activity. Mesaros et al. (2010b) used hidden markov models for acoustic event detection in real life environments. Multi-instance learning (MIL) is a supervised learning where each learning example contains a bag of instances from the audio clip. Xu et al. (2018) investigates the classification of audio set dataset and suggested an attention model to tackle the problem of false labelling of instances using MIL. Attention model works by defining probability of each bag, where each instance in the bag has a trainable probability measure for each class. Then the classification of a bag is the expectation of the classification output of the instances in the bag with respect to the learned probability measure. Ravanelli et al. (2014) explores the potential of deep learning in classifying audio on content videos, outperforming the GMM and neural network approach on similar data. Chou et al. (2017) proposes FrameCNN for detection and classification of acoustic scenes and events (DCASE2017). FrameCNN is an extended CNN for classification by adding a branch of network for predicting an acoustic event of each frame on week annotations data. Kumar et al. (2012) proposed deep convolutional neural network-based framework to learn audio event from weakly labelled data. Audio recordings are chunked into small fixed-length segments to train the CNN model. Yu, Barsim, Kong, and Yang (2018) worked on weekly labelled audio set dataset using multi-level attention model as an extension to single-level attention model Kong et al. (2017). It consists of several attention modules applied on neural network layers.

The paper Xiong, Radhakrishnan, Divakaran, and Huang (2005) is driven by the motivation of extracting the visual object using detection-based algorithm to find the semantics and objects in the video in addition to the audio classification algorithm. It is primarily driven by the possibility of building a unified framework to extract highlights from three different kind of sports. To achieve a general framework for different sports, visual markers detection is used along with traditional audio markers to increase the feasibility of detection of the framework. Although, the framework is able to work on three different kind of sports, baseball, golf and soccer as a audio-visual marker detection framework, detecting the visual object has its own challenge in the sports as multiple objects are detected at same time. Also, as the framework is generic it is impossible to categorize noise from various sports in one. Our research work is motivated by Google's idea of having a human like voice recognition system. Google published the dataset AudioSet to resolve the problem of gathering strongly labelled dataset.

Audioset - This paper is based on the publication of AudioSet, a large-scale dataset of manually annotated audio events with an ideology to create audio event recognizer. Audio event recognition is a human like ability to identify sounds from audio. Google constructed the dataset using 10 seconds video from their vast YouTube database. The data was specifically selected using searches based on metadata, context and content analysis to gather data from human labelers probing the presence of specific audio classes. The Audio Set dataset of generic audio events, comprising an ontology of 632 audio event categories and a collection of 1,789,621 labeled 10 sec excerpts from YouTube videos was released with an hope to accelerate Audio event research diaphragm in machine learning world just as ImageNet.

VGG model provided by Google to work on this dataset generates the spectrogram from the audio features in the video in VGG fashion, VGG denotes deep convolutional neural network for object recognition. VGG was a model designed on parallel line for AlexNet having an architecture with an Input of 224*224 pixels RGB Image. VGG's convolutionary layers use a very small receptive range (3x3, the smallest size always capturing left / right and up / down). There are also 1x1 convolution filters that function as a linear input transformation, followed by a ReLU unit. The convolution stage is set to 1 pixel so that after convolution the spatial resolution is retained. All the hidden layers of VGG use ReLU (a huge AlexNet innovation that reduces training time). Local Response Normalization (LRN) is typically not used by VGG as LRN increases memory usage and training time without any unique improvement in accuracy. VGG is a revolutionary platform that supports up to 19 levels of object recognition. Built as a deep CNN, VGG also performs baselines outside of ImageNet on many tasks and datasets. VGG is still one of the most widely used architectures for image recognition. The sample spectrogram generated is as shown in the Figure 2.5.

A state-of-the art approach has been applied in this paper, to create a framework for animal audio classification using the audio files. The main motive of this research is to reduce the computational power of audio processing and making the model accessible

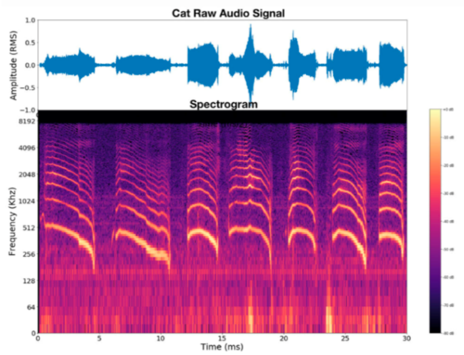


Figure 2.5: Spectrogram

for cheaper machine as well keeping the note of accuracy in mind.

Chapter 3

Experiment Design and Methodology

The aim of this chapter is to provide a summary of the hypotheses capable of answering the research question. Furthermore, this chapter provides an overview on data processing which is a significant challenge as it influences the features to be retrieved, responsible in defining the accuracy of the model. It also gives an overview of the experimental design, methodologies and considerations for the execution of this research. It also includes the statistical treatment of the results of the experiment.

3.1 Project approach

The goal of the current research is to classify animal sound datasets using specialized image patterns (melspectrogram generated from audio features) as defined in Section 3.4 of the current chapter. For the scope of the project two different tasks are performed on different dataset. Task1 is to design deep neural network for animal sound classification on weekly labelled audio.Task 2, is an additional task accomplished to ensure the framework designed by this research work can be extended to use on the video. Weekly labelled Audioset published by Google is used for this task using a pre-trained VGG model and extracting the mel-spectograms from the audio embeddings of the YouTube. As the scope for this project is limited in terms of the availability of the computional power and GPU, simple audio dataset is used to test the framework instead of the AudioSet. The differences, if any, in classification performance using mel-spectograms for different epochs, as measured by Loss, Precision, Accuracy and F1-score of classification, will be analyzed to determine if their impact on the classification performance is statistically significant or not. Also, the scores will be compared with the baseline scores already available for other audio classifiers. Specifically, the aim of the project is to answer the sub-questions defined as part of this research in Chapter 1:

- What mechanism can be used to transform audio file into spectrogram?
- Does the inclusion of audio features as image spectrogram impacts the accuracy of performance of the classifier?
- Which classifier performs best in terms of precision, accuracy and F1-score for classifying animal audio?

3.2 Design aspects

The overall system of our work can be viewed as two entities decomposed into the identification and classification of animal audio. Under the header Detailed Design and Methodology, these two entities are covered in detail in Section 3.3, where each of these entities is treated as a system in their own right and decomposed further.

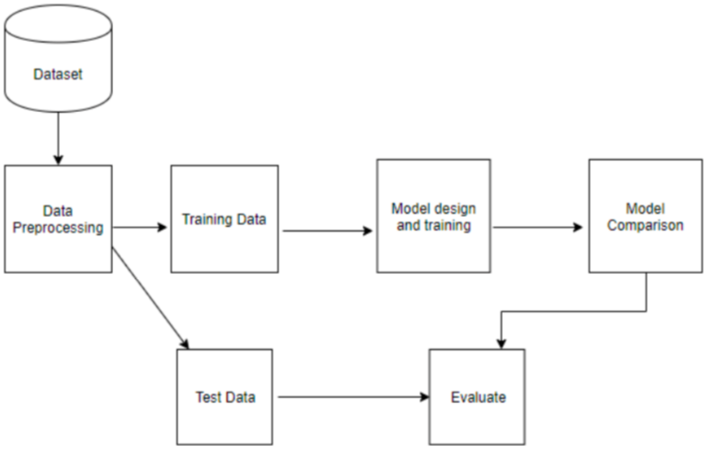


Figure 3.1: Work Flow Diagram

The Figure 3.1 provides a high-level work-flow overview modeling various steps in the research design and the components associated. In summary, the research follows an approach which involves reading the data, generating the spectrogram and analyzing the spectrogram to train the model and further make predictions on the trained model. The experiment was conducted on MacBook Pro (15-inch, 2018) with 2.2 GHz Intel Core i7 processor and 16 GB 2400 MHz DDR4 RAM memory. The graphics on the machine is 4 GB Radeon Pro 555X and 1536 MB Intel UHD Graphics 630.

3.3 Detailed Design and Methodology

Cross-Industry standard process for data mining, well known as CRISP-DM methodology is used in this research work and is described in detail below in each of its phases. CRISP-DM breaks the process of data mining into six major phases listed below in the Figure 3.2 .

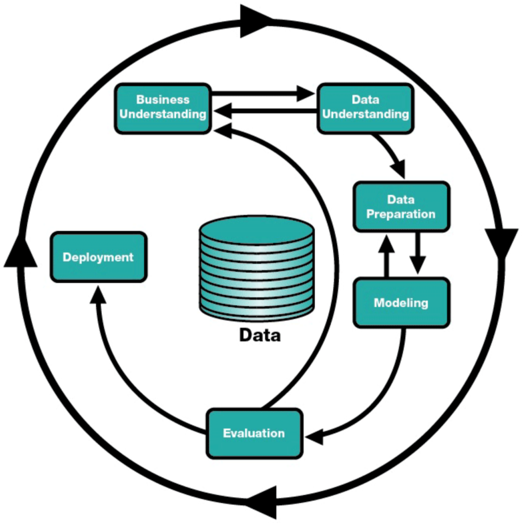


Figure 3.2: CRISP-DM Cycle

As a methodology, it includes explanation of typical project stages, the phases involved in each process, and an overview of the relationships between the phases. The model's life cycle consists of six phases with arrows showing the major and regular interphase dependencies. The sequence of the phases is not strict, it can move back and forth as per necessities of the project.

Business Understanding Focuses on identifying the goals and objectives of the project from a business perspective and then transforming this information into a representation of data mining problems and a draft plan. Chapter 1 and 2 in this research work covers phase 1 of CRISP-DM cycle. This involves understanding research goals and specifications from a business perspective that includes measures such as translating research goals into a specific definition of data mining problems and defining data mining targets and criteria for success.

Data Understanding Begin with an initial data collection and continue with activities to get acquainted with the data, define data quality problems, discover first insights into the data, or detect interesting subsets for hidden knowledge hypotheses. Section 3.4 of the research work elaborates this phase of CRISP-DM and provide reviewing the available data. Nevertheless, the aim of the chapter is to establish a concrete plan to achieve the objectives by outlining a step-by-step project action plan as well as initial review of the tools and techniques.

Data Preparation The phase of data preparation covers all activities from the initial raw data to build the final dataset. The data for this research was collected from various on-line source and is typically not suitable for direct use in analytics or training the model. For the scope of this research work, the data will be pre-processed before training the model. Data will be selected, cleaned, tested, transformed, organized and planned.

Modeling The collection and implementation of modeling techniques. Since some techniques such as neural networks have specific data shape requirements, there may be a loop back to data preparation here. Section 3.5 of this chapter provides an overview on modeling for this research work. Chapter 4 then leads to the design and development of analytical models, Data Modeling phase that involves selecting suitable modeling techniques, configuring and setting parameters for testing, designing experiments, building and evaluating models.

Evaluation When one or more models have been developed that tend to be of high quality based on what loss functions have been chosen, they must be tested to ensure that they are robust against unknown data and that all key business concerns have been examined adequately. End result of the phase is selection of the best model depending on the measurements. Chapter 5 with the header Evaluation and Analysis provides an assessment that includes assessing outcomes, reviewing procedures and observations, identifying any concerns that require immediate attention, or those measures that have been missed or that should be checked, as well as deciding the next steps.

Deployment This will generally mean implementing a model code representation into an operating system to evaluate or categorize new unseen data as it appears and developing a framework for using that new information to address the original business problem. The review, analysis, reporting and the deliverable of the framework of the final results, also known as Deployment.

3.4 Data Understanding

The objective of data understanding is to obtain general insights into the data that may be useful for further steps in the process of data analysis, but data understanding should not be driven entirely by the goals and methods to be implemented in later steps. We know much better at the end of the data understanding process whether the assumptions we made over representativeness, data quality and the presence or absence of external factors are justified during the project understanding phase. The research work entitles and scopes two tasks that will include information in this section. All the sub-section in this chapter will elaborate the details of data processing on both the datasets.

3.4.1 Data Collection

Data collection is the process of collecting and measuring information on interesting variables in an established systematic manner that allows one to answer stated research questions, test hypotheses, and evaluate results. For task 1, the dataset which comprises of the audio from different websites. 4500 animal audio files were collected from various multimedia files. The dataset of 4586 comprises of 47 different class of animal sound. A spreadsheet of the downloaded file was constructed keeping in mind the audio downloaded from the web. Dataset for task 1 thus contains audio files and CSV containing ID and labels of audio files.

For task 2, the dataset that was used for this research is freely and openly available. AudioSet is built was Google using approximately 2 million videos and this dataset is made up of 527 classes. The sound length in this audio is about 10 seconds, making it easier to recognize the algorithm and train the machine. This dataset also contains audio files and CSV file, and CSV file contains the file name and genre or class information. This dataset can be downloaded from the link here: [1]

3.4.2 Data Description

This section describes and elaborates the detail of data used for this research work. The section flows into two parts describing details of each task one by one and also, highlighting the details of count of the audio classes.

Task 1 Data description as part of the data description for the primary task of designing the framework for animal audio classification, involves description of the raw audio data collected from various websites. It involved downloading huge number of audio files and loading the same in python to understand the descriptive analysis. Downloading huge number of audio files for every class of animals was difficult, I merged few classes of animal by creating a family class. For instance, Cattle animals is a grouped class made up by merging sound of cattle, sheep, chicken, lamb, cow. Similarly, Class Birds in our dataset comprises of sounds of all birds namely pigeon, peacock, parrot, hummingbird, owl. Class Canary animals is a unique collection of endemic species from Canary Islands in Spain. Canary Islands continue to host rare and endemic species like Atlantic Canary, Tenerife Blue Chaffinch, Endemic skinks, Canary Islands stonechat, Endemic Geckos, Laurel Pigeons, Big-eared Bat, Bolle's Pigeons. Likewise, class of chirping animal is a class constructed by comprising the

[1] Dataset: `https://research.google.com/audioset/dataset/index.html`

sounds of Rodent, Squirrel, Chipmunk, Groundhog. Class of Chirping animal is the class of sound of cricket, mosquito and other reptiles who chirp. The Figure 3.3 shows the distribution of animal classes present in the underlying dataset.

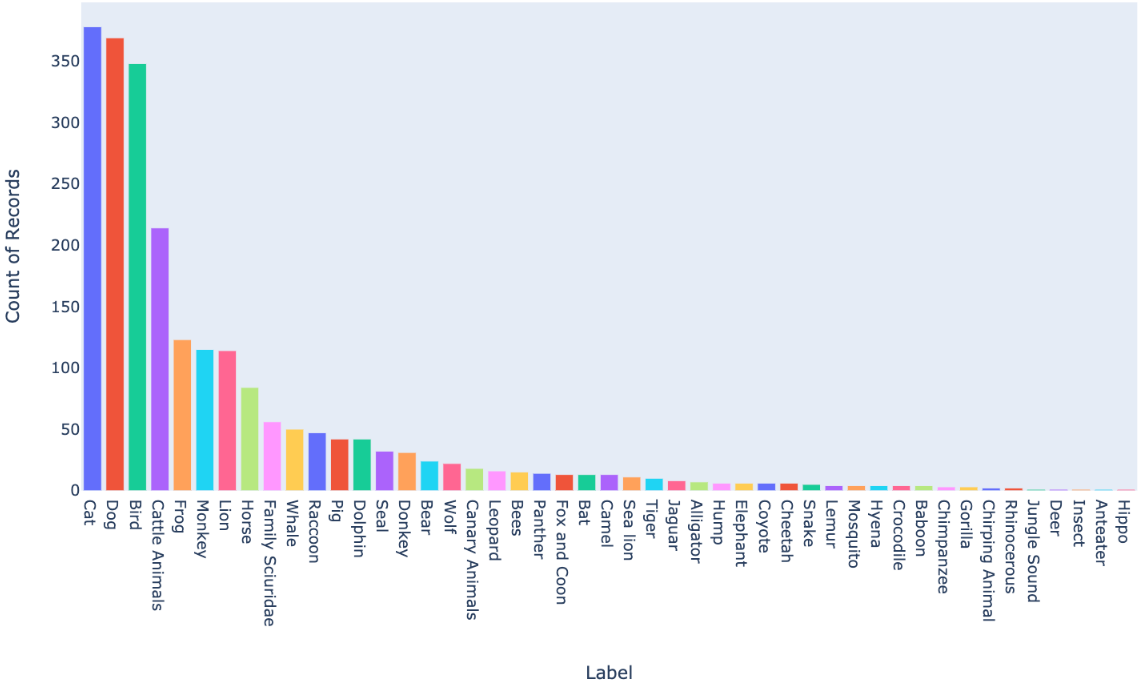
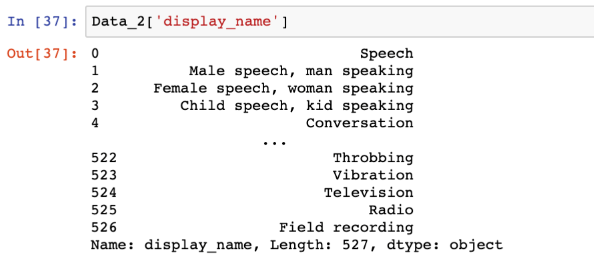


Figure 3.3: Distribution of Classes in Dataset

Task 2 Data Description this part in the section elaborates description of secondary task for the research, extracting audio features from the video and checking the feasibility of the framework designed as part of task 1. The data collected as part of task 2 contains audio files, CSV containing the detail of class labels. The CSV files was loaded in python and descriptive analysis was done on the CSV file to understand the statistics. The CSV file contains audio of 527 different class.

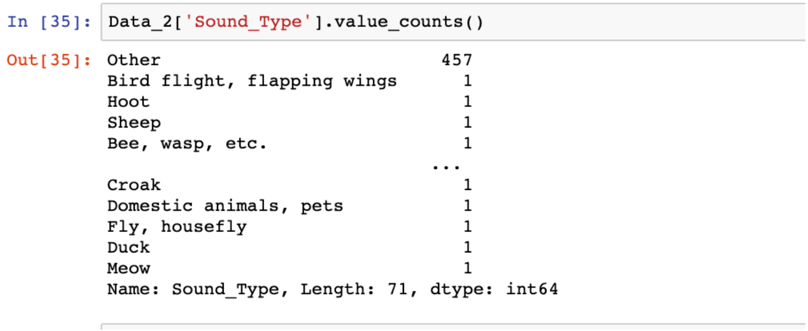
The description of the AudioSet class is as shown in the Figure 3.4 below. As scope of the research is just limited to sound of animals, we will classify all class of no interest as ‘Other’. The description of same is shown in Figure 3.5 below.



```
In [37]: Data_2['display_name']

Out[37]: 0                              Speech
         1          Male speech, man speaking
         2      Female speech, woman speaking
         3         Child speech, kid speaking
         4                        Conversation
                             ...
         522                         Throbbing
         523                         Vibration
         524                        Television
         525                             Radio
         526                   Field recording
         Name: display_name, Length: 527, dtype: object
```

Figure 3.4: Audioset Description



```
In [35]: Data_2['Sound_Type'].value_counts()

Out[35]: Other                           457
         Bird flight, flapping wings       1
         Hoot                              1
         Sheep                             1
         Bee, wasp, etc.                   1
                                        ...
         Croak                             1
         Domestic animals, pets            1
         Fly, housefly                     1
         Duck                              1
         Meow                              1
         Name: Sound_Type, Length: 71, dtype: int64
```

Figure 3.5: Data Description

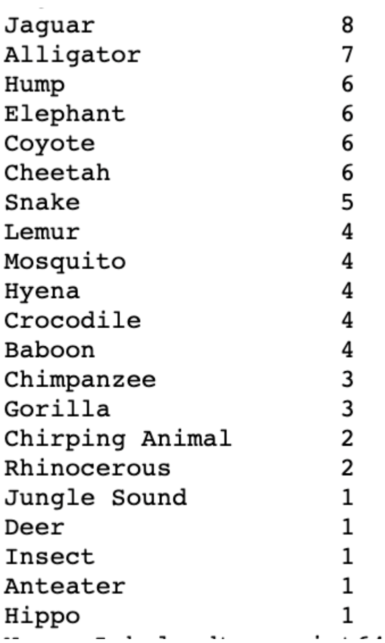
3.4.3 Data Preprocessing

Data preprocessing is a technique of data mining involving the transformation of raw data into a comprehensible format. It is a data mining technique which is used to transform the raw data in a useful and efficient format. As the data is often taken from multiple sources that are normally not too reliable and that too in different formats, when working on a machine learning problem, more than half of the time is consumed in maintaining the data quality.

For Task 1, initial raw data may have diverse issues such as skewness, distortion or outliers that can hamper the performance of the model. Also, as the data is manually downloaded from different free source website initial transformation and filtering was done make sure all the audio files have same format. Many of the websites allow downloading files in just Mp3 format which was then converted to .wav format of 16 bits per sample and sampling rate of 44.1KHz. Online audio convertor mentioned in link [2] was used to convert the files.

[2] Converter: https://audio.online-convert.com/convert-to-wav

As the framework involves training the model through deep neural network having a smaller number of audio files for training will make the framework insufficient with respect to that class. Thus, for the scope of this project we will be dropping all the records having count less than 10. All the records listed below will thus be dropped as part of preprocessing the files.



```
Jaguar              8
Alligator           7
Hump                6
Elephant            6
Coyote              6
Cheetah             6
Snake               5
Lemur               4
Mosquito            4
Hyena               4
Crocodile           4
Baboon              4
Chimpanzee          3
Gorilla             3
Chirping Animal     2
Rhinocerous         2
Jungle Sound        1
Deer                1
Insect              1
Anteater            1
Hippo               1
```

Figure 3.6: Data Label

Length of updated dataset is 4428 after extracting all the files with record count of 10 or less. At further stage, during implementation the data will be replicated to have multiple sound examples for each class.

Further, the audio files were converted to spectrograms using librosa library in python. All the audio files were converted to an mel-spectogram of same size training the model using deep neural networks with tensorflow.

For Task 2, the audio embeddings were downloaded from an authorized Google source and the embeddings are pre-processed in the research paper Kong, Yu, et al. (2019). As the scope of the research is limited to finding if the audio extracted from the videos can be processed using the framework designed, we will be reusing the existing material from the research paper which already explores and analyzes the existing dataset. The

Figure 3.7 shows a bar plot of sound events verses the number of audio clips.

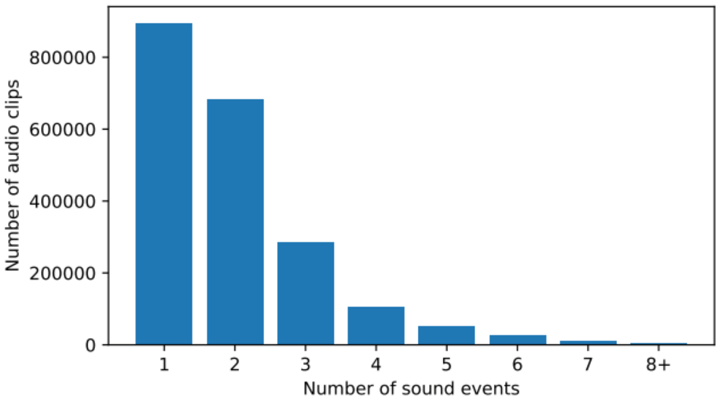


Fig. 4. Distribution of the number of sound classes in an audio clip.

Figure 3.7: Bar Plot Analysis

3.4.4 Data Split

Post preparation phase, the data was split into training and test dataset using the Hold-Out mechanism with 80% and 20% split respectively. The deep CNN model was later tested for on the test and train dataset and model accuracy was predicted accordingly. While training the model the dataset was also split into 75% and 25% of train and validation dataset helping us to predict the Validation loss and accuracy as an additional measurement in designing the model appropriately to avoid underfit or overfit.

3.4.5 Feature Extraction

MelSpectrogram uses a frequency-domain filter bank for time-windowed audio signals. As the second and third output arguments from melSpectrogram, you can get the center frequencies of the filters and the time instants corresponding to the research frames.Mathematically, the mel-scale is the result of some non-linear transformation of the frequency scale. This Mel Scale is constructed such that sounds of equal distance from each other on the Mel Scale, also “sound” to humans as they are equal in distance from one another. In contrast to Hz scale, where the difference between 500 and 1000 Hz is obvious, whereas the difference between 7500 and 8000 Hz is barely noticeable. Python library Librosa has a inbuilt command working on the computation of this non-linear transformation.For the scope of this Research, Mel-Frequency spectrogram was extracted from the audio slices using the librosa library in python which is shown for different classes in next sub-section.

3.4.6 Data Exploration

The sub-section provides details of the dataset used for this research work. Data exploration is the initial step in data analysis, where users explore a wide set of data in an unstructured fashion to discover basic trends, attributes, and points of interest. Visualization is often used by data exploration because it provides a more concise view of data sets than simply analyzing thousands of individual numbers or names.

For audio dataset used as part of this research the data was loaded into python and files were explored to analyze the dataset. The amplitude plots for all the files finally available in the .wav format is as shown in the respective figures which have been attached in the Appendix .1

The framework to be designed needs to be trained on mel-spectrogram, the audio files were converted to spectrogram using the librosa memory. The transformed files had same format for the files which is a pre-requisite to train the model of deep neural network. Though the spectrogram of all the audio files will be different varying as per their amplitude, few of the spectrogram samples are shown below. In Figure 3.8, mel-spectrogram for Bat. Figure 3.9, mel-spectrogram for Bird while Figure 3.10 shows the spectrogram of Lion.

For task 2, the data exploration used in the research paper Kong, Yu, et al. (2019) provides details on the audio files statistics. In Figure 3.11, the blue bars show the number of audio clips of classes. Red stem show the mAP of classes. While Figure 3.12, provides AudioSet statistics, Upper bars: number of audio clips for a particular sound class sorted in descending order plotted in log scale for the sound classes. Red stems:

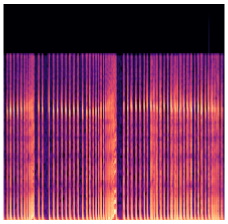


Figure 3.8: Mel-Spectrogram-Bat

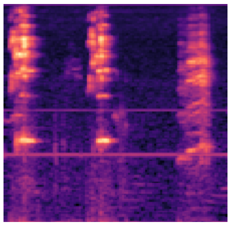


Figure 3.9: Mel-Spectrogram-Bird

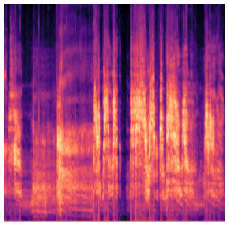


Figure 3.10: Mel-Spectrogram-Lion

average precision (AP) of sound groups with the attention model at the feature-level.

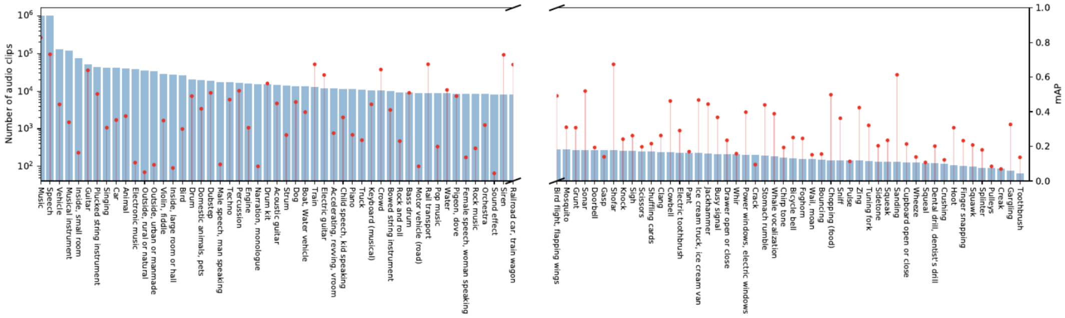


Figure 3.11: Audioset Description

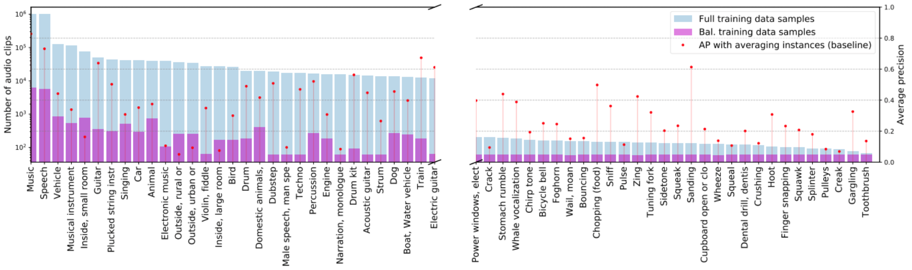


Figure 3.12: Data Description

3.4.7 Modeling

Data modeling is a technique used to identify and evaluate data specifications that are necessary to support business processes within the reach of organizations corresponding information systems. Data Modelers often use multiple models to represent the same data and ensure recognition of all processes, entities, relationships and data flows.

For task 1, the modeling approach for this research is Convolutional neural network combined Dense layers. Three models were developed for the scope of this project; CNN, CNN-SVM and CNN-XGBoost. Conv2D is a 2D convolutional Layer which creates a convolutional kernel that is covered with the input layer to produce a tensor of inputs. Kernel in case of image or mel-spectogram in this work is a convolutional matrix which can be used for edge detection, blurring, sharpening by doing a convolution between a kernel and an image. The sample convolutional architecture is as shown in the Figure 3.13

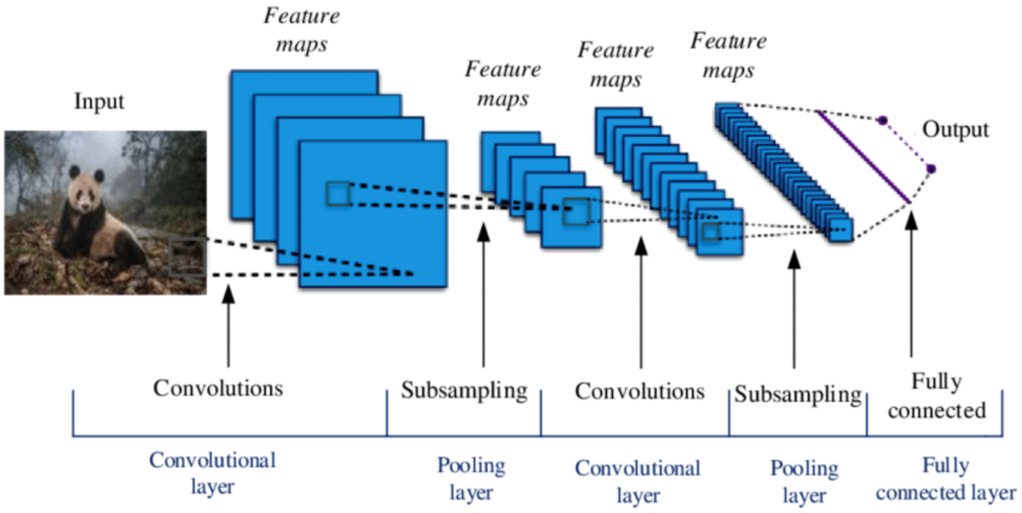


Figure 3.13: Sample Convolutional Architecture

CNNs are made up of neurons with learnable weights and biases, like neural networks. Each neuron receives multiple inputs, takes over them a weighted sum, passes it through an activation function, and responds with an output. The entire network has a loss function and on CNNs all the tips and tricks we built for neural networks are still relevant. The learning process of CNN however is different than neural network. Unlike neural network where the input is a vector, the input to CNN is a multi-channeled image as shown in Figure 3.14 .

Convolution is passing a filter and sliding it over the complete image and along the way of dot product between the filter and the image. For instance, A 5*5*3 filter was considered for 32*32*3 image with every dot product result being scalar. Now the complete image with filter is convoluted over all possible spatial locations as highlighted in the Figure 3.15.

The convolutional layers are main building block for the network, the layer com-

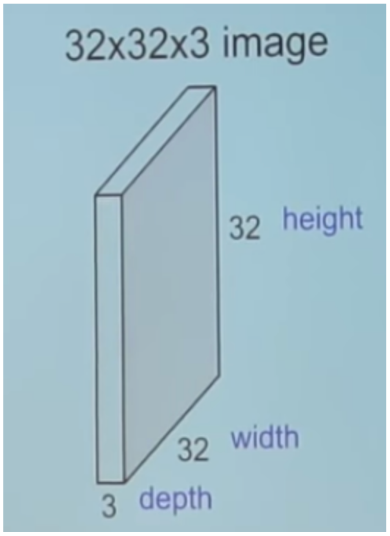


Figure 3.14: CNN Image frame

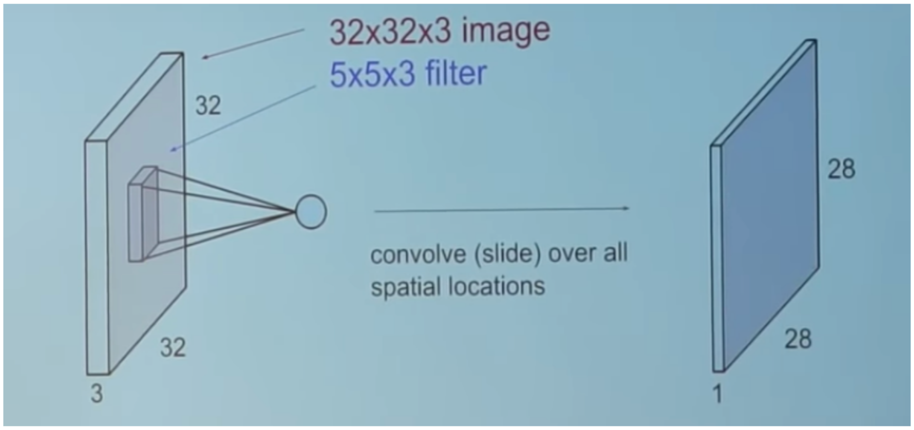
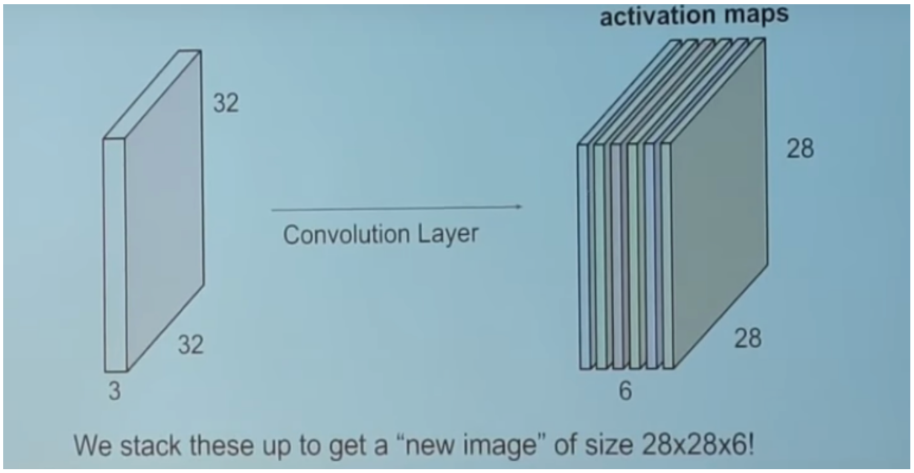


Figure 3.15: CNN Image frame selection flow

prises of ‘X’ set of independent filters which are then convolved with the input image and set of ‘X’ feature map of the input of the images. For instance, in the above example cited in the research work if the activation layers have a set of 6 independent filters, the output of the convolutional layer will be as described in the Figure 3.16 below.



We stack these up to get a “new image” of size 28x28x6!

Figure 3.16: CNN Image process flow

Convolutional neural network has an few additional concepts as parameter sharing, local connectivity, padding and max pooling as compared to neural network. Parameter sharing is sharing weights in a specific function map by all neurons. Local connectivity is the concept of each neural connection to a subset of the input image only. It makes the model more efficient by reducing the computational power. Zero padding helps in preserving the features at the edges by surrounding the matrix with zeroes. Pooling on the other hand is another block which helps in reducing the spatial size of the representation to reduce the number of parameters and computation in the network. A pooling layer is thus the layer which reduces the image dimensionality without losing features or patterns from the input. Activation layer is one of the many similarities between CNN and neural network. Keras have multiple activation function that can be used on the input data post processing by convolutional layer. Some of the most common activation function present in the Keras library are as shown in Figure 3.17. For the scope of this research, activation function ReLU is used which is piecewise linear function that will output the input directly if it is positive, otherwise, it will zero.

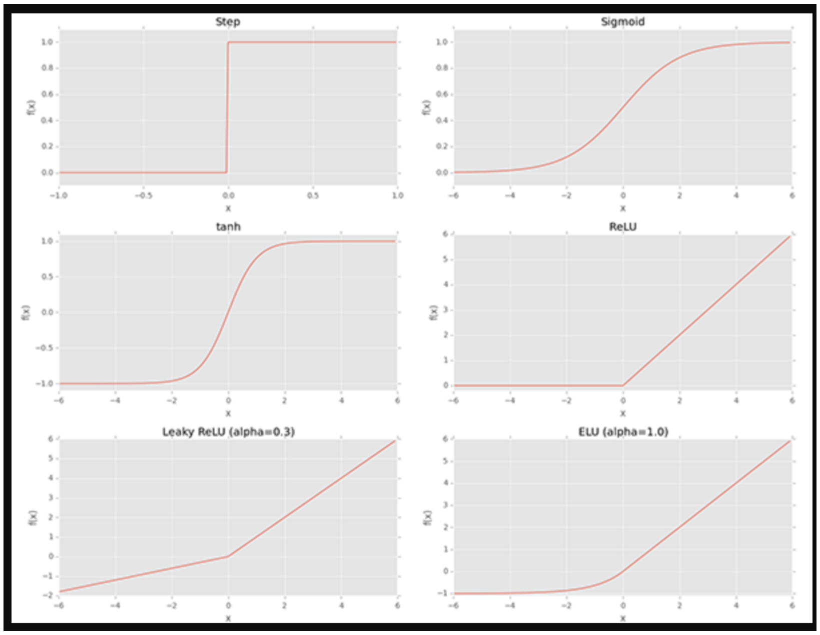


Figure 3.17: Activation Parameter value

The results of the convolutional layers are fed to dense layer, a fully connected network through one or more neural layers to generate the prediction. The output of the convolution layer, a two-dimensional matrix need to be converted to a vector for it to be fed to the dense layer and hence, Flatten layer is used between convolution layer and dense layer. The process of flatten layer is as explained in the Figure 3.18.

A CNN model was designed using all the parameters explained above. The CNN's efficiency was improved by adding more layers to the model and changing the filter and pooling parameters. For the hybrid models designed to compare the accuracy obtained from the CNN model in the first step. At first, a basic CNN model was designed, and the feature extracted which is the output was fed to SVM and XGB models making it a CNN-SVM and CNN-XGB.

For task 2, the most efficient model is using the already built high performing model to extract the features and provides the extracted feature to the framework designed by this research work. The modeling approach can be briefly outlined in two parts:

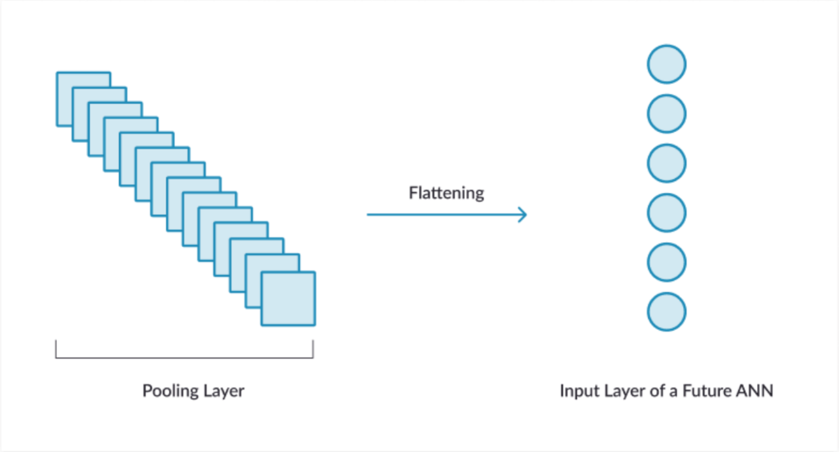


Figure 3.18: Flatten Layer

Transfer learning and working of model.

Transfer learning method implies from the name that it is the transfer of learning and therefore an already trained architecture, popularly known as VGG, in this approach. This model is designed to train the machine learning algorithm and is based on the Deep Machine Learning principle, and this model has many configurable variables that can be used to train the new model. VGG's architecture appears as shown in Figure 3.19

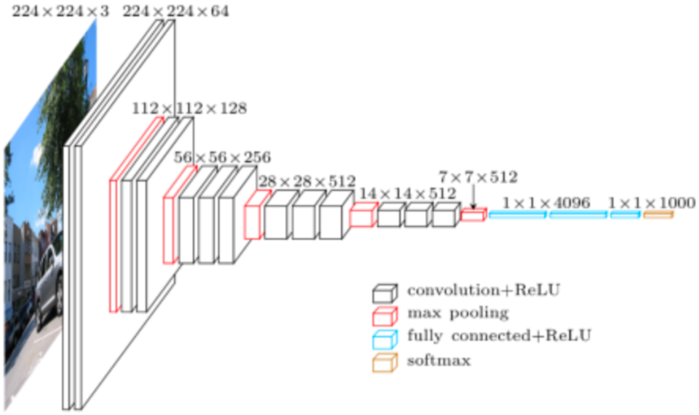


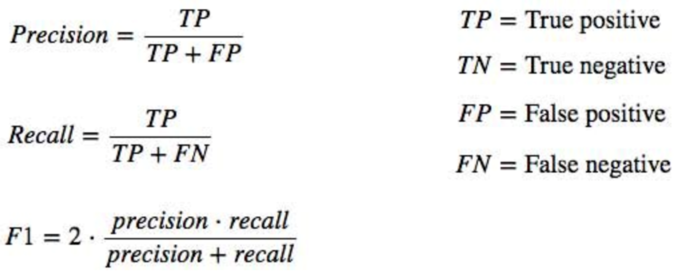
Figure 3.19: VGG-Transfer Learning Process

The primary focus in this research is on audio, as the audio contains information that can be easily understood and interpreted, and the addition of noise also does not affect it much. This procedure consists of a model based entirely on the Convolutions Neural Network's architecture and works only on the spectrograms generated from the audio data. Therefore, this spectrogram is passed through highly sophisticated CNN layers and this pre-trained model would be able to understand the deep features to allocate the audio spectrogram to the relevant class. The spectrogram is produced using the audio data files, each wave is created using the 960 milliseconds of sound and the dimensional attribute generated for this generated spectrogram image is 64 and therefore the entire spectrogram image would be around 96 * 64. This processed image is subsequently fed to the VGG model that has already being trained and the higher dimensionality can be achieved up to 128.

3.4.8 Evaluation

There are certain measures to test machine learning algorithms such as precision,accuracy,F1-score, loss function and confusion matrix. With different metrics, models give different results, so while accuracy may be good for a certain algorithm, another metric, say F1-score, may be very small. Accuracy is usually defined as the number of correctly predicted data points as a percentage of all predictions, but other metrics should be regarded as having a true model judgment.
Precision measures how accurate is the predictions. i.e the percentages of predictions that are correct. Recall on the other hand measures the efficiency of the classifier in classifying all the positives. F1 score is derived from precision and recall, by definition, a harmonic mean between precision and recall where accuracy of the model in predicting is measured by precision and recall checks on the proportion of true positive labels. The mathematical definitions being as listed in the equation Figure 3.20 .

Average statistics score that is micro average, macro average and weighted average can be calculated using the precision, recall and F1-score. Additionally, for this evaluating the framework we will also be looking at confusion matrix and measuring the loss for different epochs to make sure the model doesn't overfit. The terms be rightly defined as follows. Confusion matrix is a table representation to describe the performance of a classification model on set data. Figure 3.21



$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

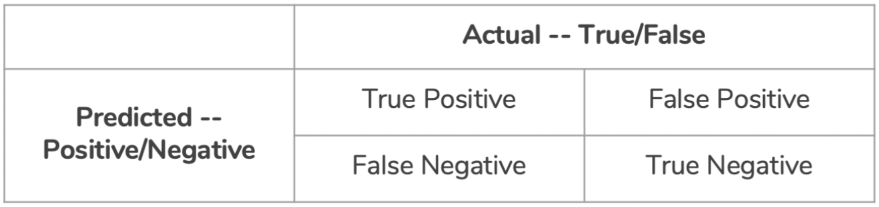
TP = True positive

TN = True negative

FP = False positive

FN = False negative

Figure 3.20: Evaluation Metrics



	Actual -- True/False	
Predicted -- Positive/Negative	True Positive	False Positive
	False Negative	True Negative

Figure 3.21: Confusion Matrix

Loss function for machine learning model being a specific function to evaluate how well specific algorithm models the data. Epoch number is a hyperparameter that defines how many times the learning algorithm is going to work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. Generally, the number of epochs is traditionally large, often hundreds or thousands, allowing the learning algorithm to run until the error from the model has minimized. A comparative analysis will be done between the model to select the best model for the framework thereby accepting for rejecting the null hypothesis primarily designed for this work.

Chapter 4

Implementation and Results

This chapter outlines the execution of this study's experiment followed by a technical assessment and evaluation of the methodology. Previous chapter covers the details on research objective, data understanding and exploration along with modeling and evaluation details for the scope of this research, details of technical implementation for audio classification is explained in this section with result obtained for each task summarized as well.

4.1 Software

The experiment for this research work was done primarily in python using various python libraries. Most frequently used libraries in this research work are Pandas, numpy, Librosa, glob, sklearn. For data exploration plotly.express and extended library for visualizations was used while for data modeling, Conv2D used keras with tensor flow backend. The summary of the results was analyzed and generated using matplotlib.

4.2 Data Modeling

In the research, various models are built on the spectrogram generated from the animal audio and comparison is run between the models. In the first model, Convolutional neural network was applied on the training data and validation data using various parameter and layer changes.The training and validation data for first model was split in 75-25 using holdout split method. For the subsequent model, basic convolutional neural network was applied on the training data and the validation data split 75-25 using random sampling method. The features extracted from the dense layer was than supplied to classical machine learning models like SVM and XGBoost.

4.2.1 Model 1: Convolution neural network

The architecture for the first model is motivated by various famous AI algorithms in which all CNNs has been a key foundation element and is continuously evolving power in the foreseeable future. AlexNet, GoogleNet are all build on the architecture shown in the figure Figure 4.1. The CNN architecture is analogous to that of the connectivity of pattern of neurons in Human being and inspired by organization of Visual Cortex to enable machines to view and perceive the world as humans do.

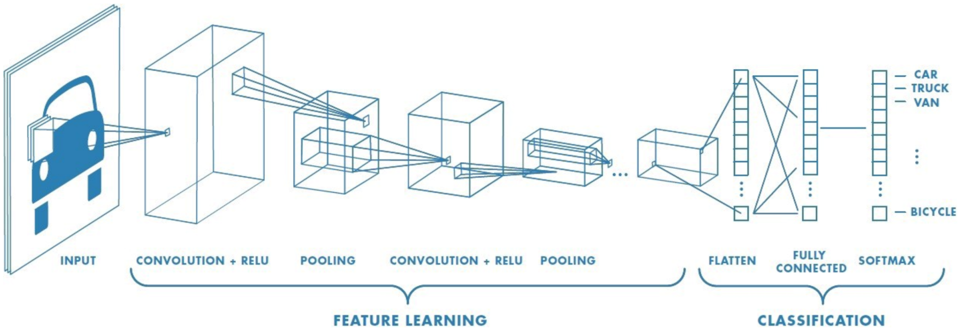


Figure 4.1: CNN Architecture

The architecture of the CNN model designed in the paper is as shown in Figure 4.2 below. It includes several building blocks, such as layers of convolution, pooling layers and layers that are completely connected.

The first layer of the model is Conv2D using 32 filter with a kernel size of 3*3. Kernel size is a 2-tuple specifying the height and width of the convolutional window. Typ-

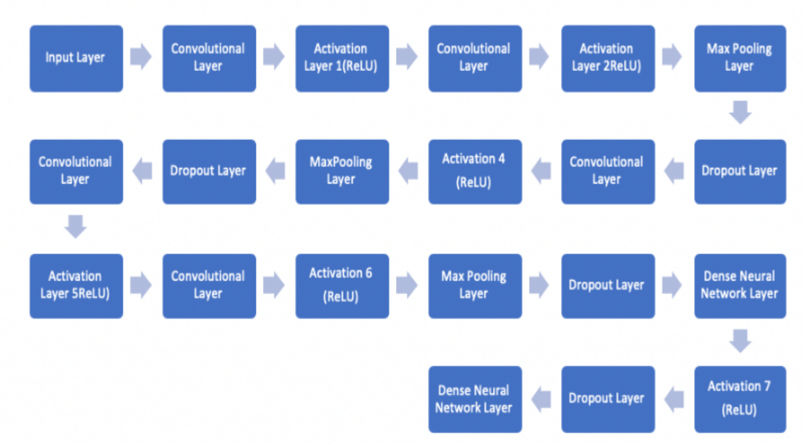


Figure 4.2: Model 1 Architecture

ically, the kernel size includes (1,1), (3,3), (5,5), (7,7) with 7*7 being the largest. It provides an input shape of (64,64,3) for 64*64*3 RGB pictures. The first layer is followed by activation layer which is simply a convenience parameter that allow us to supply a string specifying the name of the activation function that we would like to perform after performing the convolution. ReLU activation is applied on the input layer post convolution. Followed by second convolution layer Conv2D that learns from 64 filters and a kernel size of (3,3). Max pooling size is then used to reduce the spatial dimensions of the output volume. The pooling layer is followed by a dropout layer to reduce the overfitting in the neural network as patented by Google by preventing complex co-adaptations on training data. The next convolutional layer again uses 64 filters and a kernel size of 3*3 but also specifies the padding parameter. The padding parameter in general takes two values; same and valid. If the padding parameter is valid the input value is not zero-padded, and the spatial dimensions are allowed to reduce via the natural process of convolution. In our case, "same" value was passed for padding as we wanted to preserve the spatial dimensions of the input volume so that the output volume size matches the input volume size. The architecture than has two sequence of convolution, activation and pooling layers as feature extractors followed by flatten operation and fully connect dense network to interpret the features and output layer with a ReLU activation for prediction. The model was compiled with RMSprop optimizer with a learning rate of 0.005 with a loss function defined as Categorical_crossentropy as one category is applicable for each data point and the summary of the model is as shown in the Figure 4.3

The data was split into 75-25 for Train and validation, 6879 images belonging to 47 classes of audio was used to train the model. Whereas, 2293 images belonging to 47 classes was used for model validation. The model was fit using the fit generator parameter with steps per epoch equal to size of training batch. The validation steps similarly were equal to the size of validation batch. The images were feed to the final dense layer with a dropout of 0.5. The number of epochs was increased from 1 to 20 for the model. It is seen that there is no change in the validation loss and accuracy after 15 epochs and hence it will be efficient to run the model for a maximum of 15 epoch to avoid overfitting.

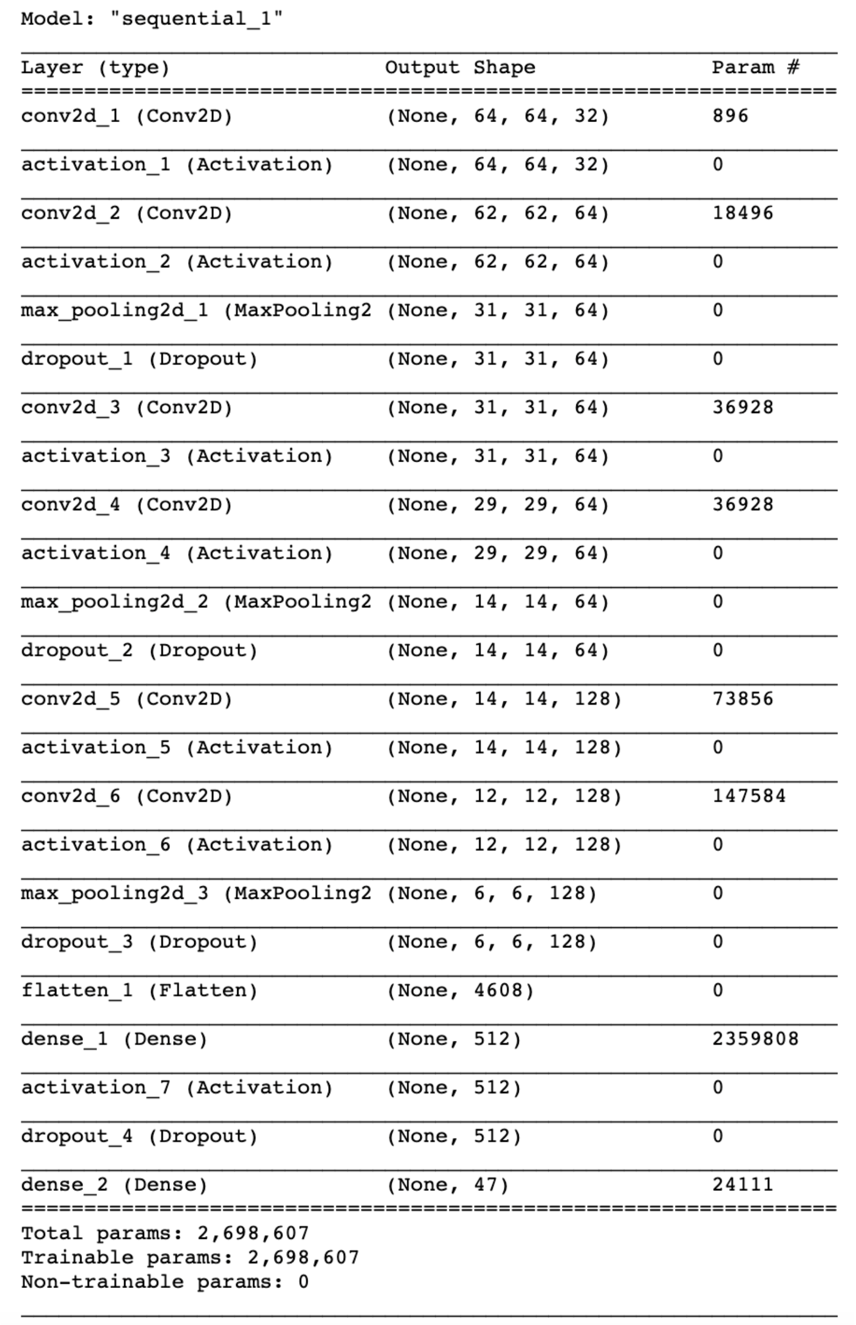
4.2.2 Comparative Models

The second part involves creating an combining architecture of convolutional neural network and SVM or a combination of convolutional neural network and XGBoost. As SVM have been the most important machine learning classifier in the past for classification, many of the researches concentrate on building audio classification models using SVM or XGBoost with CNN. This section provides implementation details of the hybrid model built as part of this research work.

CNN

A basic sequential CNN model was designed with the architecture as shown in Figure 4.4

The first layer of the model is convolutional layer using 24 filters and a kernel size of (5,5). The input shape is specified to be (128,128,1) taking 128*128*1 RGB pictures. Max pooling layer is then used to reduce the spatial dimensions of the output layer with a stride parameter. The stride parameter is a 2-tuple of integers, specifying the step of convolution along the X and Y axis of the input volume. Typically, the



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 32)        896
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 62, 62, 64)        18496
_________________________________________________________________
activation_2 (Activation)    (None, 62, 62, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 31, 31, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 31, 31, 64)        36928
_________________________________________________________________
activation_3 (Activation)    (None, 31, 31, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 29, 29, 64)        36928
_________________________________________________________________
activation_4 (Activation)    (None, 29, 29, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 14, 14, 128)       73856
_________________________________________________________________
activation_5 (Activation)    (None, 14, 14, 128)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 12, 12, 128)       147584
_________________________________________________________________
activation_6 (Activation)    (None, 12, 12, 128)       0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 6, 6, 128)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 6, 6, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4608)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               2359808
_________________________________________________________________
activation_7 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 47)                24111
=================================================================
Total params: 2,698,607
Trainable params: 2,698,607
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.3: Model 1 Summary

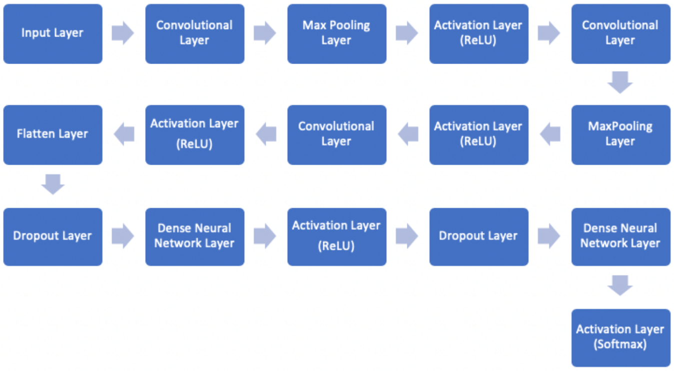


Figure 4.4: Model 2 CNN Architecture

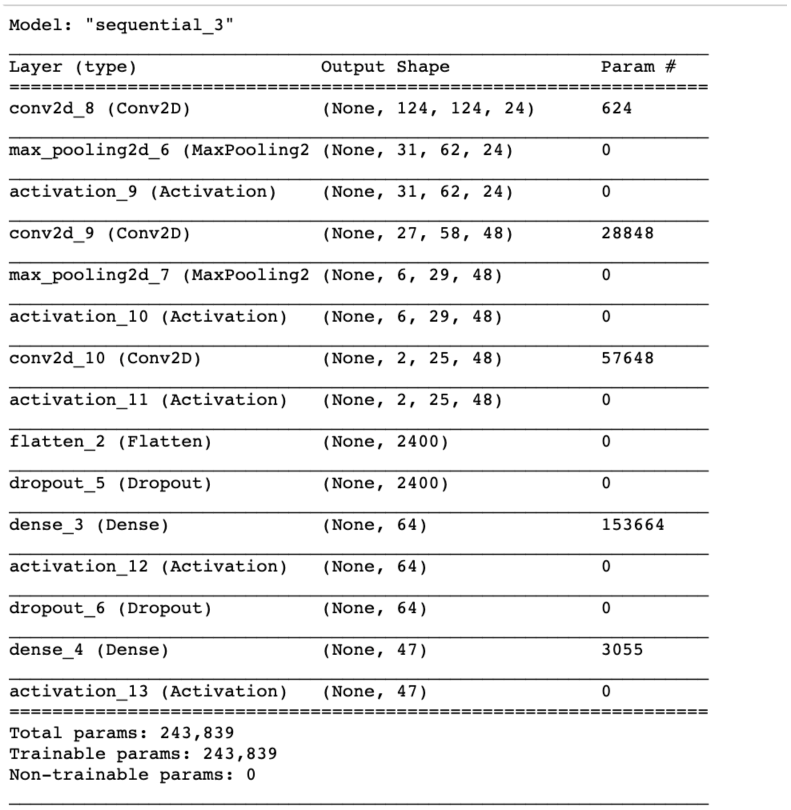
stride value is left to default value which is (1,1) but it is increased to (4,2) in this case to assure the size of the output volume is reduced. The pooling layer is followed by a activation layer of ReLU. Another set of Convolution, Pooling and activation layer with enhanced filter criteria for convolution is used followed by another layer of convolution and activation layer. The output of the activation layer is then fed to Flatten layer and 0.5 dropout layer. Next layer is fully connected dense layer input of which is from the dropout layer. In the end, dense layer and activation layer is used to interpret the features, the input features for the machine learning model is this output. The model was compiled using Adam optimizer and categorical_crossentropy loss. The summary statistics for the model is as shown in the Figure 4.5

The data was randomly split into train and test set of 75-25 and the model was trained on batch size 24 with an epoch ranging till 50. The model shows no improving in loss post epoch 10, giving the best model at epoch =10. It is thus efficient to train the model maximize to can epoch of 10.

CNN-SVM

The feature extracted as output from dense layer of the CNN model designed in 4.2.2.1 is fed to SVM. The architecture of the combined model, CNN with SVM will be as shown in Figure 4.6

.



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 124, 124, 24)      624
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 31, 62, 24)        0
_________________________________________________________________
activation_9 (Activation)    (None, 31, 62, 24)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 27, 58, 48)        28848
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 6, 29, 48)         0
_________________________________________________________________
activation_10 (Activation)   (None, 6, 29, 48)         0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 2, 25, 48)         57648
_________________________________________________________________
activation_11 (Activation)   (None, 2, 25, 48)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 2400)              0
_________________________________________________________________
dropout_5 (Dropout)          (None, 2400)              0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                153664
_________________________________________________________________
activation_12 (Activation)   (None, 64)                0
_________________________________________________________________
dropout_6 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 47)                3055
_________________________________________________________________
activation_13 (Activation)   (None, 47)                0
=================================================================
Total params: 243,839
Trainable params: 243,839
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.5: Model 2 Summary

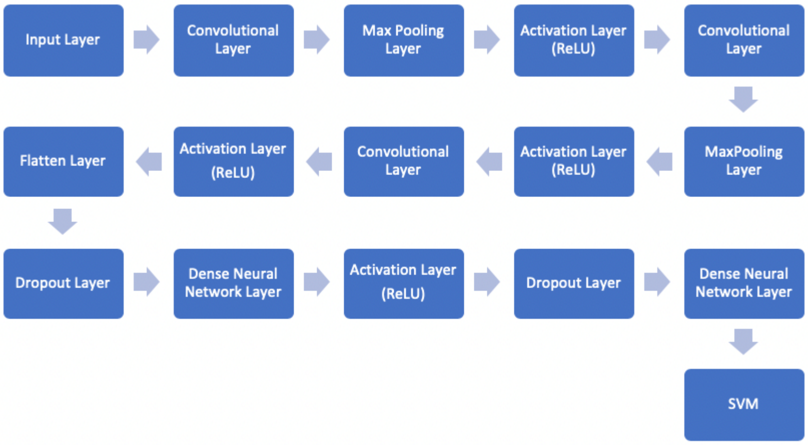
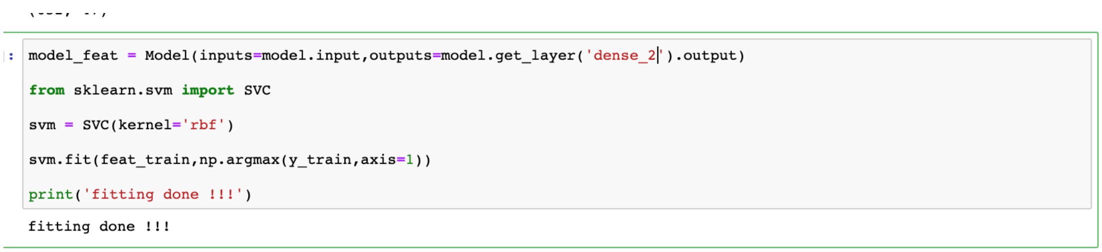


Figure 4.6: Model 2 Architecture

The motivation for this architecture is driven by using CNN for feature extraction and SVM for classification. According to Tang (2013), the classification accuracy can be improved by training a linear SVM classifier on the features extracted by the convolutional base. The Radial basis function kernel is used for SVM. The implementation detail for this classifier is shown in the Figure 4.7



```
model_feat = Model(inputs=model.input,outputs=model.get_layer('dense_2').output)
from sklearn.svm import SVC
svm = SVC(kernel='rbf')
svm.fit(feat_train,np.argmax(y_train,axis=1))
print('fitting done !!!')
fitting done !!!
```

Figure 4.7: Model 3 SVM

The model was training using 7830 images belonging to 47 different classes, validated by 218 images belonging to 47 classes and finally, tested the efficiency using 652 images of 47 different classes. The model produces a accuracy of 92.0% on the training set, 70.0% on the validation set and 68.0% on the test set.

CNN-XGBoost

The feature extracted as output from dense layer of the CNN model designed in 4.2.2.1 is fed to XGBoost. The architecture of the combined model, CNN with XGBoost will be as shown in Figure 4.8

.

The motivation for this architecture is driven by using CNN for feature extraction and SVM for classification. The implementation detail for the model is as shown in Figure 4.9

.

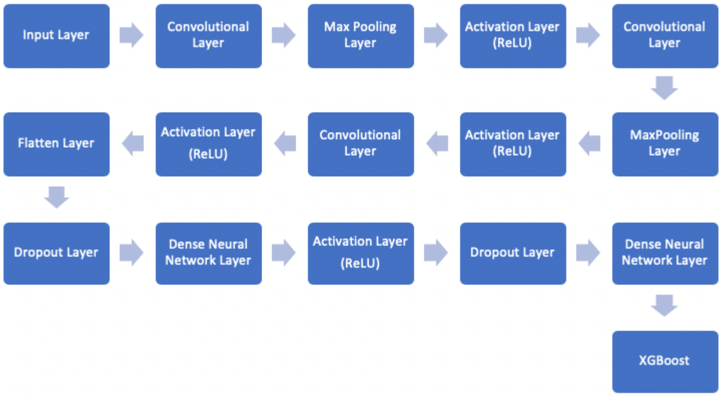
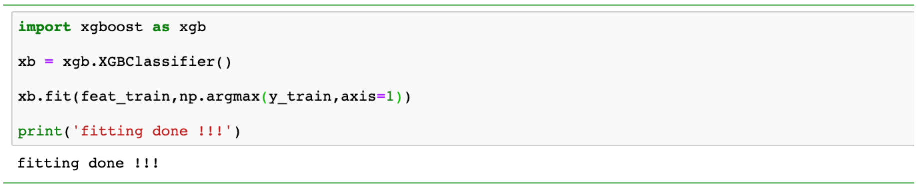


Figure 4.8: Model 3 Architecture



```
import xgboost as xgb

xb = xgb.XGBClassifier()

xb.fit(feat_train,np.argmax(y_train,axis=1))

print('fitting done !!!')

fitting done !!!
```

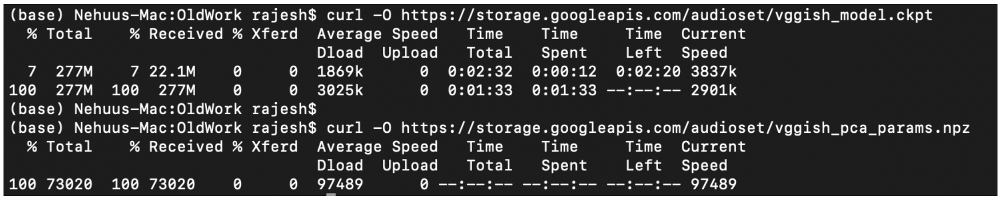
Figure 4.9: Model 3 XGBoost

The number of training, validation and test set for this classifier remains similar to SVM classifier. The model however shows better accuracy with the same set of data compared to the SVM model. The accuracy on the training set is 99.0%, validation set is 73.0% and 72.0% on the test set.

4.3 VGG pre-trained model on AudioSet

4.3.1 Implementation

For implementation of task 2, extracting the audio features and utilizing the VGG pre-trained model patented by Google to understand the flexibility of implementing our framework in future to the extracted features for classification. Details of the audio extraction and classification is mention in the research work, Gemmeke et al. (2017b) and Hershey et al. (2017b). The process follows various steps in implementation and testing. It first involves downloading VGGish model chekpoint available in tensorflow checkpoint format and Embedding PCA parameters available in NumPy compressed archive format.

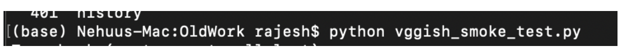


```
(base) Nehuus-Mac:OldWork rajesh$ curl -O https://storage.googleapis.com/audioset/vggish_model.ckpt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  7  277M    7 22.1M    0     0  1869k      0  0:02:32  0:00:12  0:02:20 3837k
100  277M  100  277M    0     0  3025k      0  0:01:33  0:01:33 --:--:-- 2901k
(base) Nehuus-Mac:OldWork rajesh$
(base) Nehuus-Mac:OldWork rajesh$ curl -O https://storage.googleapis.com/audioset/vggish_pca_params.npz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 73020  100 73020    0     0  97489      0 --:--:-- --:--:-- --:--:-- 97489
```

Figure 4.10: Pre Required Files

All the pre-processing python files mentioned in the research paper was downloaded and all the necessary libraries were installed before running the test on VGG model. The layout of the VGG model is designed so as to provide a TensorFlow model that extract input features for the model from the audio waveform and also post-process the model embedding output. The model embedding output is exactly similar to the embedding extract published by Google as AudioSet Dataset. In the architecture few changes were done which is available in vgglish_slim.py and vggish_params.py. The file contains model definition in tensorflow slim notation and hyperparameters. The audio waveform is converted into input example using the vggish_input.py file with the help of mel_features.py file. Vggish_postprocess.py is used for embedding postprocessing. All the python files were downloaded from the link https://github.com/ideo/LaughDetection/tree/master/audioset

vggish_smoke_test.py is the python file to make sure all the required files for VGG is available in the system and processing can be done.



```
  401  history
[(base) Nehuus-Mac:OldWork rajesh$ python vggish_smoke_test.py
```

Figure 4.11: Smoke Test

Model	**CNN**	**CNN-SVM**	**CNN-XGB**
Training Data Accuracy	92.81%	92%	99%
Validation Data Accuracy	97.74%	70%	73%
Testing Data Accuracy	98.9%	68%	72%

Table 4.1: Experiment Result

4.3.2 Summary of Task 2

VGG works as a feature extractor whose output can be transformed for other classifier for classification. VGGish model transforms audio features into a high-level, semantically meaning 128-D embedding that can be fed to a downstream classification model as an input. In a large model architecture, it adds more layers on the top of the input and enhances the input which is a prime component when the audio is extracted from the video and may contain noise.

4.4 Results

At the beginning of this section, it outlined the task and re-examined the dataset for the implementation. It provides all the details of the software used, the steps required to extract audio embeddings from the video to train using the framework designed as part of this research and also details on the model implementation done as part of research work. The section provides a brief explanation on all the architecture designed in this research work like CNN, CNN-SVM and CNN-XGBoost. Finally, the result of all the models can be compared to understand and define the best model. The Table 4.1 show details of performance of all the models used in the research.

It is seen that the CNN designed with hyperparameters outperforms the hybrid models. Though using traditional classification method with a basic CNN definitely increases the accuracy of the models as suggested by Tang (2013) in which the author provides the output from the CNN to SVM classifier. When compared between the hybrid model which has lot of interest by many researchers given the option to play more with various features and use the very familiar traditional classifier, we see performance of CNN with XGBoost classifier exceeds the performance of CNN-SVM classifier. It should also be noted that highest training efficiency amongst all the model is achieved by CNN-XGBoost model. Detail analysis and evaluation on the performance of the model is explained in detail in chapter 5 of the research work under the header “evaluation and analysis”. The comparison of the model in line with the literature review and novelty of the work will be discussed in detail in chapter 6 using the data collected during implementation.

Chapter 5

Evaluation and Analysis

This chapter summarizes and discusses the results obtained in the context of the research question and hypothesis done in chapter 4. Chapter outlines the inferences from the results that are generated and describes the performance of classifiers in detail. Evaluation of the hypothesis with respect to scope of the research is also highlighted and the chapter is concluded by conferring the strength and weaknesses of the framework designed.

5.1 Introduction

Three models were studied as part of this research and the performance of each of the model was measured by using various evaluation metrics. The model is evaluated in detail in this section using error rate, classification reports and difference between actual class and the predicted class. The model is basically evaluated on two aspects; one is how the model responds to the number of runs and how many runs it will take in stabilizing the process with precision. The second aspect in evaluating the model is how the time and processing power needed to train the model increases or decreases in training and testing dataset with noise induction.

5.2 Evaluation of the result

Convolutional neural network, the first model was trained on 6879 images and validated on 2293 images. The model was trained with 10 epochs after which there seems to be no improvement in accuracy or any changes in loss and hence, training the model for 10 epochs should be efficient to attain best performance in terms of both accuracy and computational power. The figure Figure 5.1 represents the performance of CNN for 10 epochs.

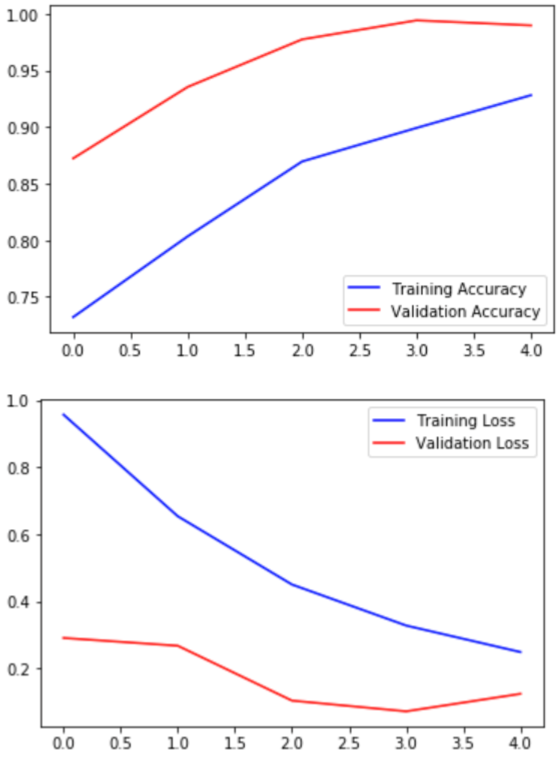
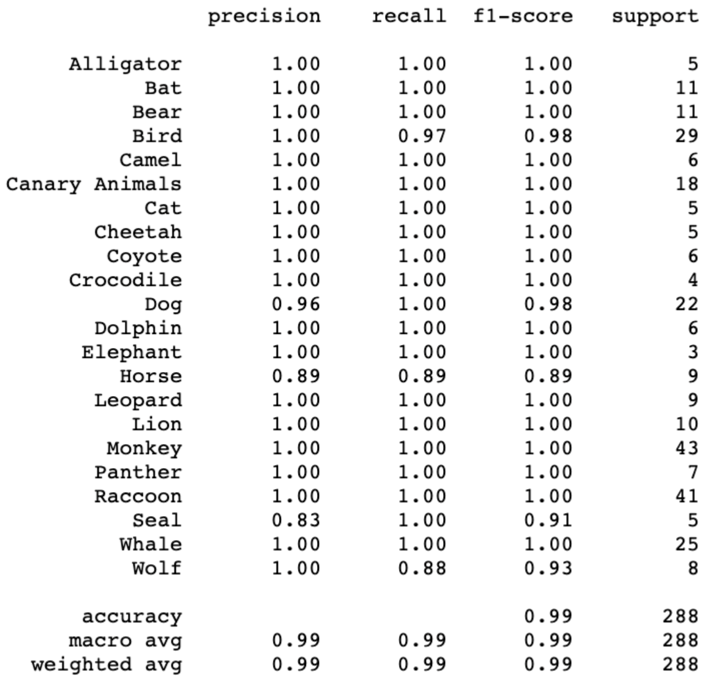


Figure 5.1: Model 1 Performance

.

The model was tested on 290 images belonging to 22 different class. An accuracy of 98.95% was achieved by the model on test dataset. The classification report of the model for the test dataset is as shown in the figure Figure 5.2.

A basic CNN model was built for combined model to be built on top. The CNN is used as a feature extractor, the classification is done by SVM and XGBoost in model



```
                precision    recall  f1-score   support

     Alligator       1.00      1.00      1.00         5
           Bat       1.00      1.00      1.00        11
          Bear       1.00      1.00      1.00        11
          Bird       1.00      0.97      0.98        29
         Camel       1.00      1.00      1.00         6
Canary Animals       1.00      1.00      1.00        18
           Cat       1.00      1.00      1.00         5
       Cheetah       1.00      1.00      1.00         5
        Coyote       1.00      1.00      1.00         6
     Crocodile       1.00      1.00      1.00         4
           Dog       0.96      1.00      0.98        22
       Dolphin       1.00      1.00      1.00         6
      Elephant       1.00      1.00      1.00         3
         Horse       0.89      0.89      0.89         9
       Leopard       1.00      1.00      1.00         9
          Lion       1.00      1.00      1.00        10
        Monkey       1.00      1.00      1.00        43
       Panther       1.00      1.00      1.00         7
       Raccoon       1.00      1.00      1.00        41
          Seal       0.83      1.00      0.91         5
         Whale       1.00      1.00      1.00        25
          Wolf       1.00      0.88      0.93         8

      accuracy                           0.99       288
     macro avg       0.99      0.99      0.99       288
  weighted avg       0.99      0.99      0.99       288
```

Figure 5.2: Model 1 Classification Report

2 and 3 respectively. The model was initially trained on epoch 1 to 50 but it was observed there was no improvement in accuracy or declination in loss post epoch 17 and the best model was recorded at epoch 17. The figure Figure 5.3 represents the performance of the classifier for basic CNN model designed.

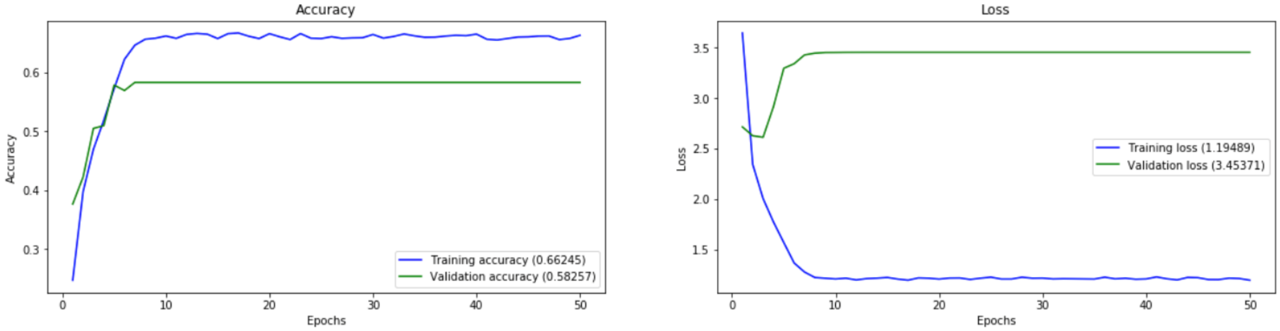


Figure 5.3: Model 2 CNN Performance

The model is then trained with 7830 images, validated on 218 images and tested on 652 images. The model's accuracy as measured to check if there is any improvement

	Train	**Validate**	**Test**
Matched	6024	127	393
Missed	1806	91	259
Total	7830	218	652
Accuracy	77%	58%	60.2%

Table 5.1: SVM as a classifier using the output of CNN

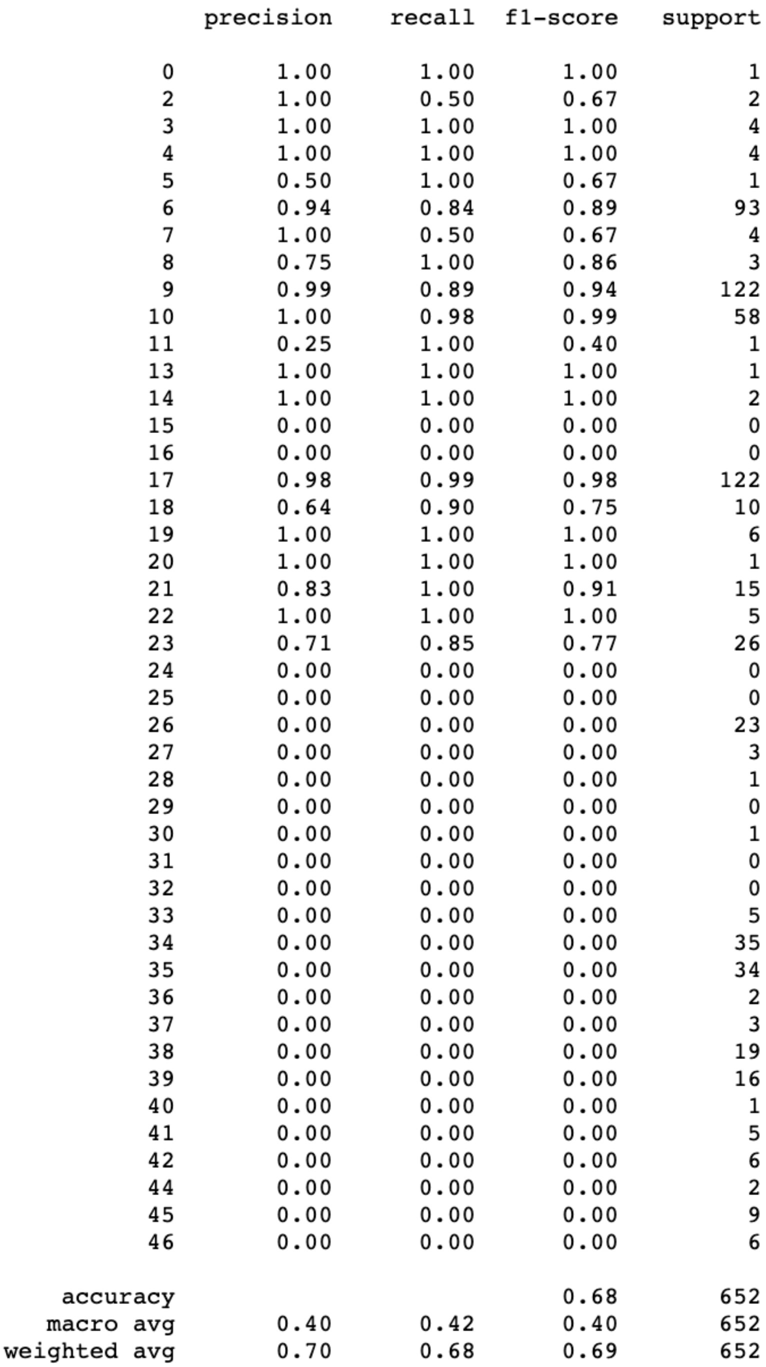
	Train	**Validate**	**Test**
Matched	7235	152	443
Missed	595	66	209
Total	7830	218	652
Accuracy	91.78%	69.72%	67.99%

Table 5.2: CNN-SVM Evaluation

in the accuracy when the classification will be done using SVM and XGBoost. The table 5.1 shows details of the evaluation done by CNN classifier in first stage.

SVM was used as a classifier with the output of the CNN. The SVM model was trained with 7830 images belonging to 47 different class, validated using 218 images belonging to 47 class and tested on 652 images of 47 class. The performance of the SVM model is as shown in the table 5.2.

The classification report for CNN-SVM on training model is shown in the figure .1. It shows the precision, recall and F1-support for all 47 different class. It is seen that the F1 score, recall and precision with training set is almost equivalent to 1. The classification report on validation report however shows lesser accuracy and also the precision, recall, f1-score is 0 for many classes. .1 In case of test data on the model, the classification report shows very few classes with a perfect f1 score, precision and recall. The report is shown in Figure 5.4.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           2       1.00      0.50      0.67         2
           3       1.00      1.00      1.00         4
           4       1.00      1.00      1.00         4
           5       0.50      1.00      0.67         1
           6       0.94      0.84      0.89        93
           7       1.00      0.50      0.67         4
           8       0.75      1.00      0.86         3
           9       0.99      0.89      0.94       122
          10       1.00      0.98      0.99        58
          11       0.25      1.00      0.40         1
          13       1.00      1.00      1.00         1
          14       1.00      1.00      1.00         2
          15       0.00      0.00      0.00         0
          16       0.00      0.00      0.00         0
          17       0.98      0.99      0.98       122
          18       0.64      0.90      0.75        10
          19       1.00      1.00      1.00         6
          20       1.00      1.00      1.00         1
          21       0.83      1.00      0.91        15
          22       1.00      1.00      1.00         5
          23       0.71      0.85      0.77        26
          24       0.00      0.00      0.00         0
          25       0.00      0.00      0.00         0
          26       0.00      0.00      0.00        23
          27       0.00      0.00      0.00         3
          28       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         0
          30       0.00      0.00      0.00         1
          31       0.00      0.00      0.00         0
          32       0.00      0.00      0.00         0
          33       0.00      0.00      0.00         5
          34       0.00      0.00      0.00        35
          35       0.00      0.00      0.00        34
          36       0.00      0.00      0.00         2
          37       0.00      0.00      0.00         3
          38       0.00      0.00      0.00        19
          39       0.00      0.00      0.00        16
          40       0.00      0.00      0.00         1
          41       0.00      0.00      0.00         5
          42       0.00      0.00      0.00         6
          44       0.00      0.00      0.00         2
          45       0.00      0.00      0.00         9
          46       0.00      0.00      0.00         6

    accuracy                           0.68       652
   macro avg       0.40      0.42      0.40       652
weighted avg       0.70      0.68      0.69       652
```

Figure 5.4: Model 2 Classification Report

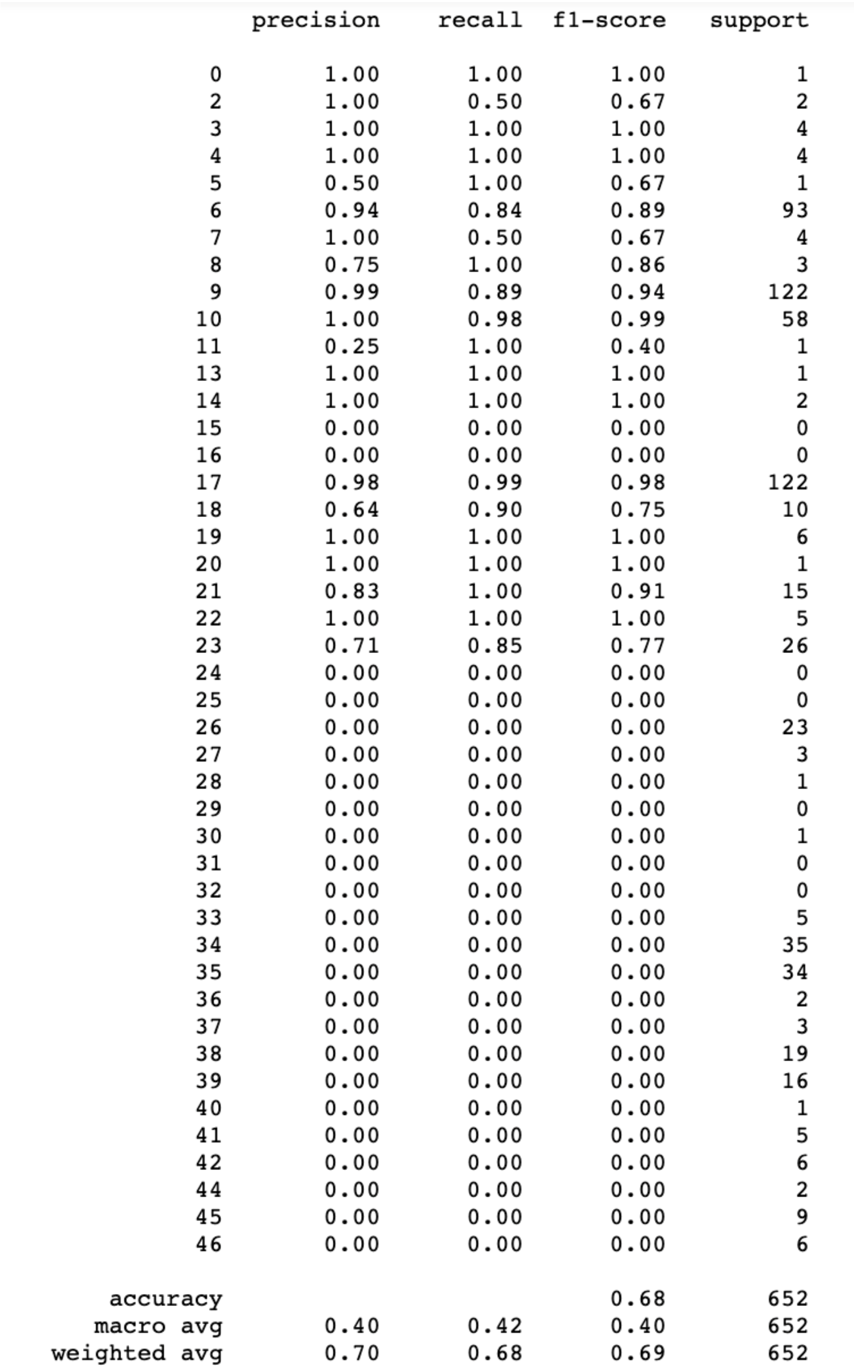
	Train	**Validate**	**Test**
Matched	7782	152	443
Missed	42	66	209
Total	7824	218	652
Accuracy	99%	73%	67.99%

Table 5.3: Model 3 Evaluation

The SVM classifier improves the accuracy of the basic CNN model but the accuracy and number of images classified correctly is less than Model 1. Details of the measurement noted will be compared and the best efficient model will be decided further in this section.
Another model was built was supplying using XGBoost classifier on the output of CNN. The performance of the model is presented in the tabular form in table 5.3. The model will be called as Model3 for the scope of this research work. In all, 7824 images were trained using CNN-SVM model while the model was validated using 218 images and tested on 652 images.

The classification report of the training, validation and test dataset is as shown in Figure 27, Figure 28, Figure 5.5.

The classification report shows that the model during training hardly misses one class for precision, recall and F1 score. The accuracy of the model is also in line with the same. The model however misses perfect precision, recall and F1 score for many classes in the validation and test dataset.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           2       1.00      0.50      0.67         2
           3       1.00      1.00      1.00         4
           4       1.00      1.00      1.00         4
           5       0.50      1.00      0.67         1
           6       0.94      0.84      0.89        93
           7       1.00      0.50      0.67         4
           8       0.75      1.00      0.86         3
           9       0.99      0.89      0.94       122
          10       1.00      0.98      0.99        58
          11       0.25      1.00      0.40         1
          13       1.00      1.00      1.00         1
          14       1.00      1.00      1.00         2
          15       0.00      0.00      0.00         0
          16       0.00      0.00      0.00         0
          17       0.98      0.99      0.98       122
          18       0.64      0.90      0.75        10
          19       1.00      1.00      1.00         6
          20       1.00      1.00      1.00         1
          21       0.83      1.00      0.91        15
          22       1.00      1.00      1.00         5
          23       0.71      0.85      0.77        26
          24       0.00      0.00      0.00         0
          25       0.00      0.00      0.00         0
          26       0.00      0.00      0.00        23
          27       0.00      0.00      0.00         3
          28       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         0
          30       0.00      0.00      0.00         1
          31       0.00      0.00      0.00         0
          32       0.00      0.00      0.00         0
          33       0.00      0.00      0.00         5
          34       0.00      0.00      0.00        35
          35       0.00      0.00      0.00        34
          36       0.00      0.00      0.00         2
          37       0.00      0.00      0.00         3
          38       0.00      0.00      0.00        19
          39       0.00      0.00      0.00        16
          40       0.00      0.00      0.00         1
          41       0.00      0.00      0.00         5
          42       0.00      0.00      0.00         6
          44       0.00      0.00      0.00         2
          45       0.00      0.00      0.00         9
          46       0.00      0.00      0.00         6

    accuracy                           0.68       652
   macro avg       0.40      0.42      0.40       652
weighted avg       0.70      0.68      0.69       652
```

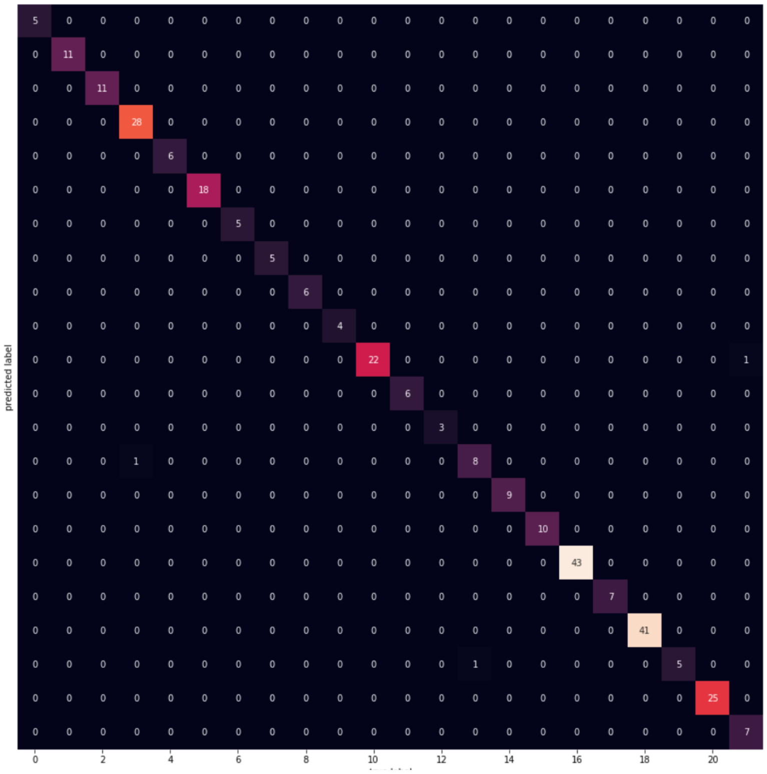
Figure 5.5: Model 3- Classification report for Testing

5.3 Evaluating the model for Animal Audio classification

Three models, CNN, CNN-SVM and CNN-XGBoost were designed and tested on Animal audio through this research work. The classification report for every model as well as the accuracy shows different measurements. As per the performance table 4.1, the accuracy of the first model designed using CNN exceeds Model 2 and Model 3. The accuracy of Model 1 also exceeds the baseline accuracy for audio classification provided by Google for AudioSet dataset and also, exceed the score achieved by Dang, Vu, and Wang (2017) on audio classification and detection of acoustic sound and event. However, even after applying many dropout layers in the CNN model to avoid overfitting, all the model score extending 95% could be due to overfit model.

This could also be a parameter while deciding the perfect model along with the accuracy and performance of the model. Out of model 2 and model 3, CNN-XGBoost definitely performs better on the training data as compared to CNN-SVM. Both the Models, Model 2 and model 3 performs almost similar on the validation and test data. A basic CNN model was designed for feature extraction to avoid severe performance issue due to combined model. Also, random sampling was used to split the dataset and many of the class have F1-score, precision almost equivalent to 0 which indicates that the model performance is very weak in comparison to model1. It could be due to fewer number of records available for training for many audio class. The performance of atleast 20-25 class is perfect with CNN-XGBoost classifier which indicate if the training size is increased to have multiple records for each class the CNN-XGB classifier might be best fit for Animal audio classification.

With respect to scope of this research and the classifiers tested, it is seen that the model 1, CNN outperforms all other classifier in predicting all the class with an accuracy of approximately 97% on the testing set. A confusion matrix or error matrix is a tabular visualization of statistical classification per class. Each row value represents the predicted class value whereas the column value represents the actual value of the class. The diagonal entries are the number of correctly classified instances while all other values represent entries that are unclassified. A confusion matrix for model 1 is shown in Figure 5.6. , the colored diagonal row in the matrix indicate the model classified the records for all class.



0		2		4		6		8		10		12		14		16		18		20	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	5	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7

Figure 5.6: Confusion matrix-Model 1

Confusion matrix for model 2, CNN-SVM is shown in Figure 5.7, only few diagonal row has value and other records are either missed or not classified.

Confusion matrix for model 3, CNN-XGBoost is shown in Figure 5.8, show stats similar to Model 2 classifiying only few records and missing the other ones.

Micro/Macro/Weighted average precision recall and F1 score of all the models can be compared to decide the best classification model out of three model designed as part of

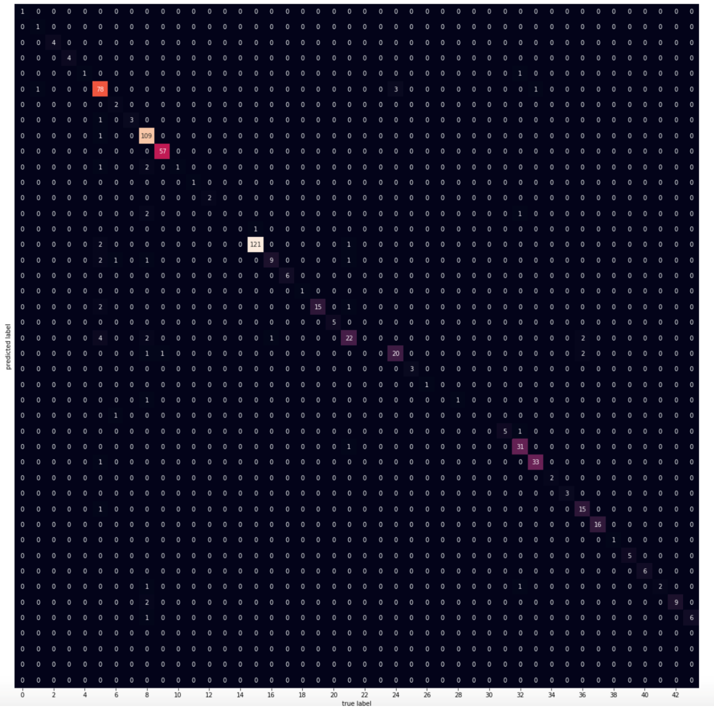


Figure 5.7: Confusion matrix-Model 2

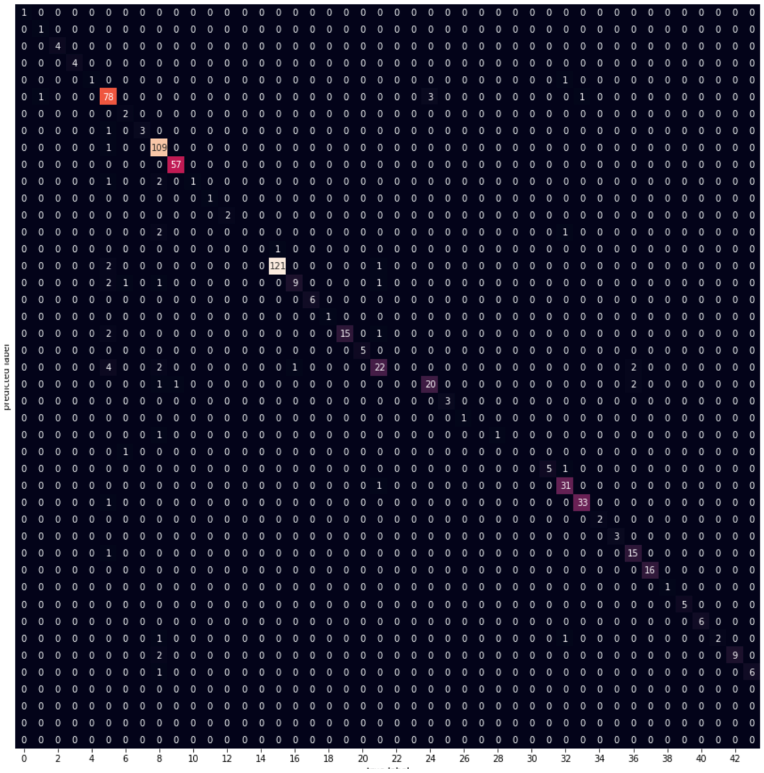


Figure 5.8: Confusion matrix-Model 3

	Precision, Recall and F-1 Score		**Weighted Average**		
Model	**Micro Average**	**Macro Average**	**Precision**	**Recall**	**F-1 Score**
CNN	0.99	0.99	0.99	0.99	0.99
CNN-SVM	0.67	0.40	0.69	0.68	0.69
CNN-XGBoost	0.68	0.40	0.69	0.68	0.69

Table 5.4: Comparative Summary

this research experiment. In micro-average method, the individual true positives, false positives and false negatives of the systems for different sets are summed and averaged to get the stats. As our research work is a case of multiclass label classification, the measurements for precision, recall and F1-score for a particular will be same. It only differs in case of weighted averaging were the produced F1-score is not between precision and recall.

The Table 5.4, shows that the precision, recall and F1-score of CNN model is higher in comparison to other models. Thus, it can be concluded that CNN model built using set of multiple convolution and max pooling performs the task of both the feature extraction and classification accurately for animal audio classification. It also decreases the computational work on the machine as the framework consist of just one classifier instead of multiple ones.

As discussed in chapter 1, the principal objective of this research was to reject the null hypothesis by answering the research objective A, B and C. The research objective A was to convert audio waveforms to mel-spectrogram using Librosa library which was completed after data processing in this research work and the count of the generated spectrogram was validated with no spectrogram was found to be missing. The objective B was to analyze the changes in performance trained with spectrogram, the performance of the designed model was recorded and was compared with the baseline accuracy of 73% in the research work by Yue-Hei Ng et al. (2015). The objective C was to compare the performance of the designed classifiers by measuring the accuracy, precision, F1-score and confusion matrix. After analyzing the evaluation score in the table 5.4, it can be seen that the CNN model built with the same dataset has an accuracy, precision and F1 score and model performance better than other models, CNN-SVM and CNN-XGBoost built with same dataset. On the basis of the analysis done and the evaluation of the research objectives we can conclude that CNN model outperforms CNN-SVM and CNN-XGBoost in classifying the animal sound thereby rejecting the null hypothesis.

5.4 Strength and Limitation of Result

As part of the research, Convolutional neural network and hybrid models were designed to classify animal audio using mel-spectogram generated from the audio waveforms. The study's key strength was its ability to accurately identify and classify the animal sound in the audio files. The designed framework though new for audio sound is actually similar to existing AI architecture like one discussed in the paper Takahashi et al. (2016). Audio waveforms were pre-processed, and mel-spectrogram were generated from the waveform to train the algorithm. One of the key strengths of this model is that it can be used in future to classify different audio classes.

The limitation of the model is it requires huge computational power when the dataset is large. For this research work, the amount of data was limited making it simpler to train the model. If the model is extended in future for processing huge number of records, machine with high computational power will be required. Also, the pre-processing done on the data was specific to the format of the files downloaded. Additional mechanisms need to be implemented to make the audio generic in nature. The noise in the audio also needs to be filtered to maintain classification accuracy. The model is train on mel-spectogram which are generated from the audio waveform which will be a limiting factor. In case audio embeddings are provided for large dataset like in AudioSet additional computational power will be required in generating the spectrogram.

Chapter 6

Discussion and Conclusion

This chapter explains and discuss the outcomes inferred through the research. A brief summary of problem definition, Experiment design, evaluation and result along with research overview is detailed in this chapter. It also provides suggestion for future work.

6.1 Research Overview

The research, Animal audio classification has been carried out in parts. The first part was classifying the audio into different animal classes and second part was to extract audio from the videos to understand the feasibility of the framework designed on video classification. Animal audio was downloaded from various sources and processed for training the model. Three models were designed as part of the research work motivated using AI architecture like ImageNet and others discussed in paper [41]. The performance of the models was evaluated and compared with existing algorithms. Also, a comparative analysis was done by measuring the classification performance of the algorithm in terms of precision, accuracy and F1 score. The statistical measurements evaluated and analyzed from all the models were used to reach decision on the formulated hypothesis.

6.2 Problem Definition

The study was defined by the research question "Can precision, accuracy and F1-score for animal sound detection and classification in the audio files be better achieved using deep convolutional neural network model trained on frame per second and spectrogram of the audio as compared to hybrid models built using CNN with traditional machine learning classifier SVM or XGBoost?" and the purpose of the research was driven to validate the hypothesis and draw conclusion for the classifier on the basis of acceptance and rejection of hypothesis. The hypothesis for this research is as illustrated in chapter 1.

Null hypothesis - Classification and detection of animal audio using deep neural network (CNN) model built on frame per second and spectrogram of audio files does not provide improvement in precision, accuracy and F1-score more than hybrid models CNN-SVM, CNN-XGBoost.

The research primarily focused in building algorithms to classify animal sound in 46 classes. Based on the evaluation and classification scores of the designed models, the results clearly show the difference in performance of CNN classifier giving basis to reject the null hypothesis.

6.3 Experimentation, Evaluation and Results

This research tries to build a model for animal audio with good accuracy and a state-of-the-art solution using machine learning methods. The experiment design was sound and well rooted, as it included a thorough analysis of the design of an animal audio classifier. The model was tested on 8000 instances belonging to 46 different class. With the increase in popularity of the website for social networking and the amount of multimedia content uploaded, constant attention was paid to content management. Recently, many datasets have been made available publicly which drove the motivation for this research. According to literature review described in chapter 2, many researchers have made progress in the field of audio classification. The research paper [42] and [43] on DCASE 2017 audio classification challenge using CNN and RNN discuses and accuracy of 82% and 74.8%. In [43], author also discussed on improving the error rate performance to 0.59% from 0.69%. For instance, research work [21] which discuses a hidden markov model for audio tagging and classification shows a precision of 0.7 and recall of 0.9 respectively. Similarly, the research work by [7] which was one of the comparison models provides an accuracy of 73%.

The framework proposed in this research focuses on using deep network properties of CNN and designing an architecture that helps in feature extraction and appropriate classification. The research aims at building three models, a CNN with set of multiple convolution and pooling layer and two hybrid models. The hybrid models were built using machine learning classifier SVM and XGBoost on the feature extracted from the CNN. Between the computation power required for hybrid models built on basic CNN with SVM or XGBoost and CNN there was difference of milliseconds. CNN-SVM and CNN-XGBoost algorithm had classification accuracy of 68% and 72% respectively on the same dataset. The CNN network won among the three models with an accuracy of 98.95% and 0.9 as precision, recall and F1-score. Finally, the designed framework improved the accuracy when compared to previous classifiers built on audio classification. Also, all the three models achieved the goal of classifying the animal audio and proves to be valid for classifying animal audio.

6.4 Contributions and Impact

Content management is an evolving space of research in recent times. With the ability to classify the content by audio the scope of multimedia classification will expand. One of the stopping stone in building model is the amount of data required to train the model initially. After ImageNet was released publicly the world of Image classification touched new horizons. Many big firms thus initiated in releasing multiple audio dataset with the hope of achieving greater heights in audio classification similar to image classification. Recently, audio embedding from YouTube video was released which attracted lot of attractions. As of now, the content management for video classification is completely driven by the images but the resolution of the images can often lead to misclassification.

If audio from the video is used along with the image the chances of misclassification will be less, which served as a motivation for task 2 of this research work. Audio from the video was converted into audio embeddings and thereby in a spectrogram using VGG architecture produced by Google. The Research primarily concentrates on building an audio classifier for animal sound that exceeds the classification accuracy of the models designed so far for audio classification. Post training the CNN classifier built as part of this research work for audio of different types, the same model can be used in future for classifying sound of all types. In addition, such as image classification having an efficient audio classification algorithm serves the purpose of classifying multimedia files of audio and video format.

6.5 Future Work and Recommendations

The model was built using free sound available on multimedia sites which had filtered audios specific to animal class and had little or less noise. In future, a better prediction model can be built by including unfiltered audio data. Future work could look into altering the modalities of the CNN built assuring less computational power for large scale audio classification. Changes can be made to the basic CNN model used in the hybrid models to increase the classification accuracy. A hybrid model, using Model 1 from this research for feature extraction and traditional machine learning classifier for classification can also be worked on in future. As of now, designing this hybrid model consumes lot of computational power and hence more research needs to be done.

References

Agnihotri, L., & Dimitrova, N. (1999). Text detection for video analysis. In *Proceedings ieee workshop on content-based access of image and video libraries (cbaivl'99)* (pp. 109–113).

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems* (pp. 892–900).

Boreczky, J. S., & Wilcox, L. D. (1998). A hidden markov model framework for video segmentation using audio and image features. In *Proceedings of the 1998 ieee international conference on acoustics, speech and signal processing, icassp'98 (cat. no. 98ch36181)* (Vol. 6, pp. 3741–3744).

Brezeale, D., & Cook, D. J. (2008). Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(3), 416–430.

Cai, R., Lu, L., Hanjalic, A., Zhang, H.-J., & Cai, L.-H. (2006). A flexible framework for key audio effects detection and auditory context inference. *IEEE Transactions on audio, speech, and language processing*, *14*(3), 1026–1039.

Chou, S.-Y., Jang, J.-S. R., & Yang, Y.-H. (2017). Framecnn: A weakly-supervised learning framework for frame-wise acoustic event detection and classification. *Recall*, *14*, 55–64.

Dang, A., Vu, T. H., & Wang, J.-C. (2017). Deep learning for dcase2017 challenge. *DCASE2017 Challenge, Tech. Rep.*

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (pp. 248–255).

Elizalde, B., Kumar, A., Shah, A., Badlani, R., Vincent, E., Raj, B., & Lane, I. (2016). Experiments on the dcase challenge 2016: Acoustic scene classification and sound event detection in real life recording. *arXiv preprint arXiv:1607.06706*.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., … Ritter, M. (2017a). Audio set: An ontology and human-labeled dataset for audio events. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 776–780).

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., … Ritter, M. (2017b). Audio set: An ontology and human-labeled dataset for audio events. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 776–780).

Hassanzadeh, H. R., & Wang, M. D. (2016). Deeperbind: Enhancing prediction of sequence specificities of dna binding proteins. In *2016 ieee international conference on bioinformatics and biomedicine (bibm)* (pp. 178–183).

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., … others (2017a). Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 131–135).

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., … others (2017b). Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 131–135).

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Pro-*

ceedings of the ieee conference on computer vision and pattern recognition (pp. 1725–1732).

Kong, Q., Xu, Y., & Plumbley, M. D. (2017). Joint detection and classification convolutional neural network on weakly labelled bird audio detection. In *2017 25th european signal processing conference (eusipco)* (pp. 1749–1753).

Kong, Q., Xu, Y., Sobieraj, I., Wang, W., & Plumbley, M. D. (2019). Sound event detection and time–frequency segmentation from weakly labelled data. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, *27*(4), 777–787.

Kong, Q., Yu, C., Iqbal, T., Xu, Y., Wang, W., & Plumbley, M. D. (2019). Weakly labelled audioset classification with attention neural networks. *arXiv preprint arXiv:1903.00765*.

Kumar, A., Dighe, P., Singh, R., Chaudhuri, S., & Raj, B. (2012). Audio event detection from acoustic unit occurrence patterns. In *2012 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 489–492).

Lee, H., Pham, P., Largman, Y., & Ng, A. Y. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems* (pp. 1096–1104).

Li, S. Z., & Guo, G.-d. (2000). Content-based audio classification and retrieval using svm learning. In *First ieee pacific-rim conference on multimedia, invited talk, australia.*

Lin, W.-H., & Hauptmann, A. (2002). News video classification using svm-based multimodal classifiers and combination strategies. In *Proceedings of the tenth acm international conference on multimedia* (pp. 323–326).

Lu, L., Ge, F., Zhao, Q., & Yan, Y. (2010). A svm-based audio event detection system. In *2010 international conference on electrical and control engineering* (pp. 292–295).

Mertens, R., Lei, H., Gottlieb, L., Friedland, G., & Divakaran, A. (2011a). Acoustic super models for large scale video event detection. In *Proceedings of the 2011 joint acm workshop on modeling and representing events* (pp. 19–24).

Mertens, R., Lei, H., Gottlieb, L., Friedland, G., & Divakaran, A. (2011b). Acoustic super models for large scale video event detection. In *Proceedings of the 2011 joint acm workshop on modeling and representing events* (pp. 19–24).

Mesaros, A., Heittola, T., Eronen, A., & Virtanen, T. (2010a). Acoustic event detection in real life recordings. In *2010 18th european signal processing conference* (pp. 1267–1271).

Mesaros, A., Heittola, T., Eronen, A., & Virtanen, T. (2010b). Acoustic event detection in real life recordings. In *2010 18th european signal processing conference* (pp. 1267–1271).

Phan, H., Maaß, M., Mazur, R., & Mertins, A. (2014). Random regression forests for acoustic event detection and classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(1), 20–31.

Pleva, M., Vozáriková, E., Ondáš, S., Juhár, J., & Čižmár, A. (2010). Automatic detection of audio events indicating threats. In *Ieee international conference on multimedia communications, services and security, krakow* (Vol. 6).

Ravanelli, M., Elizalde, B., Ni, K., & Friedland, G. (2014). Audio concept classification with hierarchical deep neural networks. In *2014 22nd european signal processing conference (eusipco)* (pp. 606–610).

Takahashi, N., Gygli, M., Pfister, B., & Van Gool, L. (2016). Deep convolutional neural networks and data augmentation for acoustic event detection. *arXiv preprint arXiv:1604.07160*.

Tang, Y. (2013). Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*.

Temko, A., Malkin, R., Zieger, C., Macho, D., Nadeu, C., & Omologo, M. (2006). Acoustic event detection and classification in smart-room environments: Evaluation of chil project systems. *Cough*, *65*(48), 5.

Xiong, Z., Radhakrishnan, R., Divakaran, A., & Huang, T. S. (2005). Highlights extraction from sports video based on an audio-visual marker detection framework. In *2005 ieee international conference on multimedia and expo* (pp. 4–pp).

Xu, Y., Kong, Q., Wang, W., & Plumbley, M. D. (2018). Large-scale weakly supervised audio classification using gated convolutional neural network. In *2018 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 121–125).

Yu, C., Barsim, K. S., Kong, Q., & Yang, B. (2018). Multi-level attention model for weakly supervised audio classification. *arXiv preprint arXiv:1803.02353*.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4694–4702).

Zhang, D., Gatica-Perez, D., Bengio, S., & McCowan, I. (2005). Semi-supervised adapted hmms for unusual event detection. In *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)* (Vol. 1, pp. 611–618).

Zhou, X., Zhuang, X., Liu, M., Tang, H., Hasegawa-Johnson, M., & Huang, T. (2007). Hmm-based acoustic event detection with adaboost feature selection. In *Multimodal technologies for perception of humans* (pp. 345–353). Springer.

Zhuang, X., Zhou, X., Huang, T. S., & Hasegawa-Johnson, M. (2008). Feature analysis and selection for acoustic event detection. In *2008 ieee international conference on acoustics, speech and signal processing* (pp. 17–20).

Appendices

.1 Data Exploration and Model Implementation

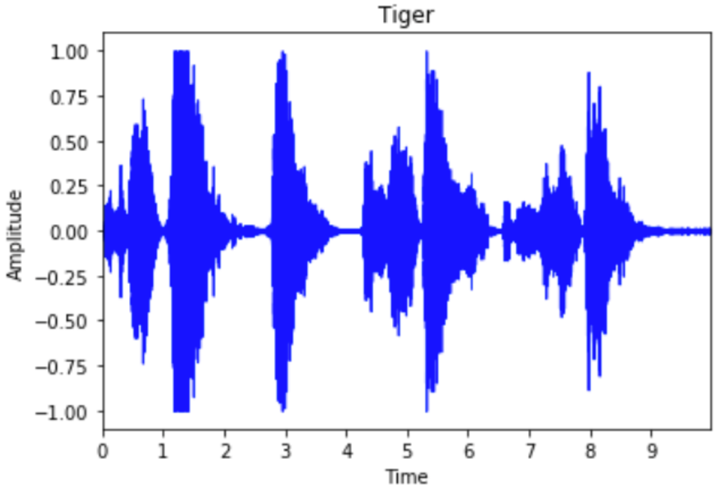


Figure 1: Tiger Amplitude plot

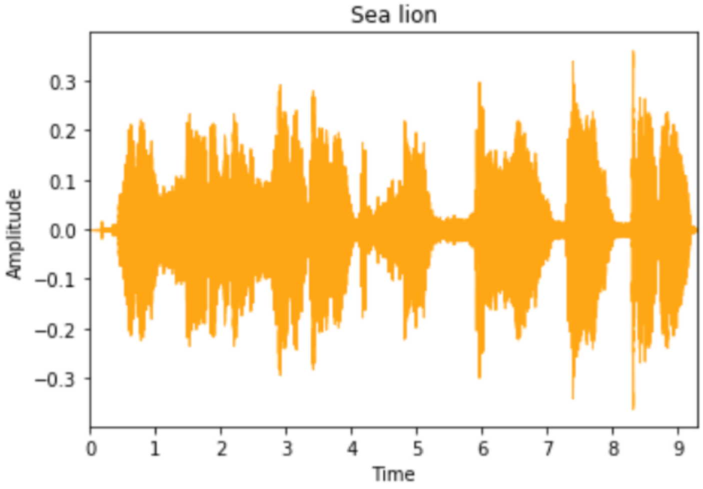


Figure 2: Sea Lion Amplitude plot

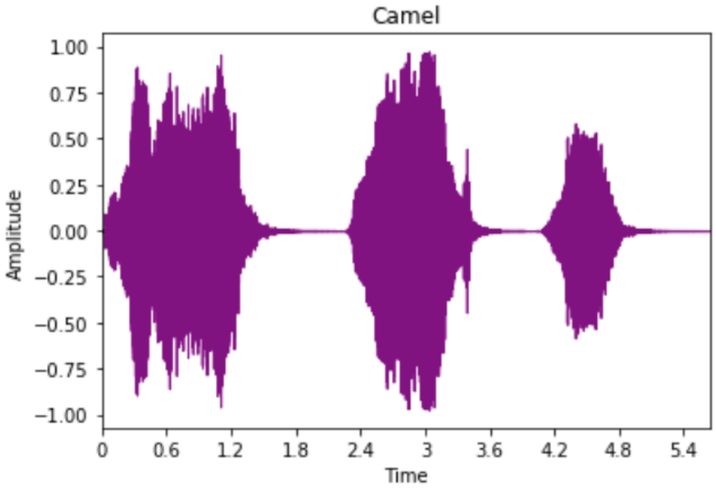


Figure 3: Camel Amplitude plot

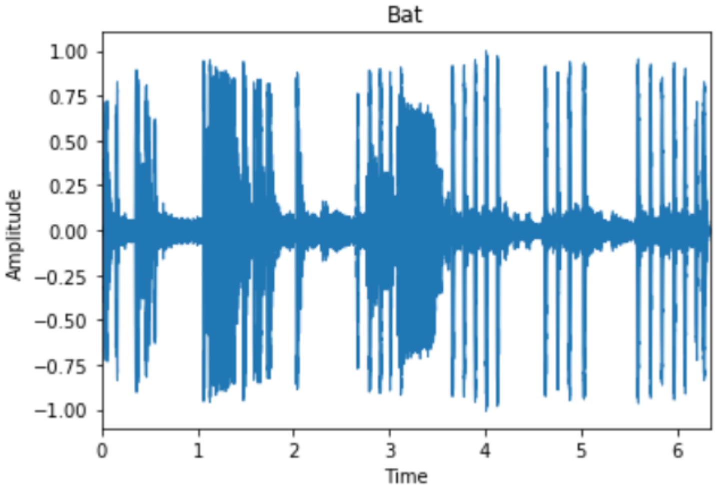


Figure 4: Bat Amplitude plot

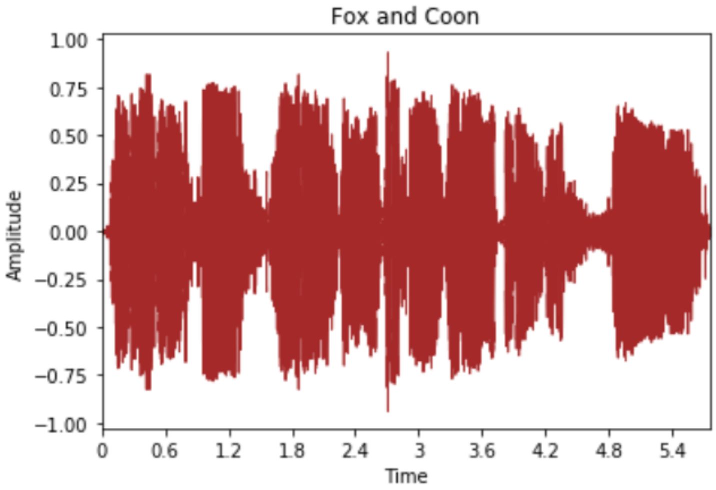


Figure 5: Fox and Coon Amplitude plot

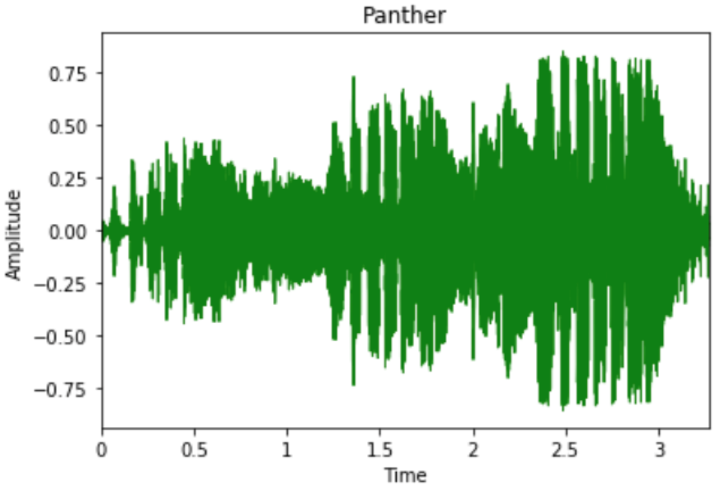


Figure 6: Panther Amplitude plot

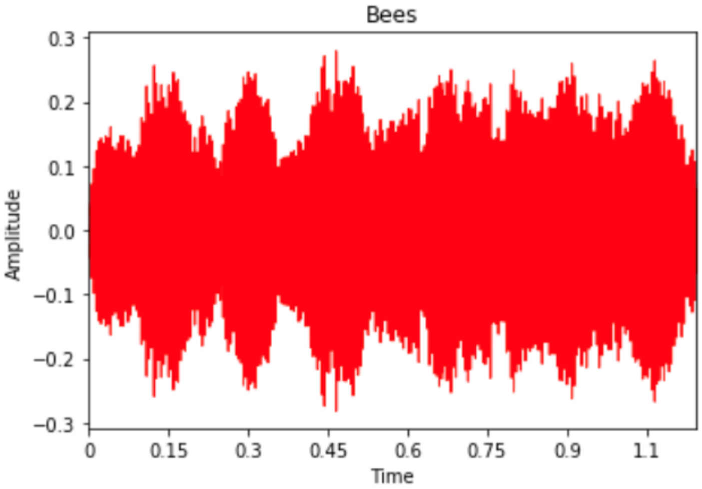


Figure 7: Bees Amplitude plot

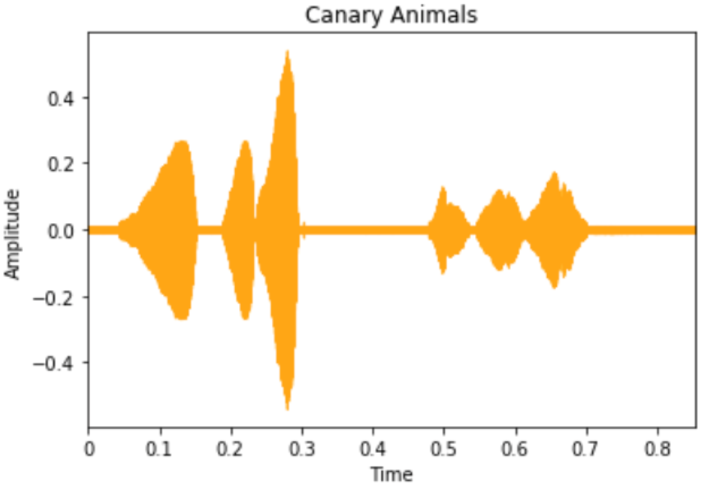


Figure 8: Canary Animals Amplitude plot

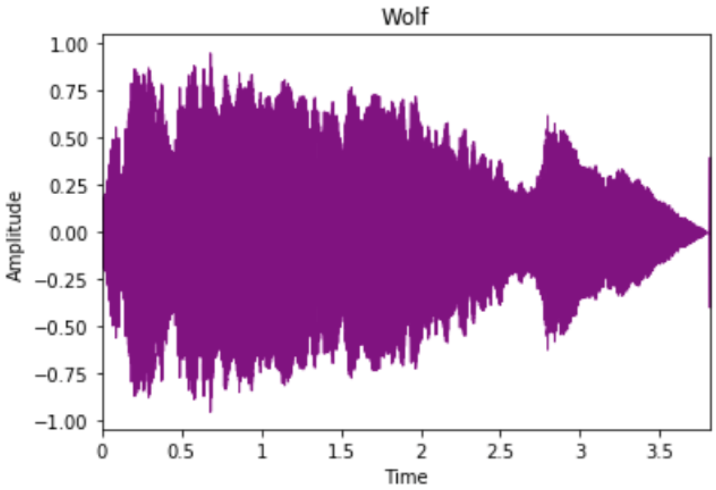


Figure 9: Wolf Amplitude plot

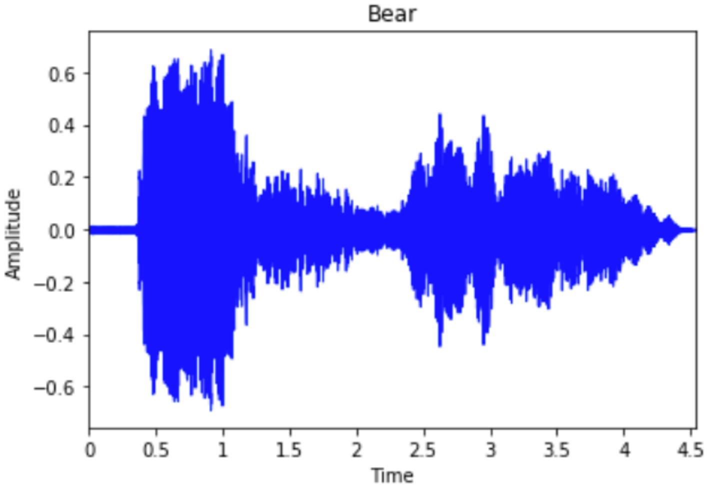


Figure 10: Bear Amplitude plot

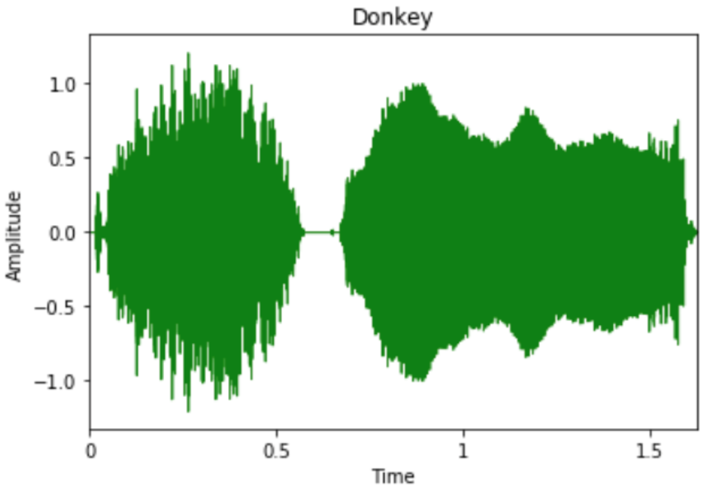


Figure 11: Donkey Amplitude plot

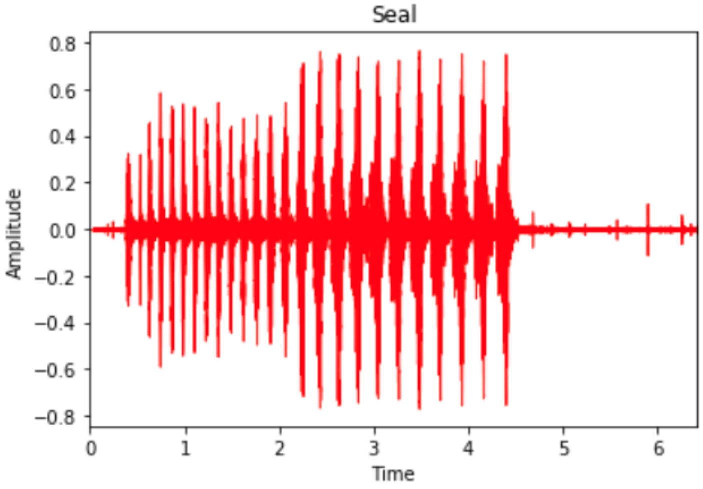


Figure 12: Seal Amplitude plot

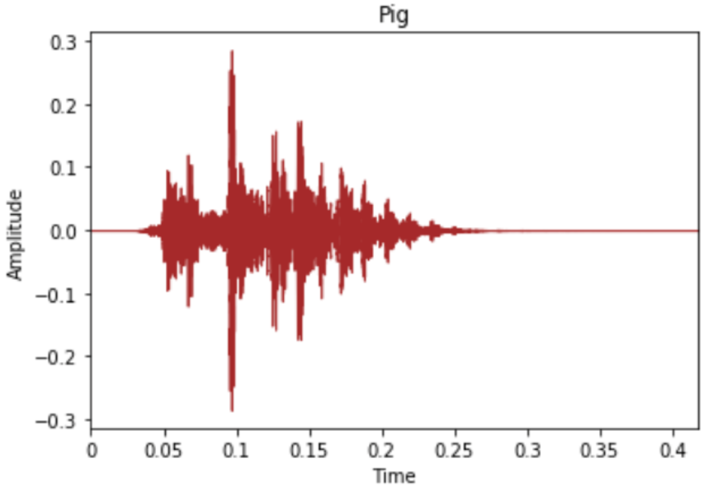


Figure 13: Pig Amplitude plot

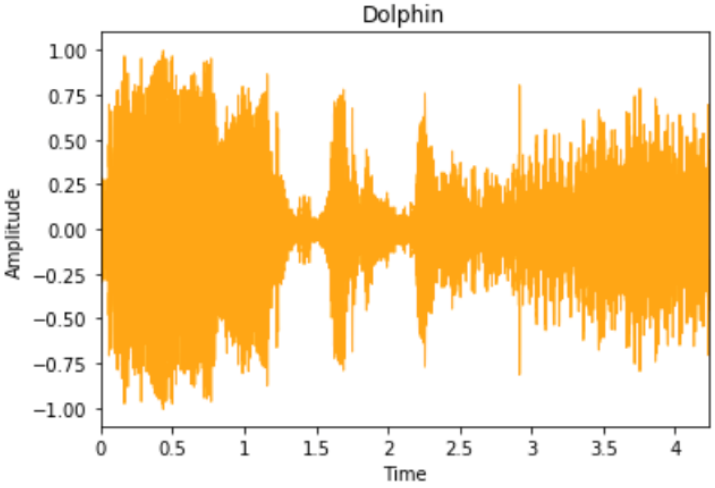


Figure 14: Dolphin Amplitude plot

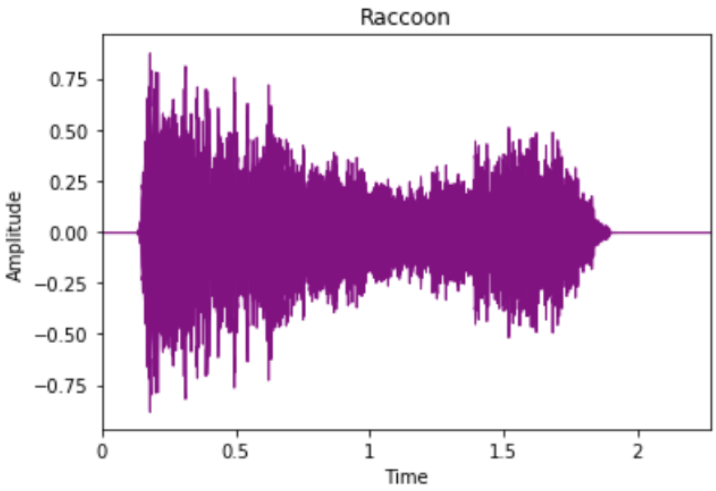


Figure 15: Raccoon Amplitude plot

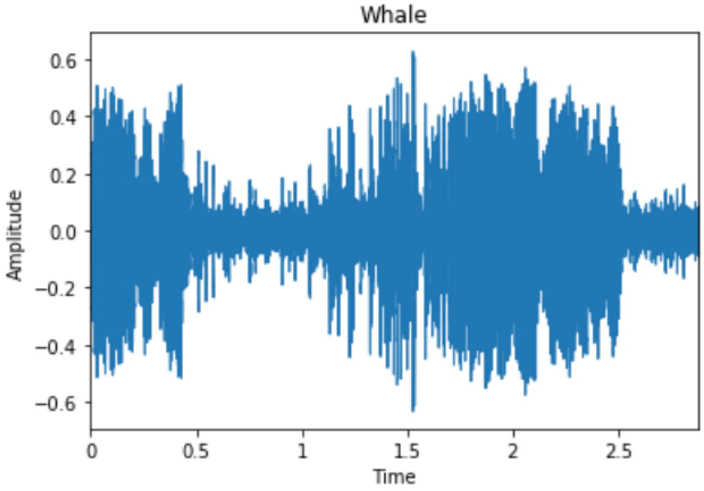


Figure 16: Whale Amplitude plot

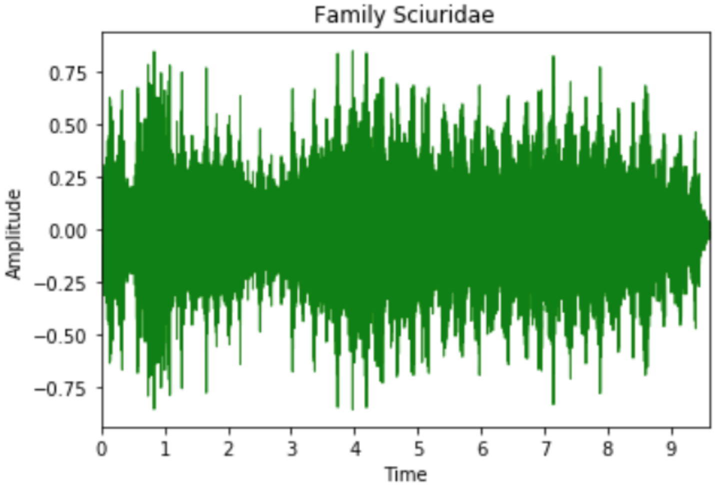


Figure 17: Family Sciuridae Amplitude plot

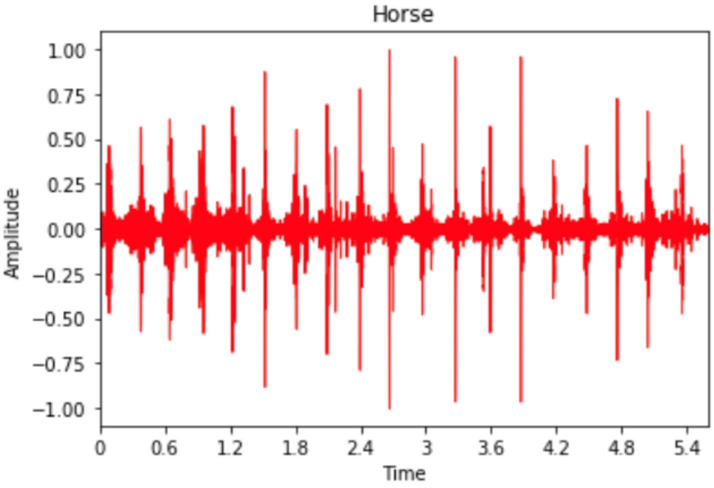


Figure 18: Horse Amplitude plot

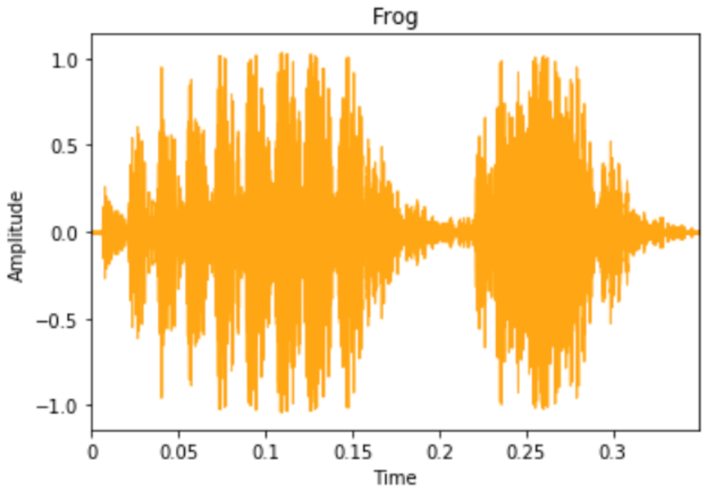


Figure 19: Frog Amplitude plot

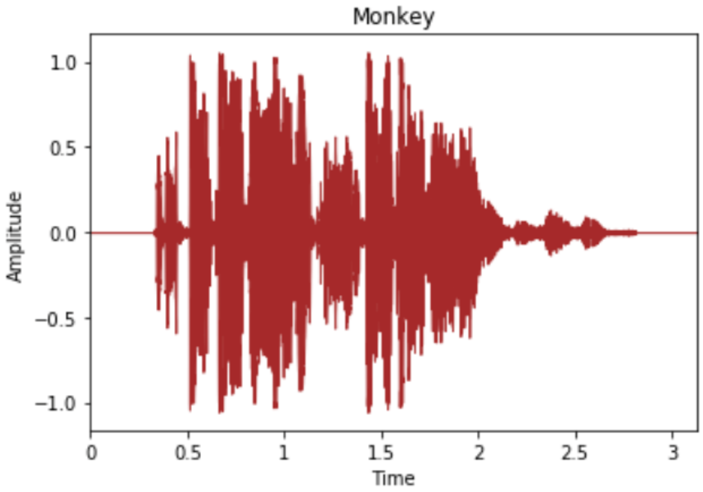


Figure 20: Monkey Amplitude plot

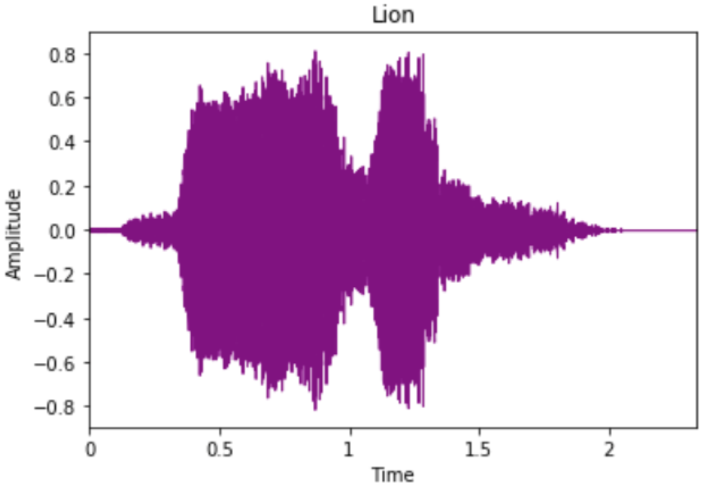


Figure 21: Lion Amplitude plot

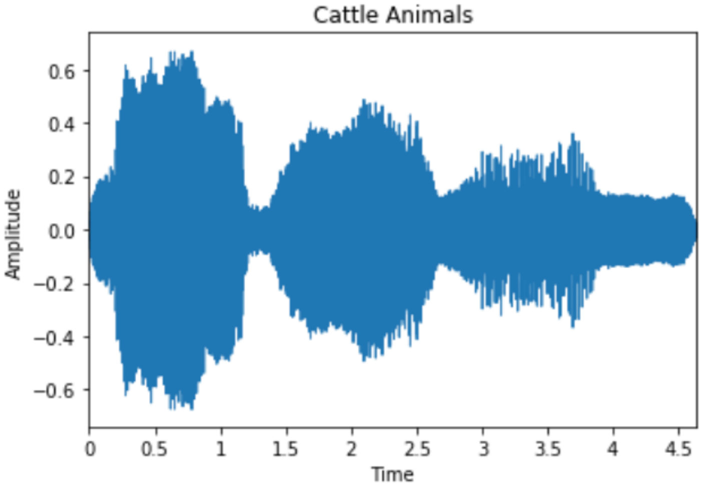


Figure 22: Cattle Animals Amplitude plot

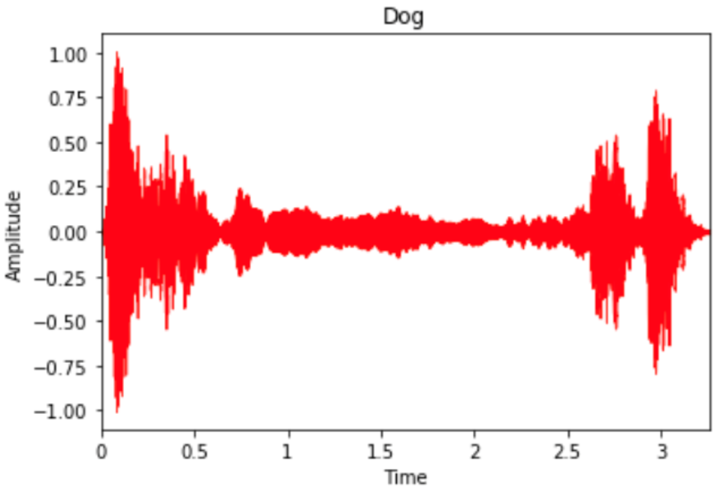


Figure 23: Dog Amplitude plot

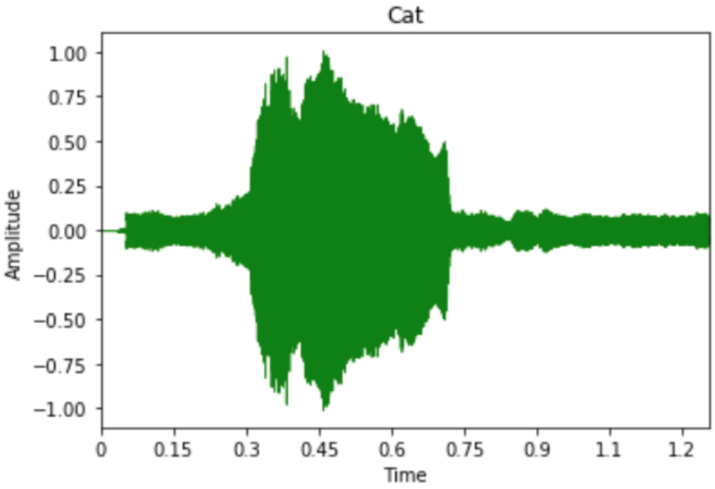
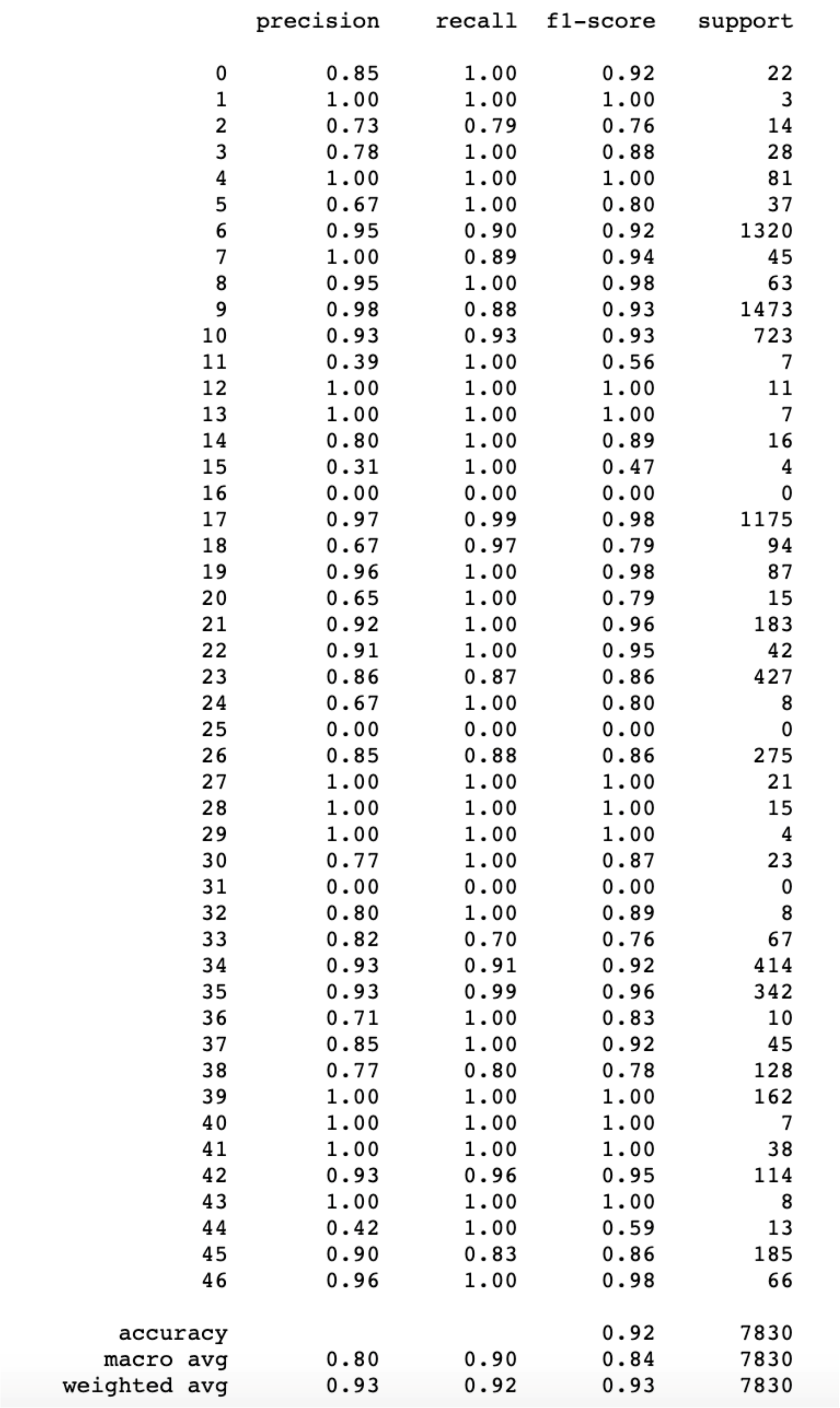


Figure 24: Cat Amplitude plot

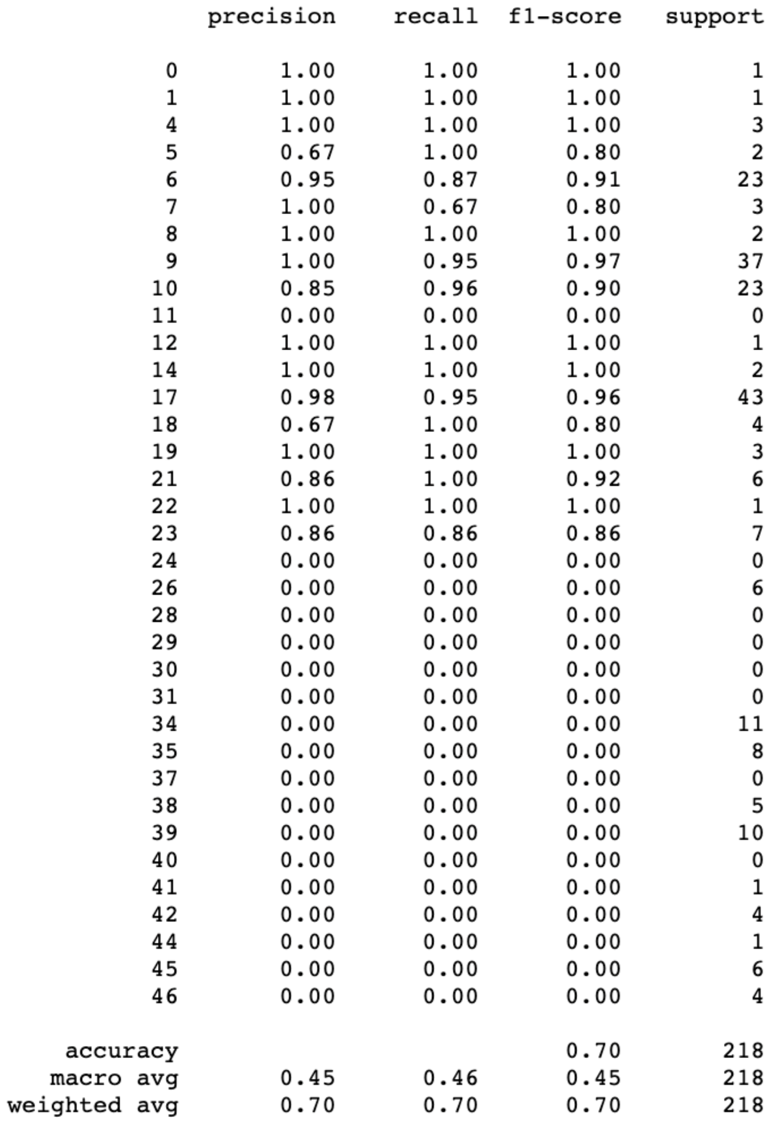


```
              precision    recall  f1-score   support

           0       0.85      1.00      0.92        22
           1       1.00      1.00      1.00         3
           2       0.73      0.79      0.76        14
           3       0.78      1.00      0.88        28
           4       1.00      1.00      1.00        81
           5       0.67      1.00      0.80        37
           6       0.95      0.90      0.92      1320
           7       1.00      0.89      0.94        45
           8       0.95      1.00      0.98        63
           9       0.98      0.88      0.93      1473
          10       0.93      0.93      0.93       723
          11       0.39      1.00      0.56         7
          12       1.00      1.00      1.00        11
          13       1.00      1.00      1.00         7
          14       0.80      1.00      0.89        16
          15       0.31      1.00      0.47         4
          16       0.00      0.00      0.00         0
          17       0.97      0.99      0.98      1175
          18       0.67      0.97      0.79        94
          19       0.96      1.00      0.98        87
          20       0.65      1.00      0.79        15
          21       0.92      1.00      0.96       183
          22       0.91      1.00      0.95        42
          23       0.86      0.87      0.86       427
          24       0.67      1.00      0.80         8
          25       0.00      0.00      0.00         0
          26       0.85      0.88      0.86       275
          27       1.00      1.00      1.00        21
          28       1.00      1.00      1.00        15
          29       1.00      1.00      1.00         4
          30       0.77      1.00      0.87        23
          31       0.00      0.00      0.00         0
          32       0.80      1.00      0.89         8
          33       0.82      0.70      0.76        67
          34       0.93      0.91      0.92       414
          35       0.93      0.99      0.96       342
          36       0.71      1.00      0.83        10
          37       0.85      1.00      0.92        45
          38       0.77      0.80      0.78       128
          39       1.00      1.00      1.00       162
          40       1.00      1.00      1.00         7
          41       1.00      1.00      1.00        38
          42       0.93      0.96      0.95       114
          43       1.00      1.00      1.00         8
          44       0.42      1.00      0.59        13
          45       0.90      0.83      0.86       185
          46       0.96      1.00      0.98        66

    accuracy                           0.92      7830
   macro avg       0.80      0.90      0.84      7830
weighted avg       0.93      0.92      0.93      7830
```

Figure 25: Classification Report Training - Model 2

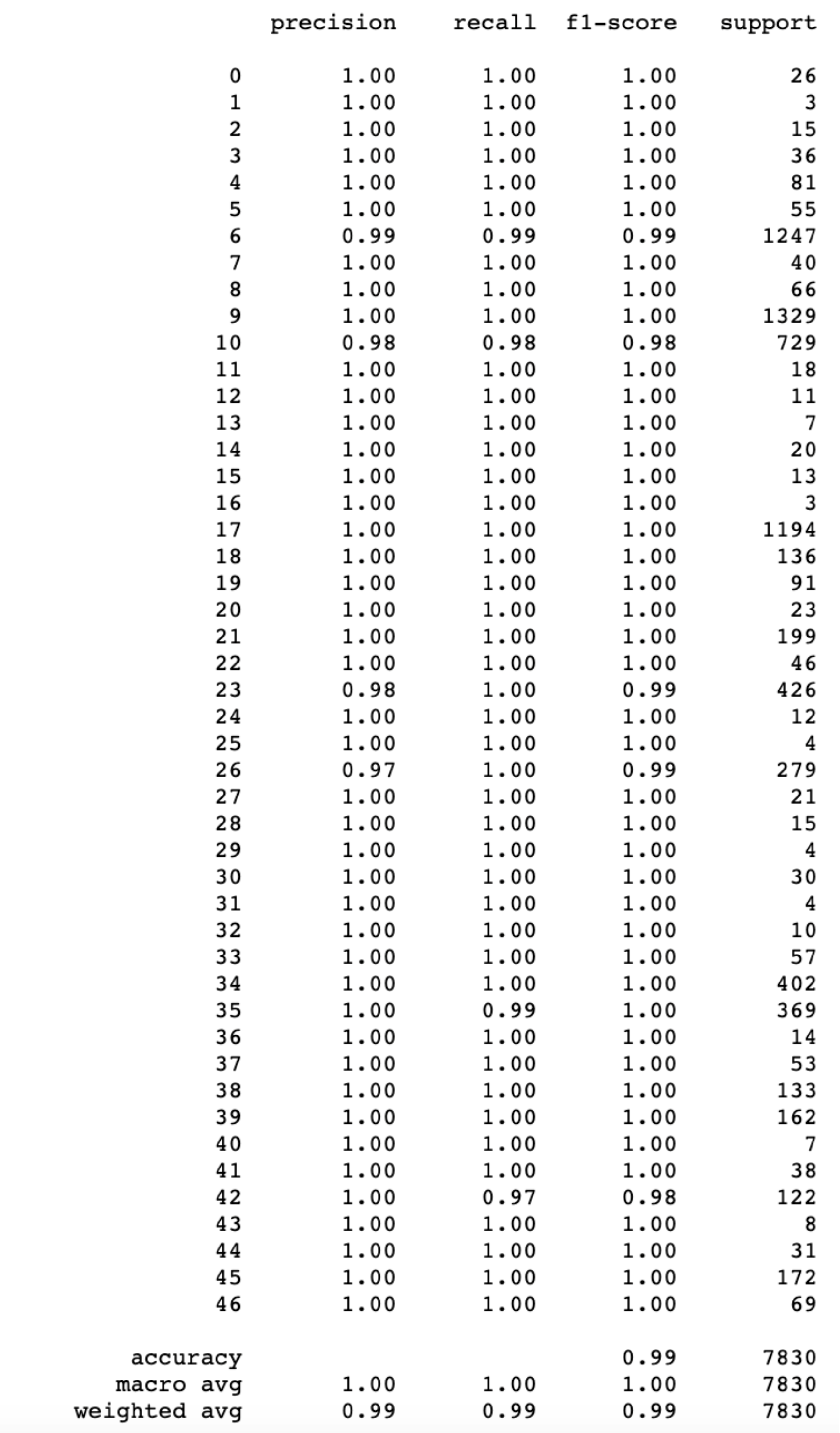


```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1
           4       1.00      1.00      1.00         3
           5       0.67      1.00      0.80         2
           6       0.95      0.87      0.91        23
           7       1.00      0.67      0.80         3
           8       1.00      1.00      1.00         2
           9       1.00      0.95      0.97        37
          10       0.85      0.96      0.90        23
          11       0.00      0.00      0.00         0
          12       1.00      1.00      1.00         1
          14       1.00      1.00      1.00         2
          17       0.98      0.95      0.96        43
          18       0.67      1.00      0.80         4
          19       1.00      1.00      1.00         3
          21       0.86      1.00      0.92         6
          22       1.00      1.00      1.00         1
          23       0.86      0.86      0.86         7
          24       0.00      0.00      0.00         0
          26       0.00      0.00      0.00         6
          28       0.00      0.00      0.00         0
          29       0.00      0.00      0.00         0
          30       0.00      0.00      0.00         0
          31       0.00      0.00      0.00         0
          34       0.00      0.00      0.00        11
          35       0.00      0.00      0.00         8
          37       0.00      0.00      0.00         0
          38       0.00      0.00      0.00         5
          39       0.00      0.00      0.00        10
          40       0.00      0.00      0.00         0
          41       0.00      0.00      0.00         1
          42       0.00      0.00      0.00         4
          44       0.00      0.00      0.00         1
          45       0.00      0.00      0.00         6
          46       0.00      0.00      0.00         4

    accuracy                           0.70       218
   macro avg       0.45      0.46      0.45       218
weighted avg       0.70      0.70      0.70       218
```

Figure 26: Classification Report Validation - Model 2

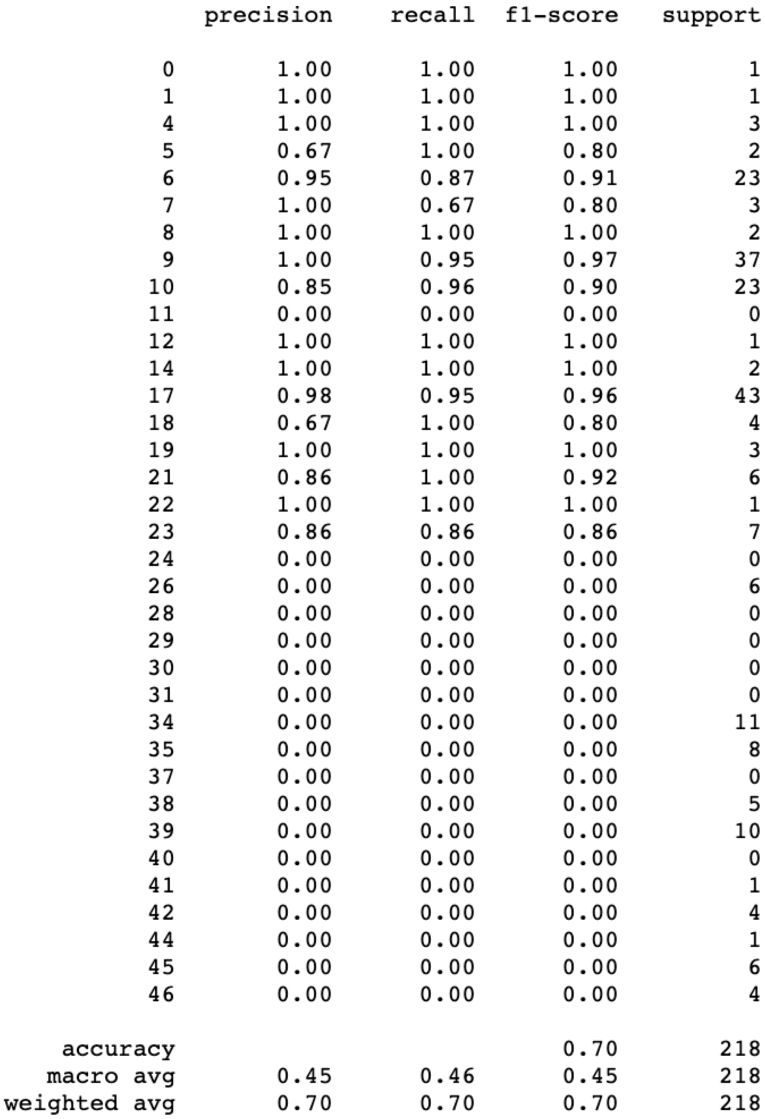


```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        26
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00        15
           3       1.00      1.00      1.00        36
           4       1.00      1.00      1.00        81
           5       1.00      1.00      1.00        55
           6       0.99      0.99      0.99      1247
           7       1.00      1.00      1.00        40
           8       1.00      1.00      1.00        66
           9       1.00      1.00      1.00      1329
          10       0.98      0.98      0.98       729
          11       1.00      1.00      1.00        18
          12       1.00      1.00      1.00        11
          13       1.00      1.00      1.00         7
          14       1.00      1.00      1.00        20
          15       1.00      1.00      1.00        13
          16       1.00      1.00      1.00         3
          17       1.00      1.00      1.00      1194
          18       1.00      1.00      1.00       136
          19       1.00      1.00      1.00        91
          20       1.00      1.00      1.00        23
          21       1.00      1.00      1.00       199
          22       1.00      1.00      1.00        46
          23       0.98      1.00      0.99       426
          24       1.00      1.00      1.00        12
          25       1.00      1.00      1.00         4
          26       0.97      1.00      0.99       279
          27       1.00      1.00      1.00        21
          28       1.00      1.00      1.00        15
          29       1.00      1.00      1.00         4
          30       1.00      1.00      1.00        30
          31       1.00      1.00      1.00         4
          32       1.00      1.00      1.00        10
          33       1.00      1.00      1.00        57
          34       1.00      1.00      1.00       402
          35       1.00      0.99      1.00       369
          36       1.00      1.00      1.00        14
          37       1.00      1.00      1.00        53
          38       1.00      1.00      1.00       133
          39       1.00      1.00      1.00       162
          40       1.00      1.00      1.00         7
          41       1.00      1.00      1.00        38
          42       1.00      0.97      0.98       122
          43       1.00      1.00      1.00         8
          44       1.00      1.00      1.00        31
          45       1.00      1.00      1.00       172
          46       1.00      1.00      1.00        69

    accuracy                           0.99      7830
   macro avg       1.00      1.00      1.00      7830
weighted avg       0.99      0.99      0.99      7830
```

Figure 27: Classification Report Training - Model 3



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1
           4       1.00      1.00      1.00         3
           5       0.67      1.00      0.80         2
           6       0.95      0.87      0.91        23
           7       1.00      0.67      0.80         3
           8       1.00      1.00      1.00         2
           9       1.00      0.95      0.97        37
          10       0.85      0.96      0.90        23
          11       0.00      0.00      0.00         0
          12       1.00      1.00      1.00         1
          14       1.00      1.00      1.00         2
          17       0.98      0.95      0.96        43
          18       0.67      1.00      0.80         4
          19       1.00      1.00      1.00         3
          21       0.86      1.00      0.92         6
          22       1.00      1.00      1.00         1
          23       0.86      0.86      0.86         7
          24       0.00      0.00      0.00         0
          26       0.00      0.00      0.00         6
          28       0.00      0.00      0.00         0
          29       0.00      0.00      0.00         0
          30       0.00      0.00      0.00         0
          31       0.00      0.00      0.00         0
          34       0.00      0.00      0.00        11
          35       0.00      0.00      0.00         8
          37       0.00      0.00      0.00         0
          38       0.00      0.00      0.00         5
          39       0.00      0.00      0.00        10
          40       0.00      0.00      0.00         0
          41       0.00      0.00      0.00         1
          42       0.00      0.00      0.00         4
          44       0.00      0.00      0.00         1
          45       0.00      0.00      0.00         6
          46       0.00      0.00      0.00         4

    accuracy                           0.70       218
   macro avg       0.45      0.46      0.45       218
weighted avg       0.70      0.70      0.70       218
```

Figure 28: Classification Report Validation - Model 3