Speech Command Classification Based on Deep Neural Networks

by

Md. Sakib Hossain
18101201
Syed Tamzidul Islam
22241133
Sujat Mazumder
18101300
Ali Imran Joy
18301179
Md. Sadman Sakib
18301061

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
23 March,2023.

Declaration

We, hereby, are glad to state that, the paper "Speech Command Classification Based on Deep Neural Networks", has been initiated while pursuing the thesis of under graduation under BRAC University, and the paper presented is a piece of our own original research. We can assure that our paper does not contain any information that has been presented or accepted by any other academic or other institution, except for where stated by reference. All sources that have been helpful to us, while completing the paper are properly cited and recognized.

Student's Full Name & Signature:

Sadman

MD. Sadman Sakib

18301061

Sakib Hossain

MD. Sakib Hossain

18101201

Tamzid

Syed Tamzidul Islam

22241133

Sujat

Sujat Mazumder

18101300

Ali Imran

Ali Imran Joy

18301179

Approval

The thesis paper entitled ‘Speech Command Classification Based on Deep Neural Networks’ has been submitted by:

1. MD. Sakib Hossain (18101201)
2. Syed Tamzidul Islam (22241133)
3. Sujat Mazumder (18101300)
4. Ali Imran Joy (18301179)
5. MD. Sadman Sakib (18301061)

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

Examining Committee:

Supervisor:
(Member)

Aminul Huq
Lecturer
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam
PhD Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi
PhD Associate Professor and Chairperson
Department of Computer Science and Engineering
BRAC University

Ethics Statement

Our study is for the betterment of the hearing impaired. The data and study provided can be fully relied upon. We did not conduct any action while pursuing our research.Optimistically, we expect our findings to be of a use in the future.

Abstract

In our day-to-day life there are lots of sounds that we are processing. To process these sounds our brain absorb sound signals and provide us informative knowledge. For human being this is not possible to extract every sounds properly so that, there are lots of equipment which helps us to extract essential information from an audio source. Around the year lots of model came to help thorough extract informations using various algorithms. Also, some models are Convolutional Neural Network (CNN), Region-Convolutional Neural Network (R-CNN), Artificial Neural Network (ANN), VGG16, ResNet50 and Numerous machine learning algorithms have been utilized to effectively categorize audio, and these methods have recently demonstrated encouraging results in separating spectrotemporal images from various sound classifications. The study purpose of this research was to analyze which feature extraction method shows maximum result using Convolutional Neural Network (CNN), VGG16 and ResNet50. In the proposed model, MFCC feature extraction method are taken from the dataset and trained using a multiple layer-based con volution neural network. In the experimental assessment, a sound dataset consisting of 105829 audio clips separated up into multiple groups of important sounds during study used to develop the models. Additionally, we evaluated the models' validity which reach an accuracy of 94.53% on SpeechCommand dataset.

Keywords: Sound Classification, Spectrograms, SpeechCommand, CNN, ResNet50.

Acknowledgement

First and foremost, we would want to express our gratitude to Allah, the Almighty, for keeping us safe during the Covid pandemic. Although some of us felt ill, with Allah's help, we were spared from contracting the terrible virus. Because of the existence of despite being under heavy lockdown, we were nevertheless able to contact with one another effectively and simply thanks to the internet. We received a ton of encouragement and assistance from our esteemed thesis professors throughout the writing process. Aminul Huq, helped us whenever we required it. Without them, together with Rafeed Rahman, our thesis co-supervisor, our work would not have been able to be finished. In addition, our parents deserve a lot of praise for giving us the ongoing support and attention we required. Thanks to their ongoing support, we were able to comfortably finish our paper during our whole undergrad program without encountering any significant obstacles. Finally, we would want to express our gratitude to BRAC University, our university, for giving us the resources and connections that not only enabled us to finish our thesis paper but also helped us improve throughout the course of four incredible and healthy years.

Nomenclature

CNN Convolutional Neural Network

VGG Visual Geometry Group

MFCC Mel-frequency cepstral coefficients

ReLU Rectified linear activation function

RNN Recurrent Neural Network

SNR Signal to Noise Ratio

BN Batch Normalization

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

The emergence of AI and intelligent assistants like Google Assistant, Amazon Alexa, Apple Siri, and Microsoft Cortana have boosted the use of speech recognition and its appeal in the modern world. Speech recognition is the capacity of a computer or software to understand spoken words and phrases and translate them into a machine-readable format. Speech recognition has several applications, including voice calling, call forwarding, search phrases, and simple data entry. The majority of sectors today use speech recognition as a key technology, including television, voice call routing, voice dialing, search keywords, and simple data entry. We are aware that whenever we call a customer service number, a virtual assistant will answer the phone and assist us until we can speak with the real person we are contacting about. Call routing was used in this situation, which is the technique of routing voice calls to a certain queue in accordance with predetermined criteria.

A call distribution program is another name for a call routing system. The process of routing calls to the appropriate agent became crucial since conventional models of customer care relied heavily on phone assistance or call support as one of the key methods of contact between customers and businesses for business purposes. Today's agents communicate with clients in a variety of ways.

Voice-activated dialing, By allowing users to make audio calls to anyone without providing any phone numbers, it has increased the convenience of calling. However, voice-enabled calling might not work well if someone is in a location with a lot of noise or another disruption since more than two or more voices will blend together and make it difficult for them to hear. In order to get around this, we need to employ the best noise-canceling technique we can discover that can minimize noise to some extent. Speech recognition software is used for voice dialing, often known as a voice-enabled calling. We may now enter certain data into Word or Excel documents using our voice. We may do hands-free data entry by saying the text or numbers we want to enter in the current cell. By only saying their names, we may even utilize voice commands to select menu items, dialogue box options, or even toolbar buttons. This saves us time and enables us to complete the task more rapidly than typing.

For voice data entry, speech recognition software has been used. When using Speech Recognition to narrate data entry, we must keep the microphone in the same position while speaking into it. Depending on the quality of our microphone, it may take some time before our words show on the Formula bar and in the current cell. We must thus talk naturally, in a low but not monotonous voice, halting only when we reach the end of a thought or the data input for that cell. This can be improved by training an increasing volume of audio data using deep learning or machine learning approaches.

As we all know, Google Assistant and Microsoft Cortana are used frequently nowadays to perform tasks like searching for information online or on a computer, including looking for files, folders, documents, and a variety of other things. All of these actions are carried out using our voice, whether we utter anything or instruct Google Assistant or Microsoft Cortana to look something up and provide us with the information we need. Therefore, Google Assistant, Cortana, and Siri on Apple devices all heavily rely on speech recognition. The accuracy of Google Assistant, Microsoft Cortana, and Apple Siri's speech-to-text conversion is 3 getting better every day. If we want to classify audio input, we can focus on the amplitude of the signal, or on the time domain as well. The time domain is more beneficial since it takes into account frequency content as well.

1.2 Research Objective

Research on automatic sound recognition has accelerated recently and has been applied in a variety of diverse domains, including multimedia [9], bioacoustics monitoring [8], and speech recognition [11].ambient sounds [17], audio surveillance [10], and intrusion detection in wildlife regions. The three stages of the sound recognition challenge are signal pre-processing, feature extraction, and feature classification. The input signal is split into many segments during signal pre-processing, which are then used to extract associated features. Data size is decreased through feature extraction, which converts complex data into feature vectors. Additionally, the performance of sound recognition and classification systems was improved by the use of several machine learning and soft computing approaches including the Hidden and Gaussian Mixture Model, Random Forest, Multi-Layer Perceptron, and Emerging Deep Learning Networks.

Similar to how humans classify things, these features are utilized to divide sounds into separate groups. A spectrogram can be used to see the frequency spectrum of a sound wave, as seen in Fig. 1. It can be described as a snapshot of the frequency spectrum found in a sound wave [15].

This research aims to identify environmental noises using deep learning networks based on the generated spectrograms of these sounds.

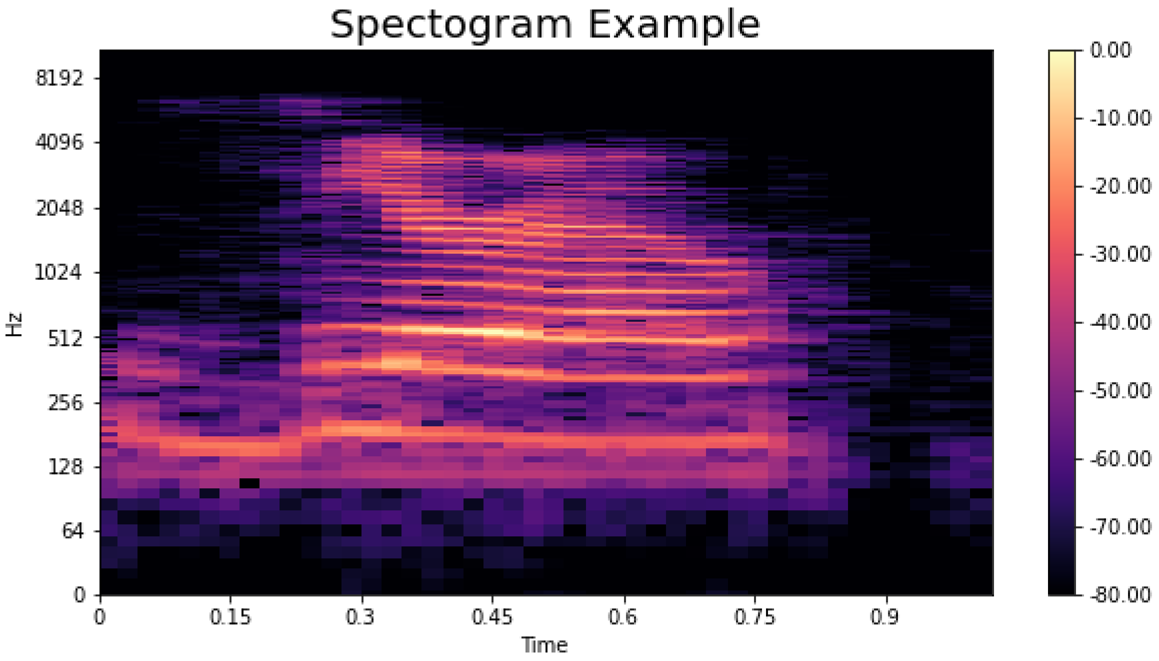

Figure 1.1: Figure of Spectrogram

1.3 Thesis Orientation

We have divided our thesis into 5 chapters, the description of the chapter are as follows:
In chapter 1, we talked about the introduction to sound-related applications, and we talked about our problem statement and our Research objectives.

In chapter 2, We listed all relevant works that we thought might be useful.

In chapter 3, We've included a thorough explanation of the models we utilized to create our project.

In chapter 4, gives a description of the data collection we used. We discussed feature extraction from the gathered data and data preprocessing.

In chapter 5, In numerous tables and visualizations, the output of the classifier is compared to our own observations of various classifiers and circumstances.

Chapter 2

Literature Review

This paper was written by E. S¸a¸smaz and F. B. Tek. MFCC and Chroma STFT [6] stack together features which are used in the LTSM model giving state-of-the-art results. In our world, we are surrounded by many sounds which are not possible to process by our brain cells. That's why many research fields are working on sound classification for many purposes such as audio surveillance, multimedia, and many more. so, as said before many features are used in this paper. Those are spectral contrast, tonnetz, MFCC, Chroma CENS and Chroma CQT. From models, there are used CNN and LSTM. There are lots of other papers that are used in this paper for better performance. The dataset shows deep learning models are performing better performance than machine learning models. The audio samples were in a wave but it was then changed into a one-dimensional NumPy array of digital values. There was a librosa package used for normalizing so that values can be represented in the array. Also works as time stretch and pitch shifting. After implementing all spectrograms in CNN and LSTM shows MFCC was the best feature. Where stacking MFCC and Chroma STFT helps to reach a validation accuracy of 98.81% where the best performance was 98.60%. Also, LTSM does a better job because the LSTM memory cell includes constant error backpropagation which deals with data noise.

Cardiovascular diseases have a leading cause of mortality rate where Wavelet representations and RNN [9] can be used for recognizing three heart sounds i.e., normal, mild, and severe. These works need physicians who are trained also approx. 20% of the medical interns on average can use efficient use of stereoscopes to measure subjects' heartbeats. There are some limitations in work such as publicly accessible data, and deep learning methods are not comprehensively studied. In the experiment, coif3 was selected for wavelet type. DRNN model was optimized to have three layers with adam optimizer. So, as result, wavelet-based DRNN works with excellent performance in the reckoning mild class. The value of normal and severe can be improved. This is hard for this model to distinguish between these three models.

Machine learning advances have sparked renewed interest [16] in a variety of classification challenges, particularly those incorporating data in the form of photos, videos, and audio recordings. Classifying sounds and predicting their category is one of the most common classification challenges. Security systems, classifying music clips to identify the genre of the song, classifying distinct surrounding sounds, speaker recognition, and verification are some of the real-world uses for such a classification model. The task of assessing diverse audio signals is known as audio classification. They gave a brief overview of the field of audio classification in this work, describing the system, several modules of feature extraction and modeling, applications, underlying approaches, and certain performance indicators. Following this introduction, we'll go over some of the present classification technologies' strengths and drawbacks, as well as some prospective future research, development, and application trends. Many audio classification subtasks rely heavily on inputs, network topologies, temporal pooling strategies, and objective functions, so we paid special attention to them. Finally, the study discusses future trends and research prospects in this field.

One audio finger methodology [20] for audio classification is presented in this research. The audio signal's fingerprint is a unique digest that can be used to identify it. To establish a distinctive fingerprint of the audio files, the suggested model employs the audio fingerprinting methodology. The MFCC spectrum is extracted, its mean is calculated, and the spectrum is then transformed into a binary image to create the fingerprints. These pictures are then sent to the LSTM network, which uses them to categorize the ambient sounds present in the UrbanSound8K dataset with an overall accuracy of 98.8% over all 10 folds

It's very tough to deliver security for women and children in light of increased crime against them. Several contemporary techniques Nowadays, they're employed to provide security. Sensors-based gadgets [10] are also included here, while others are mobile apps. But When sensors are present, hardware-based gadgets do not function properly and are cut off from their bodies Existing applications on Mobile gadgets are not always functional. Victims must take action. Specific reactions such as the phone starting to shake and SOS button is pressed, which may or may not be available at all times. As audio one of the most advanced applications of deep learning is classification. They presented an idea to protect women and children through learning. Audio classification is used by youngsters. The victim's screaming sounds, which can be heard from quite a distance, have alerted them of danger. For audio classification, Some deep neural network models are being used, so differences in audio and reaction screams can be considered a sign of danger. They've also created a new dataset specifically for this.

As part of a unique strategy to deal with the availability of unknown sound classes (open set) and the lack of training resources, variational auto-encoder (VAE), data augmentation, and detection-classification combined training are included in typical GAN networks [9]. The VAE input to GAN generator helps in the generation of realistic outlier samples that are not too distant from the in-distribution class, improving the open-set discriminating capabilities of classifiers. Then, the augmentation-enhanced GAN scheme developed in their past work for close-set audio classification would incorporate physical data augmentations with traditional GAN-produced samples to address the constrained training resources. Overfitting will be avoided, and optimization convergence will be improved. The VAE and Augmentation GAN strengths are combined in training for detection and classification to improve task performance. Experiments using the Google Speech Command database reveal a significant improvement in open set classification accuracy from 62.41% to 88.29%when only 10% of the training data is used.

To improve labor activity recognition and remote construction project monitoring, the Deep Belief Network (DBN)-based algorithm [6] for audio signal categorization is used. The goal of this project is to provide a flexible platform for carrying out and managing unmanned development location observation using dispersed sound sensors. In this study, ten classes of various construction tools and equipment that are often and widely used on construction sites were gathered and studied in order to conduct and validate the proposed technique. A concatenation of various statistics is sent to the DBN and assessed using a variety of spectral characteristics, including MFCCs and mel-scaled spectrograms. The architecture that was offered, as well as the preprocessing and feature extraction steps, have all been carefully characterized, and numerical results based on real-world recordings have demonstrated the applicability of the suggested concept. Up to 98 percent overall accuracy on the test set was achieved, which is far higher than previous cutting-edge approaches. There has also been a suggestion for the practical use of the technique to apply the classification system to sound data collected in diverse environmental contexts.

Sound files for Animal sound classification using deep learning and CNN architecture were preprocessed [9] in such a way that it can extract MFCC using librosa. Some studies show that with a limited dataset CNN using log MEL-spectrogram performs best with 64.5% accuracy. 11 The data was collected online in WAV format and these are used in 3 different ways. First, they examine all manually to prevent low-quality sound. Secondly, all are WAV format because MFCC supports WAV format in feature extraction. Thirdly, the size of the sound file was 3 kilobytes. By using Librosa all datasets were preprocessed as binary files for Training and Testing. For activation functions it used Relu. For testing, it takes 80% of the data and for training, it takes 20% of data finally shows 75% accuracy by Nesterov-accelerated adaptive moment estimation.

Urban sound classifications have multiple features [17], which are implicated in different neural networks and it shows which model gives the ameliorated accuracy in audio classification signals. In our world, we are surrounded by many sounds which are not possible to process by our brain cells. That's why many research fields work on sound classification for many purposes such as audio surveillance, multimedia, and many more. so, as said before, many features are used in this paper. From models, there are CNN and LSTM. There are lots of other papers that are used in this paper for better performance. The dataset, shows deep learning models are performing better performance than machine learning models. The audio samples were in wave form but they were converted into a one-dimensional NumPy array of digital values. There was a librosa package used for normalizing so that values can be represented in the array. Also works as time stretch and pitch shifting. After implementing all spectrograms in CNN and LSTM it shows that MFCC has the best features. Where stacking MFCC and Chroma STFT helps to reach a validation accuracy of 98.81% where the best performance was 98.60%. Also, LTSM does a better job because the LTSM memory cell includes constant error back propagation which deals with data noise. So, lastly, this paper shows that using MFCC and

Chroma STFT stack together features that have been implemented in the LTSM model are giving state-of-the-art results.

CNN is a good way to implement Environmental sound classifications [22]. For this CNN some steps were followed in this paper. All the data are labeled for efficient learning. There were multiple datasets used (ECS-50, ECS-10). After implementing it in CNN it shows better results even with a limited dataset. When the dataset number will increase it will perform more accurately in the environmental dataset.

Speaking Faces combines synchronized audio, thermal, and visual information gathered from a variety of subjects. To demonstrate the usefulness of their data, they performed multimodal gender categorization utilizing thermal, visual, and aural data streams as well as thermal-to-visible picture translation. Based on the outcomes of the experiments, they discovered that Speaking Faces had the following positive benefits. It first enables a more thorough investigation into multimodal recognition systems that employ optical, thermal, and auditory modalities. Second, the dataset's vast sample size allows for the development and testing of data-hungry neural network approaches. Finally, synchronized multimodal data can open up fresh research perspectives on domain transfer. In the future, they want to use our dataset for other multimodal tasks including speaker and audiovisual-thermal speech. We proposed data selection methods based on word and idea level confidences to make use of cheaply accessible untranscribed in-domain data. This was utilized as an addition to the in-domain transcription data to enable the language and acoustic models to be modified. In the future, we'll look into employing bigger amounts of untranscribed data.

After establishing the VQC-based QNN, the CNN-QNN-based SCR system may be employed in SCR as a hybrid classical to quantum transfer learning technique for QNNs, as described in [19]. A hybrid transfer learning framework receives a pre-trained CNN framework. so that we may enhance the CNN-QNN system's capabilities Researchers discovered that hybrid classical-to-quantum transfer learning increases classification accuracy and decreases cross-entropy loss of the CNN-QNN model's value using the Google speech command dataset.

A smart home system [19] saves time and energy, especially when a large number of people are involved. A large number of people are involved. It might be expanded to include video surveillance to detect persons in crowded places like bus stops, theatres, and train stations, where the perpetrator's identity can be verified. Face recognition techniques are used. In the field of computer vision, the recognition system is a challenging matter to tackle. Due to its wide uses in a variety of sectors, it has recently sparked a lot of attention. Despite extensive efforts, Research efforts in this field have yielded robust facial recognition systems that can work in a variety of environments. They are still far from fulfilling the ideal of being able to perform well in limited spaces.

Door Access Control System [13] only enables those who have an approved key card and whose voice has been recorded in the system. The technology of voice recognition recognizes your voice and the words you say. Because your voice can-not be stolen, it is a safe way to prevent illicit admission into an organization. If you're experiencing difficulty speaking, An RFID-enabled key card can be utilized (probably sick). As a prototype, the work was successful, and the prototype was successful. 14 The door was able to unlock when the user spoke into the microphone or swiped the RFID card. This is the first version. Further improvements to this work can be done before.

The largest challenge to speech recognition, especially for Deep Learning, is adversarial attacks. [22]. Some researchers have created hostile speech samples for speech recognition based on FGSM. Moreover, genetic algorithms were employed to trick voice recognition systems into believing all is well. This essay the effectiveness of adversarial assaults on voice recognition components is examined for mission-critical systems and offers defenses against these attacks. The current of voice recognition technology is discussed, along with the difficulties and upcoming research directions. Speech recognition is a simple method of communication. It allows automated systems to understand spoken language and a textual transcription is provided. The potential uses for combining speech recognition and conversational systems are numerous. The several methods for Mel Frequency Cepstral Coefficients (MFCC), perceptual, and other features are extracted. Fast Fourier Transform, Line Prediction Cepstral Coefficient (LPCC), Discrete Wavelet Transform (DWT), Linear Prediction (PLP), Linear Predictive Coding (LPC), and FFT, Line Spectral Frequencies, and (LSF). Considering the most probable order, the During the decoding process, words from the input audio file are created. To achieve this, learning strategies, especially hidden Markov models (HMMs) or deep learning the Viterbi algorithm, which minimizes searches and identifies the ideal polynomial-time route), are used. Using the statistical method GMM-HMM, making sense of hidden information in visual data. The voice recognition system's decoder As seen, the system is modeled as a Markov process with unknown parameters. by the parameters of known observation. combined Hidden Markov Model systems perform better when combined with a Gaussian Mixture Model than when done independently. models of hybrid DNNHMMs, where a DNN takes the role of the acoustic module while maintaining the other modules served as the initial Deep Learning strategy's foundation. for the detection of speech. It is currently common to construct extensive voice recognition systems using DNNs. This method can circumvent the weakness of the previous system that The entire situation might not be ideal.

Short-time Fourier transform (STFT) spectrograms are utilized to extract features using a convolutional neural network [18]. The ICBHI 2017 Respiratory 16 Sound Database was used to train and test the model, which produced ground-breaking results using three different data-splitting techniques. The score is 64.92%, specificity is 82.46%, accuracy is 73.69%, and sensitivity is 47.37%. The four types of lung sounds—normal, crackles, wheezes, and both—as well as the temporal correlations between the data are remembered using long short-term memory (LSTM) networks. To extract characteristics, short-time Fourier transformations were initially applied. In this work, they proposed a hybrid DL architecture that combines CNN and LSTM models for detecting intentional and unintentional lung sounds. Lung noises, comprising three distinct types of unintentional sounds like crackles and wheezes as well as ordinary sounds, are used to train and test the model. In this work, respiratory cycles were transformed into STFT spectrograms and put through a CNN that can extract the important characteristics of those spectrograms. The retrieved characteristics were put into an LSTM that understands and recalls long-term relationships between them.

The CNN-TT-DNN model, which may drastically reduce the number of model parameters while maintaining the CNN model's baseline performance, replaces fully connected (FC) layers with TT ones. According to the outcomes of our studies, the proposed CNN +(TT-DNN) model achieves a competitive accuracy of 96.31% with four times fewer model parameters than the CNN model. In order to create each of the four parts that make up the overall CNN architecture, 1D convolutional layers with batch normalization and ReLU activation are stacked. The spectral properties associated with the CNN framework's outputs are sent to the FC layers or TT layers. Our basic system was built using the CNN+DNN architecture, with several FC layers added on top of the CNN layers. THEIR models are trained to utilize the exact SCR dataset from the beginning without any data augmentation to achieve fair architecture-wise study. To describe its ultimate outcomes, we expand the 10-class training configuration that was applied in 35 courses. Three more neural networks that are mentioned in the literature are contrasted with the CNN+(TT) and CNN+(DNN) models. As the default SCR system, the CNN+DNN model achieves 94.42 percent accuracy and a CE score of 0.251. It performs better in terms of smaller model sizes, lower CE values, and higher classification accuracy than the DenseNet, neural attention, and QCNN models.

TTTD [15] can maintain and even generate better results than the basic models. Baseline models (DenseNet, neural attention, and QCNN models) are applied when the CNN+(TT-DNN) model is either created from a CNN+(TT-DNN) model that has been trained before or initialized randomly. Tucker decomposition is unable to preserve the DNN baseline results acquired with CNN + DNN models. This study investigates the application of the TT technique to build a simple SCR system. The CNN+ hybrid model is offered as an end-to-end SCR pipeline (TT-DNN). Either a trained CNN+DNN or a random generator can produce the recommended model.

By weighing trade-offs between model complexity and real performance, low-complexity hybrid tensor networks are created [19]. Moreover, CNN+(LRTT-DNN) has numerous TT layers at the top for problem-solving in regression and classification and convolutional layers at the bottom for feature extraction. Initially, we construct the low-rank tensor-train deep neural network (LR-TTDNN), a comprehensive deep learning pipeline (TT-DNN). An over-parameterized deep neural network (DNN) simplifies the optimization landscape and guarantees that local optimum points are close to global optimum points. Several innovative applications, like mobile-based audio de-noise and speech recognition systems that operate on users' phones without sending queries to a distant server where a large deep learning model is built up, may be made possible by an effective low-complexity speech improvement system. Deep learning models may be shrunk in two ways. One example of a novel deep learning architecture that might be employed is convolutional neural networks (CNN) with a variety of cutting-edge topologies. Model pruning and sparseness approaches are mentioned by another. The experiments of voice augmentation and spoken command recognition (SCR) systems are utilized to show the value of our suggested models as we study the implementation of low-rank tensor-train (TT) networks. They have created a fresh CNN+TT-DNN deep hybrid tensor-train model. A CNN is at the bottom, and a DNN is at the top. Time-series data is transformed into the relevant spectral properties using the CNN model. The DNN is used to further resolve problems with classification or regression. To evaluate the empirical performance of the two models, we use spoken-recording experiments and speech augmentation independently. The TT-DNN model achieves a worse MAE score despite having significantly fewer model parameters than the DNN model (0.604Mb vs. 30.425Mb) (0.664 vs. 0.675) With better PESQ and STOI scores, the hybrid model CNN+DNN may significantly increase the DNN base. The performance of two innovative TT models—LR-TT DNN and CNN-TT DNN—that were created in this study was evaluated using SCR and voice enhancement tasks.

Chapter 3

Background Analysis

3.1 CNN Algorithm

One type of Deep Learning algorithm is the Convolutional Neural Network (CNN), which accepts input, prioritizes different input features by adding learnable weights and biases, and can discriminate between them. [5] Along with other layers that feature several filters or kernels that carry out various operations, it also has some convolutional and pooling layers.

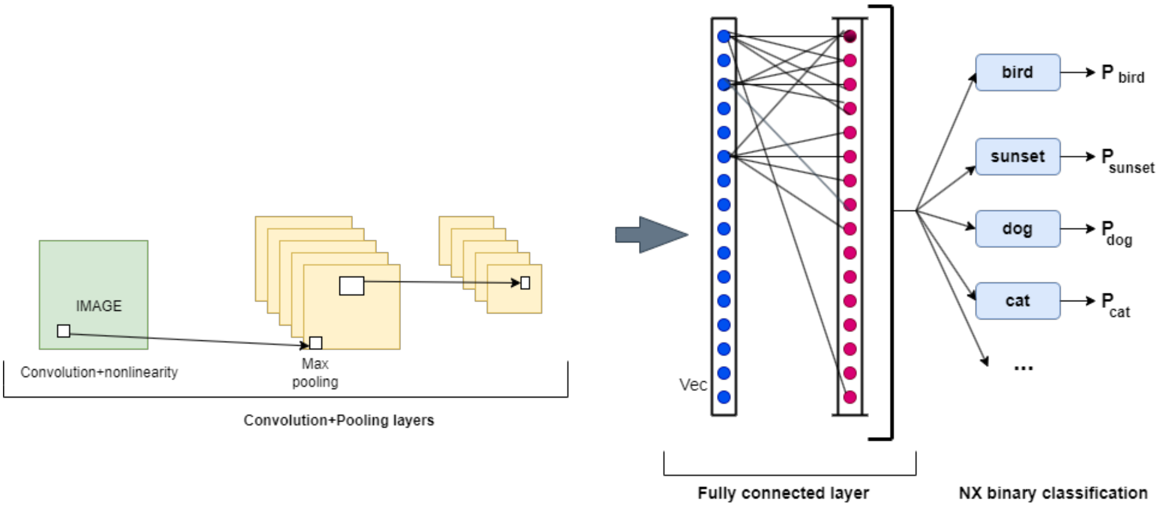

Figure 3.1: CNN Architecture

By applying a filter or kernel to the input, the convolution layer collects the features from the input for classification purposes. These convolutional layers consist of a number of layers, each of which creates a number of activation functions, outputs the activation value, and passes the output on to the next layer after receiving the weighted total of all of the inputs.

The first layer collects fundamental information and sends the output to the second layer, which finds more intricate features. As the number of layers rises, so does the complexity.
The pooling layer is used to reduce the computing resources required for data processing by lowering the input dimensions with the use of filters and kernels after the

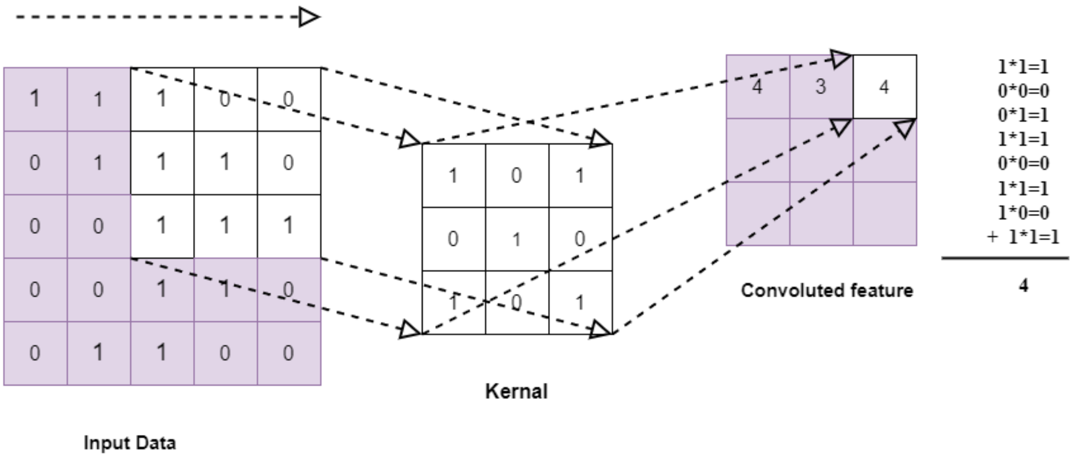

Figure 3.2: Convoluting characteristics from input data

convolution layer. Average pooling and maximal pooling are two separate categories of pooling algorithms. The maximum value is chosen in maximum pooling. from a portion of the input that the kernel's dimension covers.

Because it can eliminate noisy activations, conduct de-noising, and reduce input dimension, max pooling may be thought of as a noise limiter. [12] The mean value of all the values from a certain subset of the input that is covered by the filter's dimension is what the average pool returns. Max pooling is a superior pooling approach to Average pooling since Average pooling can only reduce the dimension of the input for noise reduction.

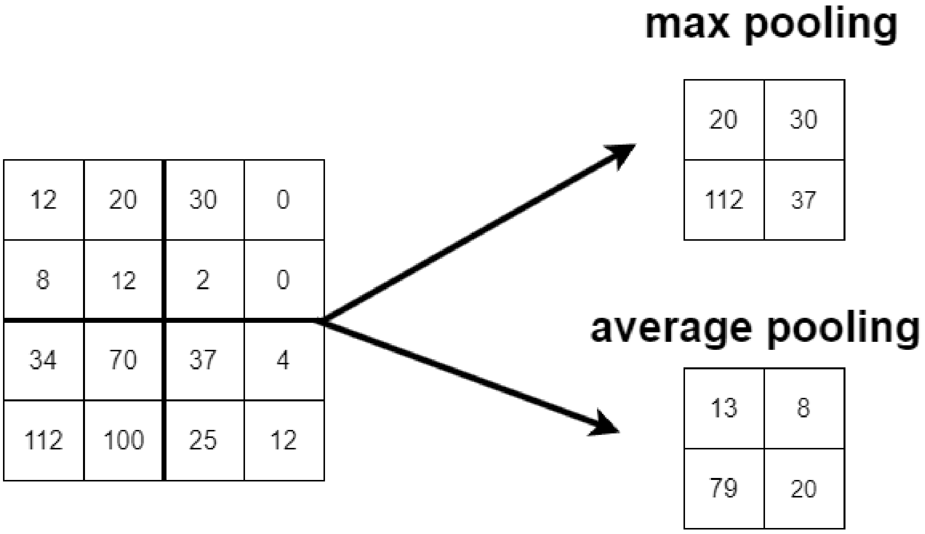

Figure 3.3: Particular pooling methods

Multiple iterations of the convolution and pooling layer can flatten the input so that it contains only the information needed for classification. The input is then passed through a fully connected feedforward network that classifies the input and produces output in accordance with that classification.

CNN can find important properties without the assistance of a person.[1] CNN also needs a lot less preparation than any other classification technique. CNN is useful not just for image recognition but also for text, sound, and voice classification, among many other fields.

3.1.1 Conv 2D

One-dimensional (1D) convolution, often known as convolution, is a type of convolution method. Convolution between two signals that span and have a mutual perpendicular dimension is known as two-dimensional convolution. [4] A multidimensional (2D) convolution process is referred to as a convolution 2D process. By multiplying and accumulating the current values of overlapping samples that correspond to the 2D inputs, it is done. Here, one of the inputs is inverted twice, and the segmentation map is predicted for a particular area or subset of the input using 2D convolution kernels. One of the most popular varieties of convolution layer is convolution 2D.

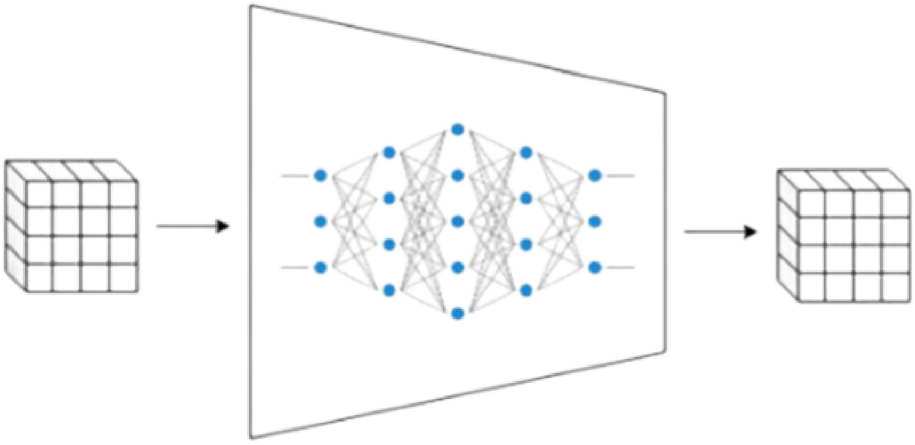

Figure 3.4: Conv 2D network with 2D input and 2D output

Depending on the filter being used to process the input in the convolutional layer, the kernel may vary. [14] To extract features from the input, the kernel slides in two directions. The weighted sums of the characteristics are taken from the inputs and displayed in the output. The amount of input features that are merged to create a new output feature depends on the size of the kernel. [7]

The kernel may gather the original edge values to be in the center by using padding to add extra values to the inputs, and stride extends the padded values beyond the edge to provide an output with the same dimensions as the input.
Striding makes outputs smaller in size than the input.

Since some of the kernel's slide sections are omitted in this case, every slide now functions as a regular convolution process, which will reduce the output.

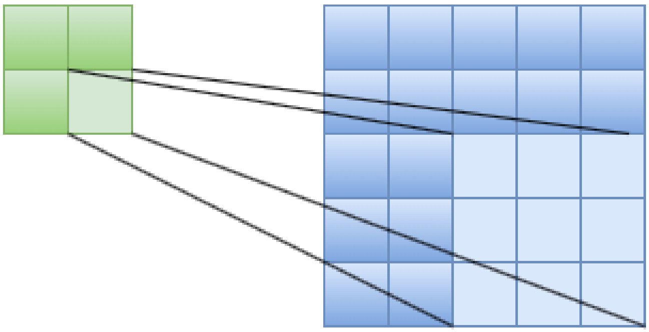

Figure 3.5: Applying Padding to the input

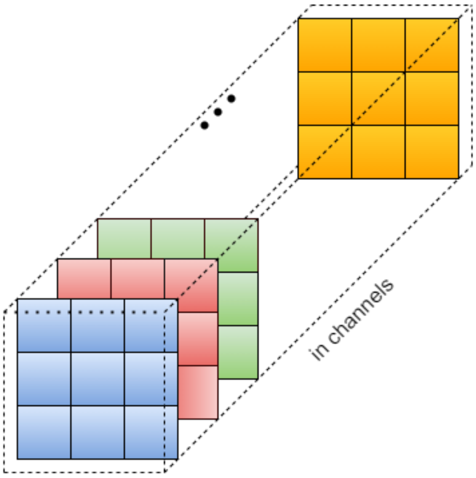

Figure 3.6: Set of kernels

Additionally, filters with a group of kernels are used for multi-channel inputs; one kernel is applied to the layer for each individual input channel, and each kernel is distinct from the others.

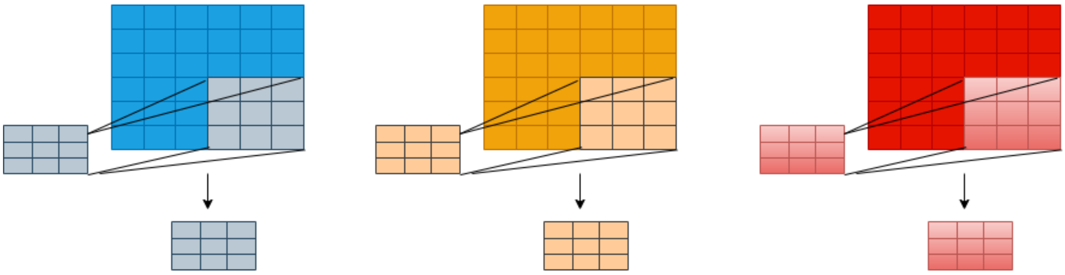

Figure 3.7: Kernel slides over respective input to produce a processed version of input

Each of these kernel-level filters moves over its corresponding channel to create a processed version of each input, which is then merged to create a single overall output channel.

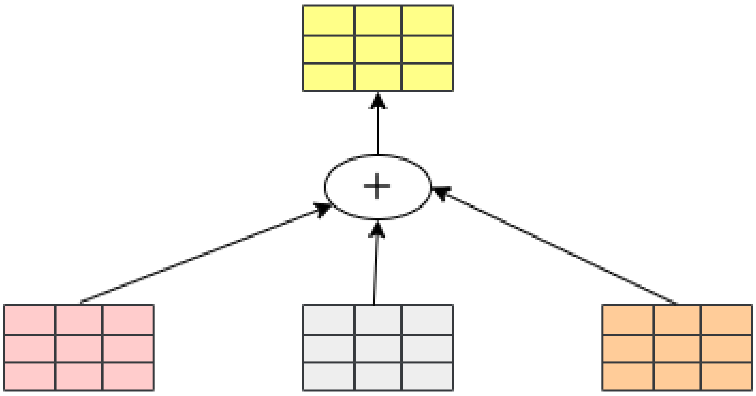

Figure 3.8: Adding all the processed version of input to generate one output

Each output filter has one bias and processes inputs using a distinct set of kernels, therefore the bias is added to the output channel before being combined to create the final output channel.

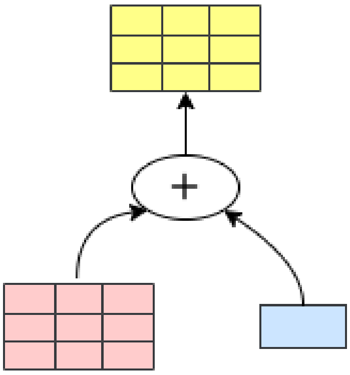

Figure 3.9: Adding bias to the output

Digital image classification, image processing, object identification, and several other procedures including video and audio classification may all benefit from the usage of convolution 2D.

3.1.2 Max Pool

Convolutional neural networks (CNN) frequently use the max pooling technique to downscale the input's spatial dimensions while preserving crucial data. In other words, it decreases the input's spatial resolution while maintaining the most crucial properties.

Typically, max pooling is used after a CNN's convolutional layers. It operates by splitting the input into a number of rectangular, non-overlapping areas, and then outputs the maximum value for each zone. The stride (i.e., the step size) is often chosen to be the same as the size of the regions, which is typically intended to be tiny, such as 2x2 or 3x3.

The primary goal of max pooling is to decrease the input's dimensionality while retaining the most crucial information. By doing this, it increases the CNN's resistance to slight distortions and translations of the input and lowers the amount of computation needed for the network's subsequent layers.

3.1.3 Batch Normalization

Batch normalization is a method for resolving the issue of internal covariate shift. The input "features," which are frequently represented by the letter X, are known as covariates. Covariate shift, which shows how the distribution of the features is different in different regions of the training/test data, breaks the i.i.d assumption, which is used throughout most ML. The term "covariate shift" refers to internal covariate shift, such as switching from layer 2 to layer 3 within a neural network. This occurs because the distribution of outputs from a particular network layer varies as the network learns and the weights are changed. Learning is slowed slower as a result of having to adapt to that drift by higher levels.

3.1.4 Dropout

The Dropout layer acts as a mask, preserving the viability of all other neurons but excluding some neuron's contributions to the layer above it. If we apply a Dropout layer, characteristics are eliminated; however, if we apply it to a hidden layer, some hidden neurons are lost.

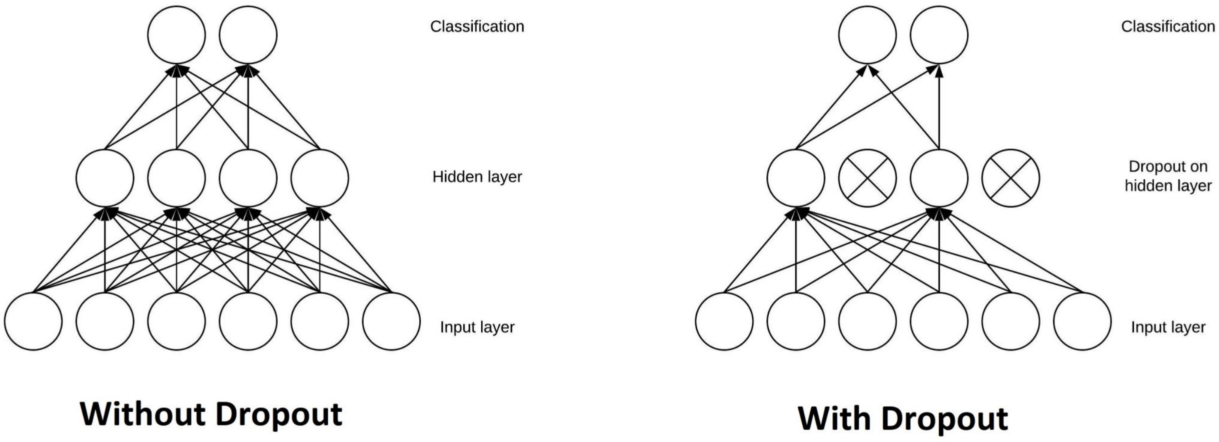

Figure 3.3: Dropout

Dropout layers are essential in the training of CNNs because they prevent overfitting the training data. The initial batch of training examples has an overly big influence on learning if they aren't there. This would prevent features from being learned that appear in later samples or batches. Ten photos of a circle from CNN are shown one after the other. CNN will be confused if we subsequently show it an image of a rectangle since it doesn't know that straight lines may exist. We can prevent these problems by introducing Dropout layers into the network's architecture to reduce overfitting..

3.2 ResNet50

Deep convolutional neural network design called ResNet-50, which contains 50 layers, is typically used for image classification tasks like the ImageNet challenge. It starts with a number of convolutional layer blocks, such as identity, bottleneck, and convolutional layers, and is then followed by a pooling layer for the global average and a fully connected layer for classification. After demonstrating state-of-the-art performance on a variety of computer vision applications, the ResNet-50 architecture gained widespread acceptance in both academia and business.

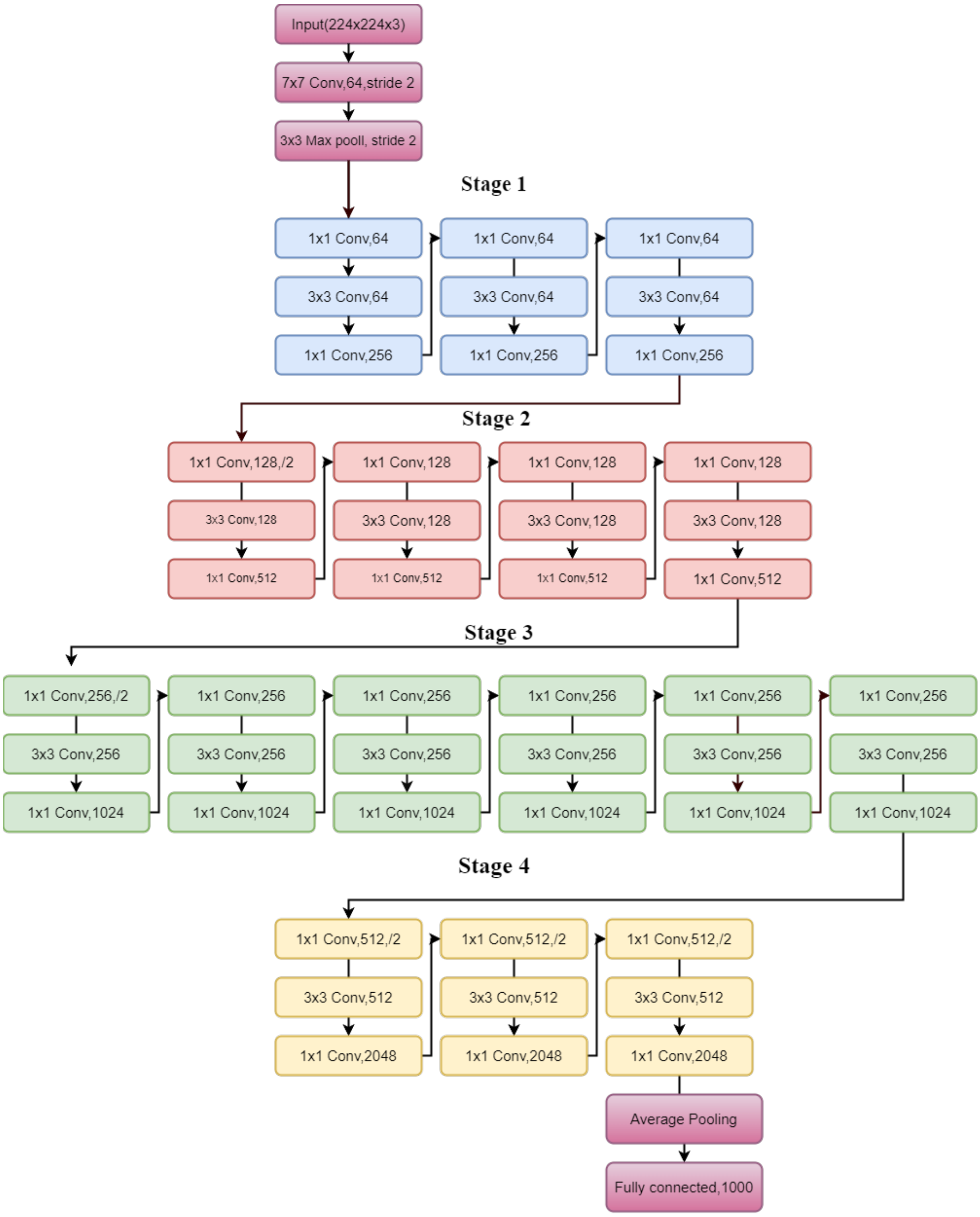

Figure 3.10: Architecture of ResNet-50

The architecture of ResNet-50 can be divided into four main blocks: the convolutional block, the bottleneck block, the identity block, and the pooling layer. The convolutional block is the first block of the ResNet-50 architecture. It consists of a convolutional layer followed by a batch normalization layer and a ReLU activation function. The purpose of this block is to extract features from the input image.

The bottleneck block is the second block of the ResNet-50 architecture. It consists of three convolutional layers with a 1x1, 3x3, and 1x1 kernel size. [2] These layers are followed by batch normalization and ReLU activation functions. The purpose of this block is to reduce the number of parameters in the model while still maintaining high accuracy.

The identity block is the third block of the ResNet-50 architecture. It consists of three convolutional layers, with the first and last layers having a 1x1 kernel size and the middle layer having a 3x3 kernel size. These layers are followed by batch normalization and ReLU activation functions. The purpose of this block is to learn the residual mapping between the input and output feature maps.The pooling layer is the final layer of the ResNet-50 architecture. It is a global average pooling layer that averages the spatial dimensions of the feature maps. The output of the pooling layer is then passed through a fully connected layer and a softmax activation function to produce the final output classification.

The ResNet50 concept is broken down into four steps as seen in the following diagram. A image with height, breadth, and channel widths that are multiples of 32 may be accepted by the network.To make things clear, we'll assume the input size is 224 × 224 x 3. Every ResNet design uses 77 and 33 kernel sizes for the initial convolution and max-pooling operations, respectively. The network then enters Stage 1, which consists of 3 Residual blocks with a total of 6 layers. The three levels of the stage 1 block's convolution process employ kernels that are 64, 64, and 128 bits in size, respectively. The identity relationship is represented by curved arrows.The dashed connecting arrow indicates that stride 2 is used for the convolution operation in the residual block; as a result, the input's height and breadth will be cut in half while its channel width will be doubled. As we move through the phases, the input size is halved and the channel width is doubled.

3.3 VGG-16

The VGG16 architecture has 16 layers, comprising 3 fully linked levels and 13 convolutional layers. A max pooling layer and a rectified linear unit (ReLU) activation function are placed after each convolutional layer. Following the completely linked layers, a softmax activation function generates the final output for classification.

A succession of convolutional layers, max pooling layers, and eventually fully linked layers make up the VGG-16 architecture. The max pooling layers downsample the feature maps to minimise the spatial dimensionality of the data, while the convolutional layers are in charge of extracting features from the input picture. The final categorization of the input picture into one of the potential classes is carried out by the fully linked layers.

The VGG16 architecture is composed of 13 convolutional layers and 3 fully connected layers. The first two layers are 3x3 convolutional layers with 64 filters, followed by a max pooling layer. The next two layers are 3x3 convolutional layers with 128 filters, followed by another max pooling layer. The pattern continues with two 3x3 convolutional layers with 256 filters, two 3x3 convolutional layers with 512 filters, and two 3x3 convolutional layers with 512 filters, each followed by a max pooling layer. The fully connected layers at the end of the network have 4096 neurons each, and the final output layer has the number of neurons corresponding to the number of classes in the dataset being used.

Layer	Patch size	Input size
convx2	3x3/1	3x224x224
pool	2x2	64x224x224
convx2	3x3/1	64x224x224
pool	2x2	128x112x112
convx3	3x3/1	128x56x56
pool	2x2	256x28x28
convx3	3x3/1	512x28x28
pool	2x2	512x28x28
convx3	3x3/1	512x14x14
pool	2x2	512x14x14
fc	25088x4096	25088
fc	4096x4096	4096

Table 3.1: Architecture of VGG-16 [3]

One of the strengths of the VGG-16 architecture is its simplicity and uniformity, making it easy to understand and implement. The convolutional layers have a small receptive field (3x3) and are stacked one after the other, resulting in a very deep network that can capture a wide range of image features.

Chapter 4

Implementation

4.1 Model Implementation and Optimization

We will discuss our overall system workflow and all of the methods we used to conduct our study in detail in this chapter. We will go over every step we took to do the investigation and research for our report. Data can be preprocessed in a number of different ways, and our accuracy varies depending on the preprocessing. We want people to be able to use audio devices more precisely by using our model.

Our device will be really simple to use, allowing customers to carry it around with them and use it as needed.

In our article, sound recognition functions in a way that makes it possible for our model to quickly recognize the ambient noise in a room and provide it to users in accordance. Our model will preprocess each sound that it receives, extract the feature later, and compare the feature to the sound in our database to determine the sound properly.

4.1.1 Dataset

We wanted to create our own data set for our paper, but it was practically difficult for us to get primary data owing to the lack of access to devices. The Speech Command Audio Dataset V02 [23] was used, and we made every effort to locate audio files online. These items are all tuples made up of the following components: label, speaker ID, waveform, sample rate, and utterance number. The waveform and label will be used to train our model, and the sample rate will be used to make the data easier to interpret.

We previously had trouble locating the data collection we were seeking when searching online. We required data collection for our thesis that will include several types of extracted sound samples saved in it. There were several data sets that included voice, but we didn't require it for our project at the time. However, in the future, we may start translating speech to text so that those who can hardly hear without a hearing aid may use our program to connect with others.

We used an additional data set. 37 categories of sounds from daily life have been discovered. We've divided them into groups and compiled the many categories. putting the sound clip for preprocessing in one location. We have chosen a variety of categories for the sound clips, including happy, house, off, background noise, eight, seven, bed, six, forward, go, nine, learn, yes, up, dog, stop, cat, and others.

The audio file we used was a wav file. We removed any extraneous sound from the audio files and divided the audio files into 1-second chunks with precise sound. Our sound files are edited using librosa, a library that the Python programming language may import. Librosa's AudioSegment was used to convert the wav file to the visual format. The files were converted to wav format using AudioSegment's extract tool.

The sample rate from the audio data collected in the downsampling follows the default sample rate of the librosa program, which we utilized as a Python tool to analyze audio. Later, we remove any files that are not necessary for our content. The features were then retrieved after the splitting, and they were then put in our data set.

Our data set was evenly distributed, with each category taking up around the same amount of space in our file. Which was advantageous as our data set was not skewed. As we utilize the data set to train and test our model, it is quite important for a machine learning project. Errors, flaws, or biases in the data collection cause serious issues for the overall project since the computed accuracy may not be totally accurate, which makes the model behave incorrectly. .

Below there are some waveplot for different sound types-

4.2 Data Preprocessing

Before using the data set, data preparation is a crucial step. The process of converting an initial data set into a clean data set is known as data preparation. Data preprocessing is done to remove extraneous data columns from the dataset. Since some data may contain any valid information at all, it is important to repair or remove these data before using them as they may provide problems in the future.

We had to gather various background noises because we were gathering our own data collection. There are audio snippets of animals like dogs, cats, birds, numbers, and more. After gathering all of our sound samples, we used MFCC to extract each feature. After keeping all of our audio samples in a list, we extracted the data and put it into a CSV file. Each piece of data was given the appropriate label.

The Python library Librosa is used to examine audio and music files. The primary functions of librosa include automatic voice recognition and music production. The time series that is delivered by librosa.load is known as a "time series" in the librosa glossary. We used librosa.load with a three-second time. Please take note that the sampling rate of our audio files is 16000 Hz.

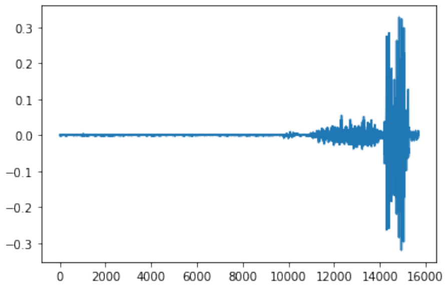

Figure 4.1: Waveplot for happy Sound

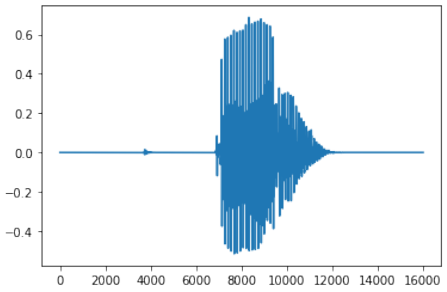

Figure 4.2: Waveplot for On Sound

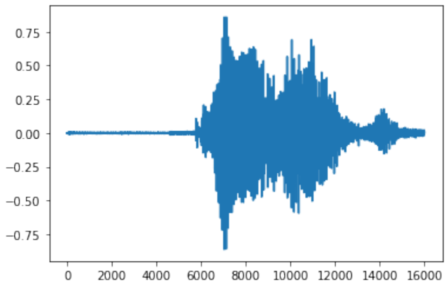

Figure 4.3: Waveplot for Dogs Sound

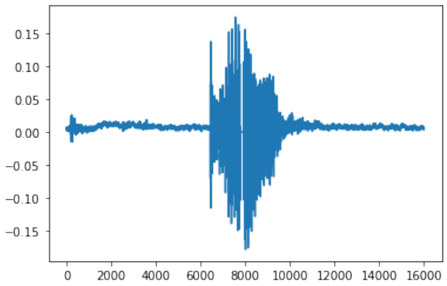

Figure 4.4: Waveplot for Go Sound

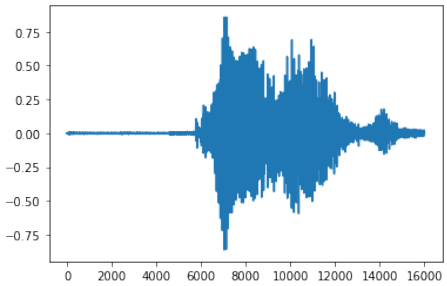

Figure 4.5:
Waveplot for Dogs Sound

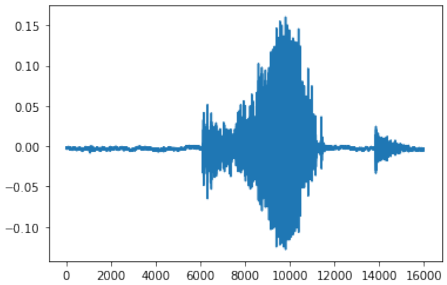

Figure 4.6: Waveplot for Cat Sound

Figure 4.7: Waveplot for different sounds

4.3 Features of Sound

The features of a sound make it apart from other classes that are shown in our data collection. Each class can be distinguished from other classes by using a specific characteristic.

In our instance, we used the MFCC to extract our features, and after that, the data was saved in the data set so that the subsequent process could occur.

4.4 Feature Extraction

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

A well-known method for extracting speech characteristics, the Mel-Frequency Cepstral Coefficients (MFCC) feature extraction method, is currently being looked into for potential performance improvements. The Delta-Delta MFCC, which enhances speaker verification, is one of the newest MFCC implementations. The distributed MFCC feature extraction approach presented in this paper is new and is based on the discrete cosine transform (DCT-II). The three alternative feature extraction techniques used in the proposed speaker verification tests are distributed DCT-II based Delta-Delta MFCC, standard MFCC, and standard MFCC plus a Gaussian Mixture Model (GMM) classifier.

4.4.1 MFCC

Cepstral Coefficients for Mel Frequencies (MFCC), The MFCC are the inverse-fft of the log of the frequency-warped spectrum, and the Cepstral coefficients are the inverse-fft of the log of the spectrum. In order to interpret all the information included in speech signals, the goal of MFCC is to transfer audio from the time domain into the frequency domain. By utilizing Mel filters to simulate the cochlea's function, MFCC transforms time-domain signals into frequency-domain signals.

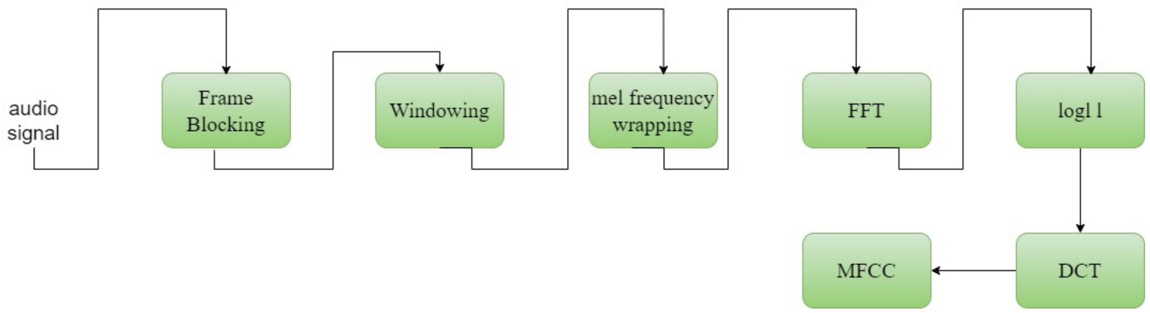

Figure 4.8: Figure of MFCC

4.5 Feature Selection and Engineering

4.5.1 Feature selection

We employ feature selection to enhance the efficiency of our model and decrease computation time and cost. In order for our model to quickly compute the data, we reduce the number of input variables during feature selection. feature choice may aid in data interpretation and visualization, and while resolving the issue of multidimensionality to improve the performance of our model, it may also shorten utilization times, training times, and storage requirements.

The feature we choose for our model is audio file data, while the remaining classes are retained in labels.

4.5.2 Feature Engineering

All machine learning algorithms require some input data, in order to generate outputs. Feature engineering, the act of developing new input features for machine learning, is an efficient approach to enhancing prediction models. One of the primary aims of predictive modeling is to discover an effective and accurate predictive connection between a collection of accessible data and an outcome, such as the possibility of a customer doing a particular action. Choosing and manipulating variables is the process of feature engineering when using machine learning to build a predictive model. It is a great strategy to enhance prediction models since it involves gathering crucial data, stressing patterns, and bringing in someone with domain expertise.

4.6 Train-Test split

Finding out how well our algorithm works on a certain data set is the key goal of the train-test split. We primarily divide our data set into two pieces, with the first half being utilized for training. We typically place more than half of the rows as being the most important aspect of a system, a data collection should be used in the training phase. Our model will be able to recognize our target more readily if it is trained with more accurate data. The remaining data will be utilized to test our model, which is how we truly determine how well it works. Consequently, we may state that we shall divide our data set into two parts

i Testing dataset.
ii Training dataset.

Our data set was divided using scikit-train learn's test split() function using the industry-standard 4:1 split ratio. The ratio, which in our instance is 4:1, will be used to divide both our features and labels.

4.7 Models

CNN is the model we have used. The specific model utilized is explained in detail below. We modified the hyperparameters for each model separately.

4.7.1 CNN

In our project, we utilized CNN Conv2D. Using Python, the data collection was divided into features and given the appropriate labels. So that the algorithm could use the model as intended, the label was then encoded.

We divided our data set into four equal halves in order to test our model. Our model utilized the activation function of ReLU and included several hidden layers. The piecewise linear function known as ReLU—or rectified linear unit—gives the output as input if the input is positive and produces zero when it is not. Information is lost because ReLU ignores negative values and produces zero if the input is negative. ReLU does not have any vanishing gradients, and it is also more computationally efficient than the Sigmoid activation function, which is the major reason we chose to use it.

Since softmax functions better with output activation, this was done. Issues with multiclass categorization.

Our model contained categorical cross entropy as a loss function. Categorical cross entropy serves as a loss function in situations involving multi-class categorization. We assembled our model after fitting the training data set into it, then ran it for 40 epochs.

4.7.2 ResNet50

For Resnet50 we Split out the dataset into an 80:20 ratio. We have used Residual blocks. Each residual block consists of two convolutional layers, each followed by a batch normalization layer and a ReLU activation function. The output of the second convolutional layer is added to the input of the residual block before the ReLU activation function. An average-sized pooling layer follows the remaining blocks a stride of 1 and a 7 × 7 grid. This reduces the spatial dimensions of the output feature maps. The use of residual connections in the residual blocks allows for the gradients to flow through the network more easily, leading to faster convergence and better performance. After incorporating the training data set into our model, we put it together and ran it for 20 epochs.

4.7.3 VGG16

Similar to cnn model we have used dataset split in our VGG16 model. We split our dataset into 80% for training and 20% for testing.VGG16 model consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. There are 13 convolutional layers, each followed by a ReLU activation function. The first layer has 64 filters of size 3 x 3, and the subsequent layers have 128, 256, and 512 filters of size 3 x 3. The final two convolutional layers have 512 filters of size 3 x 3. Although we didn't utilize any activation functions during the procedure, we did apply softmax for the result. Above other activation functions, the key benefit of softmax is the breadth of output options. Softmax is employed because the activation function enables us to calculate the probabilities of each class while maintaining a high chance for the target class. Dropout is typically added to a model to stop it from overfitting which has a dropout rate of 0.2. We assembled our model after fitting the training data set into it, then ran it for 20 epochs.

Chapter 5

Results and Discussion

One of the most significant ways that we use AI every day is with audio classification. We automatically call the right person (whether in a busy office or a crowded store checkout) to handle an incoming phone call or text message, or send appropriate alerts to security personnel so they can reach our location faster. Once you understand how it's done, you can apply this technology in your own organization to increase efficiency and productivity among employees.

All of the models utilized in this study were trained using Google Colaboratory.

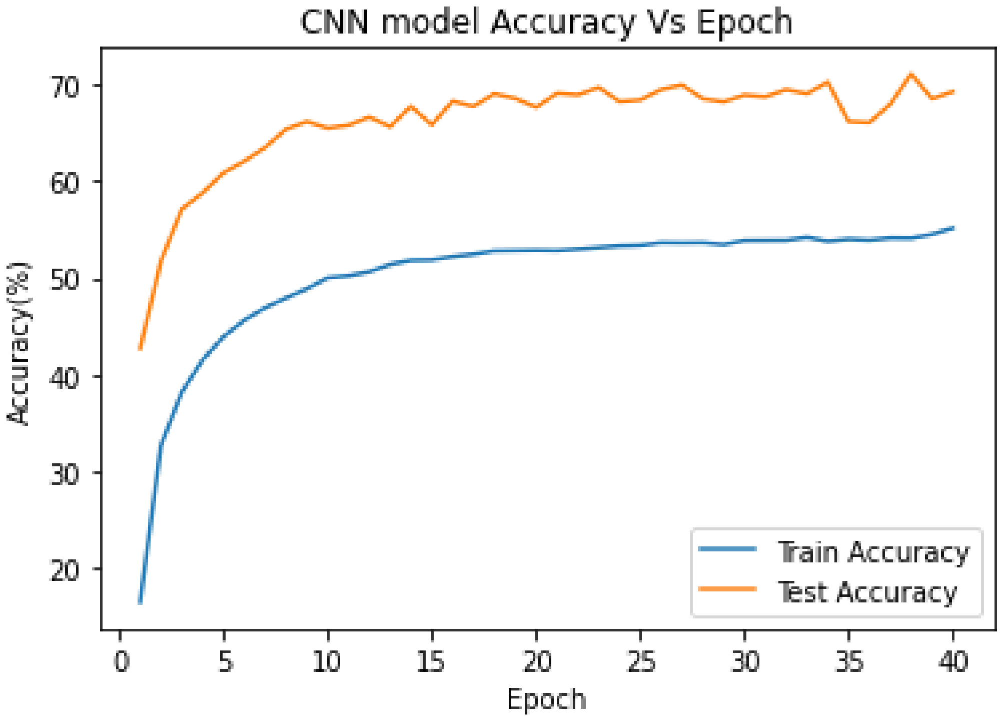

Figure 5.1: CNN Accuracy vs epoch

The complete Jupyter notebook is connected with Google Drive in Google Colaboratory, a cloud-based Jupyter notebook environment. For this experiment, we used CNN, ResNet50 and vgg16 models, and the results were quite similar for CNN and vgg16 but the ResNet50 model had better result.. The bar chart (Figure) shows that CNN had an accuracy of 69.29% after 40 epochs. For ResNet50 and vgg16 we found our required accuracy after 20 epochs. Which are 94.53% and 76.64%

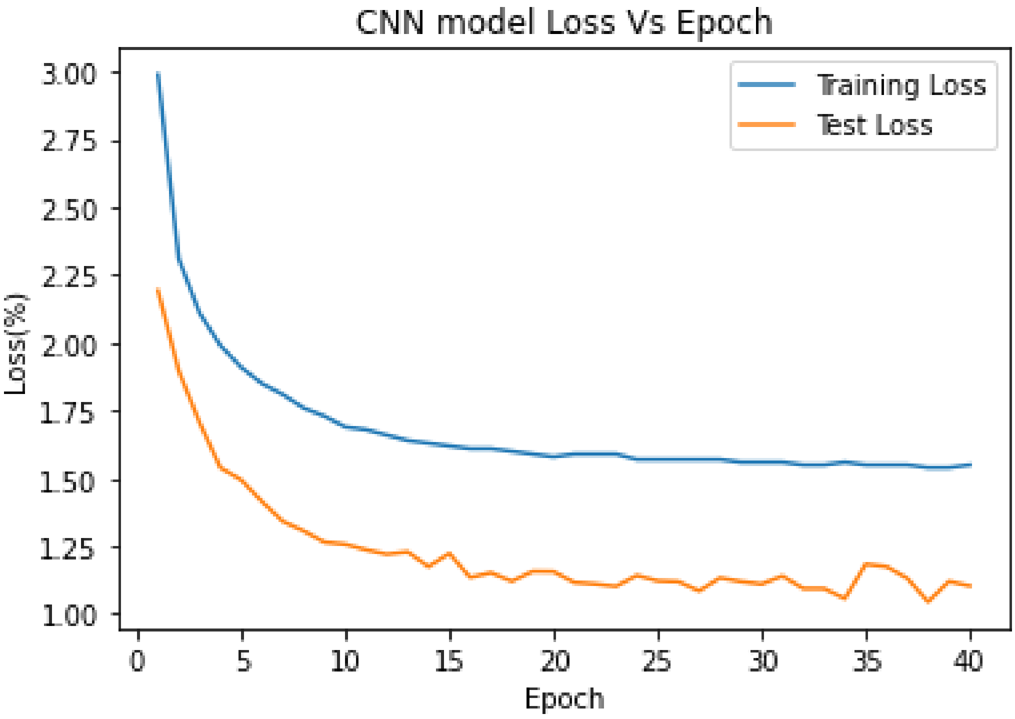

Figure 5.2: CNN LOSS vs epoch

respectively.We observed that MFCC took around 1:45 hours for CNN model, and for vgg16 it took 6:25 hours. On the other hand ResNet50 it took a mammoth 11 hours to train. CNN feature extraction went through 40 epochs. ResNet50 and vgg16 had 20 epochs each.

For MFCC data. When it was running on CNN, the loss value was very large and exceeded 1. However, when the number of epochs increased, the loss value likewise decreased.

But we discovered that the value was frequently changing. That indicates that the value occasionally increased while occasionally falling. But it dropped below 2.193 after 40 epoch. Additionally, the accuracy score was 69.29% which was rising at the same time. While the MFCC was running on vgg16, the loss value was above 2 but it decreased as the epochs started increasing. Here we saw that the values were decreasing as the epochs went by, while the accuracy was increasing at a decent rate.

Moreover in ResNet50 when the MFCC was running the loss value was also above 2 and it also started dropping as the epochs progressed. From the very beginning we saw a decent accuracy which later became our highest among all the models we trained. So, after 20 epochs we got 94.53% which was still rising.

We used pre-trained vgg16 and ResNet50 models for our tests. Initially, when we finished testing our software, we obtained an accuracy of 94.53%; which was according to our expectations however, We analyzed each audio file to determine how

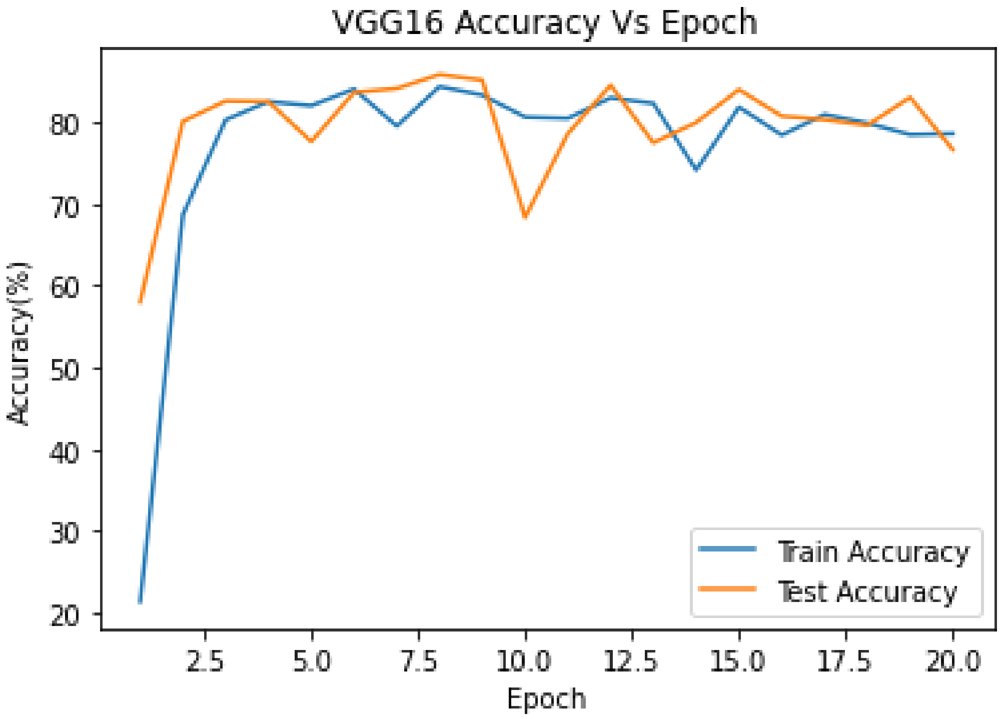

Figure 5.3: VGG-16 Accuracy vs epoch

many empty sounds it contained, resized the audio files appropriately, and removed any files with no audio from our data set. We then added new audio files to replace the ones we deleted, and that's how we were able to get our accuracy to 94.53%.

We used pre-trained vgg16 and ResNet50 models for our tests. Initially, when we finished testing our softwares, we obtained an accuracy of 94.53%; which was according to our expectations, however, We analyzed each audio file to determine how many empty sounds it contained, resized the audio files appropriately, and removed any files with no audio from our data set. We then added new audio files to replace the ones we deleted, and that's how we were able to get our accuracy to 94.53%.

We've included a table for the classification report for CNN, ResNet50, and vgg16. The classification report includes precision, accuracy, recall, and f1-score. The algorithm's capacity to determine how many of the total data points are really correctly calculated by the model is known as precision. It is common to display the ratio between the total number of data that were calculated and the total number of data included in the data collection. The recall is the capacity of a model to correctly identify all of its positive examples. The f1-score is used to compute the harmonic mean of accuracy and recall. In relation to all other classes, the scores for each class show how well the classifier classified the data points in that class. The evidence is provided by how many actual answer samples fit into that category. The amount of accurate estimates made by our model is referred to as accuracy. The macro average represents the mean average of all the values we have discovered. Less of one class suggests that its accuracy, recall, or F1 score has a lesser impact on the weighted average for each of those items since the weighted average takes into consideration how many of each class were utilized in the computation.

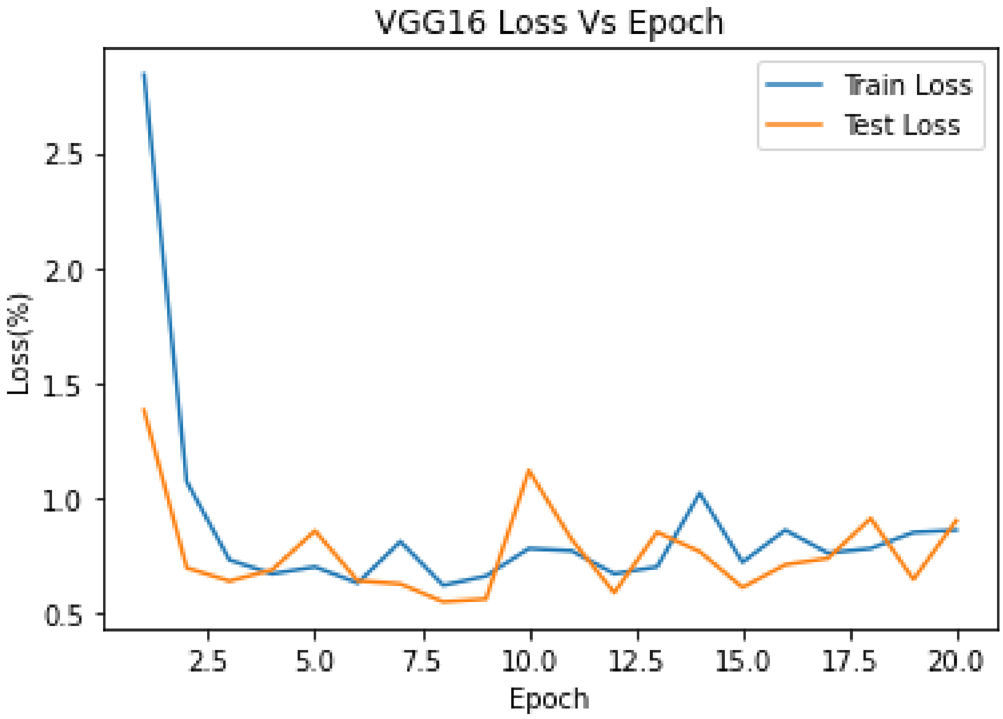

Figure 5.4: VGG-16 LOSS vs epoch

According to earlier research for audio categorization that was conducted utilizing several feature extraction techniques. We came to the conclusion that ResNet50 can be a very good option for audio categorization based on the outcomes of the experiment and research. According to the findings of our research, a model like ResNet50 may be used to classify audio with an accuracy of 94.53% when utilizing MFCC data. Our study goal was to develop a better feature extraction technique for audio-detecting systems that may aid people. Because our system can accurately identify a variety of noises in our surroundings, it is a reliable alternative for other applications and can assist those in need.

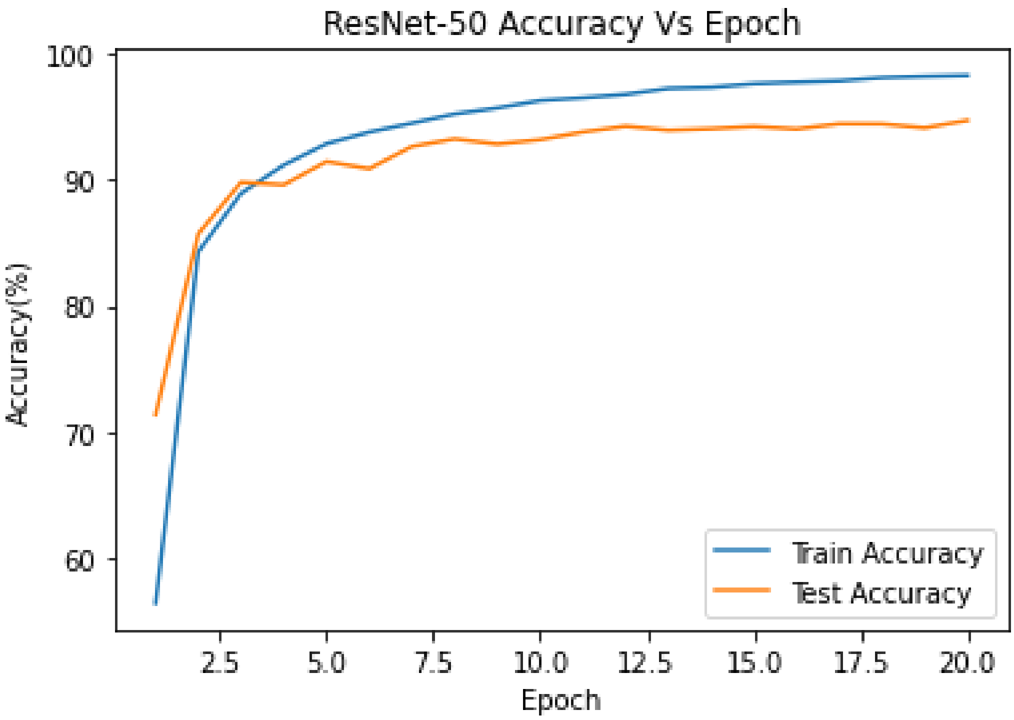

Figure 5.5: ResNet-50 Accuracy vs epoch

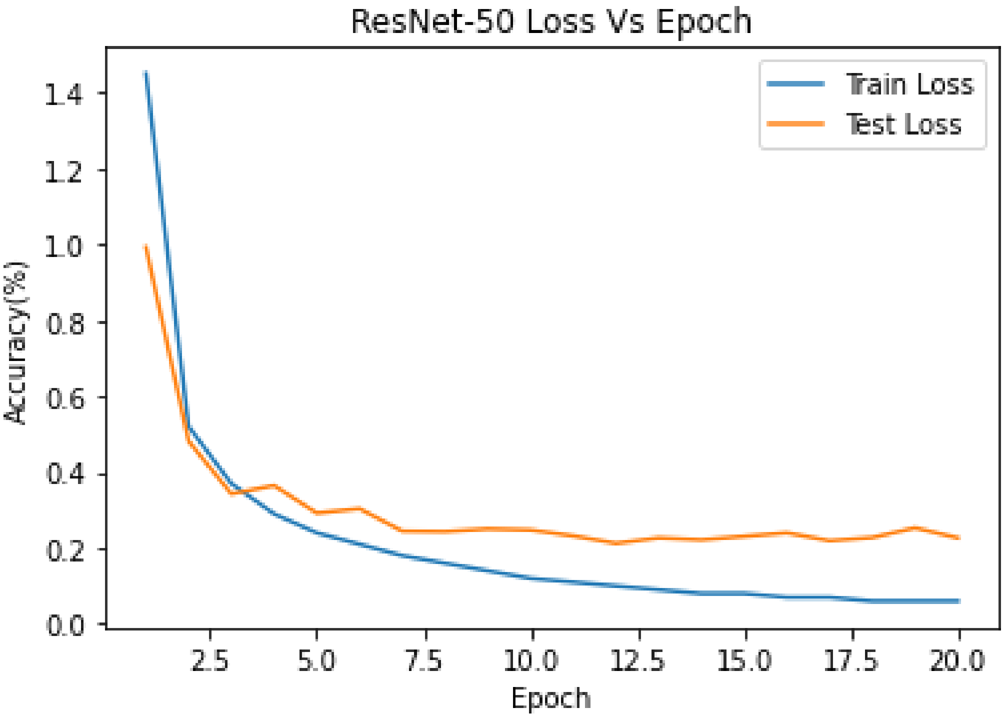

Figure 5.6: ResNet-50 LOSS vs epoch

Class	Precision	Recall	f1-Score	Support
backward	0.946	0.903	0.924	712
bed	0.969	0.932	0.950	705
bird	0.977	0.968	0.972	308
cat	0.945	0.884	0.914	329
dog	0.886	0.886	0.886	290
down	0.901	0.928	0.914	362
eight	0.912	0.958	0.934	357
five	0.972	0.972	0.972	721
follow	0.978	0.958	0.968	692
forward	0.950	0.912	0.930	704
four	0.969	0.976	0.973	707
go	0.912	0.960	0.935	645
happy	0.921	0.882	0.901	625
house	0.994	0.906	0.948	352
learn	0.909	0.965	0.936	695
left	0.940	0.980	0.960	707
marvin	0.981	0.971	0.976	752
nine	0.891	0.959	0.924	708
no	0.889	0.944	0.916	340
off	0.912	0.934	0.923	692
on	0.968	0.866	0.914	382
one	0.966	0.979	0.973	706
right	0.935	0.908	0.921	315
seven	0.981	0.903	0.940	673
sheila	0.959	0.959	0.959	709
six	0.956	0.966	0.961	358
stop	0.966	0.953	0.959	739
three	0.955	0.892	0.922	286
tree	0.956	0.976	0.966	334
two	0.928	0.973	0.950	706
up	0.956	0.982	0.969	738
visual	0.954	0.959	0.957	368
wow	0.953	0.967	0.960	674
yes	0.983	0.962	0.972	367
zero	0.868	0.900	0.884	321

Table 5.1: Classification Report of ResNet-50

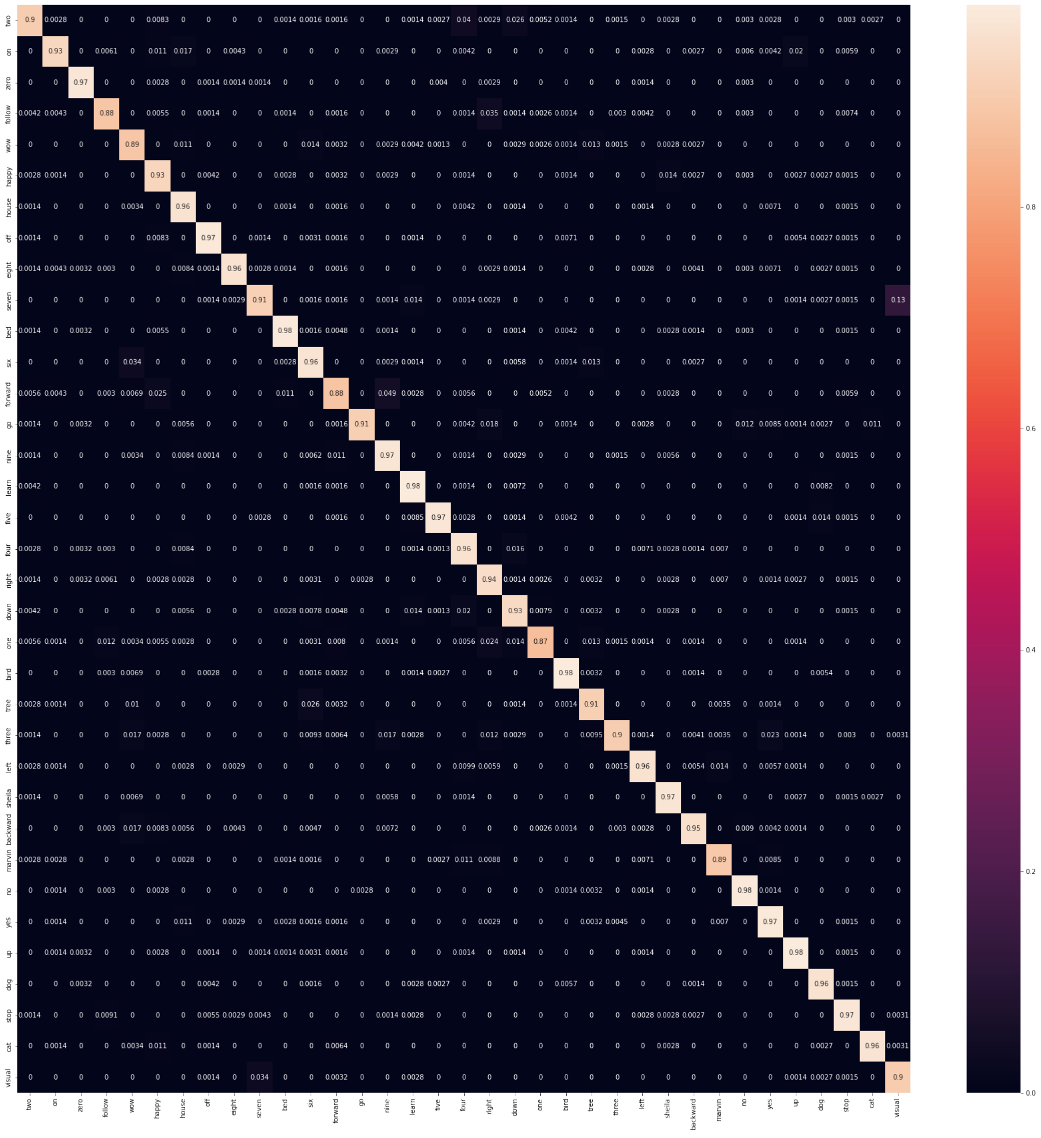

Figure 5.7: Confusion matrix of ResNet-50

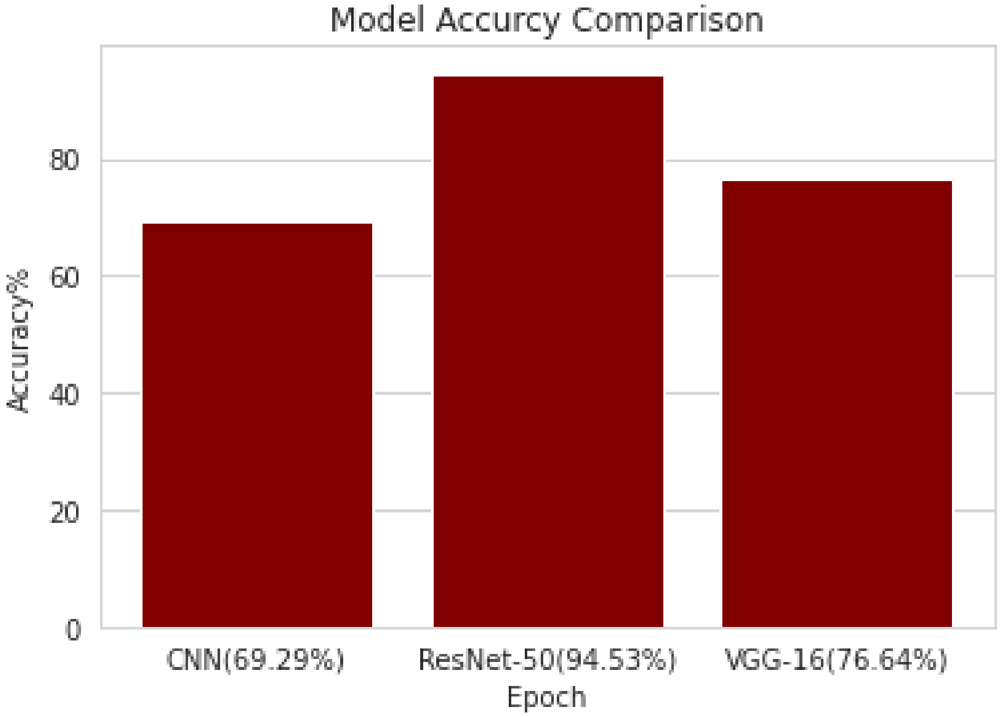

Figure 5.8: Model Comparison

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The major goal of our research was to develop a model that individuals may use in their daily lives to correctly discriminate between sounds and be alerted anytime a sound occurs nearby. The approach will assist individuals in making good distinctions, which will ultimately assist them in making wise decisions and regaining their full participation in society. With the aid of our model, deaf individuals will be better able to comprehend their surroundings and the environment in which they live. We have created our own Python code and collected our data set to input into the algorithm we employed. To divide and cut our audio files, we utilized PyTorch, a Python computer language package. Later, after gathering our data set, we used the Python library librosa's melspectrogram and MFCC to extract our features. The CNN algorithm was then applied to our project. Finally, utilizing CNN and our model, we were able to achieve an accuracy of 69.29%.

Bangladesh, a developing nation, has had strong GDP development over the previous ten years. In fact, Bangladesh successfully achieved the greatest GDP growth it has ever experienced in 2019 with a total GDP increase of 8.15% in 2019[21]. The number of people with hearing impairments was estimated to be around 13 million in 2014,[24] but that number may now have surpassed 20 million. By utilizing these individuals and enabling them to be as productive as any other normal human being, we can greatly benefit from their contributions and ensure a better future for the next generation. These 20 million individuals will have a significant beneficial impact on Bangladesh's GDP if they receive the necessary assistance. A system that is designed for people with impairments has to be constructed with a strong user interface. If our strategy is correctly implemented, it will really assist the hearing challenged in contributing to society in the same way that others do.

Audio classification is the process of automatically identifying the audio to which a particular pattern of audio data corresponds. As an example, let's suppose we are listening to a voice recording of our favorite song. we know exactly what this song sounds like, so you can instantly identify it just by hearing it. However, in order for automated systems to identify these songs automatically (or any other form of audio), additional processing is required. Processing may be done by humans or machines, but it includes building a model that represents the desired input and output data set. Calculations based on this model are subsequently used to accurately classify new recordings that have not yet been labeled (classified) with their corresponding data set identifier. Additionally, our project will be implemented in a way that will provide users with a tool that allows them to analyze sound and perform tasks properly based on that analysis.

Bibliography

[1] A. Dertat, *Applied deep learning - part 4: Convolutional neural networks*, Nov. 2017. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2.

[2] E. R. S. de Rezende, G. C. S. Ruppert, A. Theophilo, and T. Carvalho, *Exposing computer generated images by using deep convolutional neural networks*, 2017. DOI: 10.48550/ARXIV.1711.10394. [Online]. Available: https://arxiv.org/abs/1711.10394.

[3] T. L. I. Sugata and C. K. Yang, "Leaf app: Leaf recognition with deep convolutional neural networks," *IOP Conference Series: Materials Science and Engineering*, vol. 273, no. 1, p. 012 004, Nov. 2017. DOI: 10.1088/1757-899X/245/1/012004. [Online]. Available: https://dx.doi.org/10.1088/1757-899X/245/1/012004.

[4] S. H.L, *2d convolution in image processing - technical articles*, 2018. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/two-dimensional-convolution-in-image-processing/.

[5] S. Saha, "A comprehensive guide to convolutional neural networks—the eli5 way," *Towards data science*, vol. 15, 2018.

[6] E. Şaşmaz and F. B. Tek, "Animal sound classification using a convolutional neural network," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, IEEE, 2018, pp. 625–629.

[7] I. Shafkat, *Intuitively understanding convolutions for deep learning*, Jun. 2018. [Online]. Available: https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1.

[8] K. Qian, Z. Ren, F. Dong, W.-H. Lai, B. W. Schuller, and Y. Yamamoto, "Deep wavelets for heart sound classification," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2019, pp. 1–2. DOI: 10.1109/ISPACS48206.2019.8986277.

[9] J. K. Das, A. Ghosh, A. K. Pal, S. Dutta, and A. Chakrabarty, "Urban sound classification using convolutional neural network and long short term memory based on multiple features," in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, 2020, pp. 1–9. DOI: 10.1109/ICDS50568.2020.9268723.

[10] M. Ashikuzzaman, A. A. Fime, A. Aziz, and T. Tasnima, "Danger detection for women and child using audio classification and deep learning," in *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*, 2021, pp. 1–6. DOI: 10.1109/EICT54103.2021.9733601.

[11] B. Kaur and J. Singh, "Audio classification: Environmental sounds classification," working paper or preprint, Dec. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03501143.

[12] M. Mandal, *Introduction to convolutional neural networks (cnn)*, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.

[13] A. Meghanani, C. Anoop, and A. Ramakrishnan, "An exploration of log-mel spectrogram and mfcc features for alzheimer's dementia recognition from spontaneous speech," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, pp. 670–677.

[14] NamyaLG, *2-dimensional convolution*, Apr. 2021. [Online]. Available: https://medium.com/theleanprogrammer/2-dimensional-convolution-189abb174d92.

[15] M. Scarpiniti, F. Colasante, S. Di Tanna, M. Ciancia, Y.-C. Lee, and A. Uncini, "Deep belief network based audio classification for construction sites monitoring," *Expert Systems with Applications*, vol. 177, p. 114 839, 2021.

[16] K. K. Teh and H. D. Tran, "Open-set audio classification with limited training resources based on augmentation enhanced variational auto-encoder gan with detection-classification joint training.," in *Interspeech*, 2021, pp. 4169–4173.

[17] K. Banuroopa and D. Shanmuga Priyaa, "Mfcc based hybrid fingerprinting method for audio classification through lstm," *International Journal of Nonlinear Analysis and Applications*, vol. 12, no. Special Issue, pp. 2125–2136, 2022, ISSN: 2008-6822. DOI: 10.22075/ijnaa.2022.6049. eprint: https://ijnaa.semnan.ac.ir/article_6049_ef1741ff5695e864d6ad5ab7bd3161b0.pdf. [Online]. Available: https://ijnaa.semnan.ac.ir/article_6049.html.

[18] N. D. Huynh, M. R. Bouadjenek, I. Razzak, *et al.*, "Adversarial attacks on speech recognition systems for mission-critical applications: A survey," *arXiv preprint arXiv:2202.10594*, 2022.

[19] J. Qi, C.-H. H. Yang, P.-Y. Chen, and J. Tejedor, "Exploiting low-rank tensor-train deep neural networks based on riemannian gradient descent with illustrations of speech processing," *arXiv preprint arXiv:2203.06031*, 2022.

[20] M. Badgujar, A. Wagh, S. Chavan, P. Chumbhale, and R. Sonawane, "Iot based automatic door lock system by face and voice recognition,"

[21] *Bangladesh gdp growth rate2022 data - 2023 forecast - 1994-2021 historical - chart.* [Online]. Available: https://tradingeconomics.com/bangladesh/gdp-growth.

[22] C. Okafor, S. Nnebe, T. Alumona, V. Onuzuluike, and U. Jideofor, "Door access control using rfid and voice recognition system,"

[23] *Speech command classification with torchaudio.* [Online]. Available: https://pytorch.org/tutorials/intermediate/speech_command_classification_with_torchaudio_tutorial.html?fbclid=IwAR27eIlMe1CJ-Wrxsg3RJpFSFMDTzFBmgVKjhpS1_wcVjPuB6ClHFmzAasc.

[24] *Statistical yearbook for asia and the pacific 2014.* [Online]. Available: https://www.unescap.org/publications/statistical-yearbook-asia-and-pacific-2014.