

Gender Classification in Bangla Language Using Deep Learning-Based Voice Analysis

by

Talukder Juhaer Hakim
19301134

Sayema Binte Monsur
19301030

Abtahi Maskawath Shuvo
19301131

Tasmia Azrine
20301165

Md. Zarif Labib
19301165

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
Brac University
Summer 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing the degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material that has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Talukder Juhaer hakim
19301134



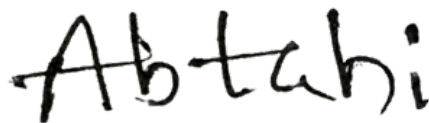
Sayema Binte Monsur
19301030



Md. Zarif Labib
19301165



Tasmia Azrine
20301165



Abtahi Maskawth Shuvo
19301131

Approval

The thesis titled “Gender Classification in Bangla Language Using Deep Learning-Based Voice Analysis” submitted by

1. Md. Zarif Labib (19301165)
2. Sayema Binte Monsur (19301030)
3. Abtahi Maskawath Shuvo (19301131)
4. Tasmia Azrine (20301165)
5. Talukdar Juhaer Hakim (19301134)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September 25, 2023.

Examining Committee:

Supervisor:

(Member)

Dr. Md. Ashraful Alam

Associate Professor
Dept. of Computer Science and Engineering
BRAC University

Program Coordinator:

(Member)

Dr. Md. Golam Rabiul Alam

Professor
Dept. of Computer Science and Engineering
BRAC University

Head of Department:

(Chair)

Dr. Sadia Hamid Kazi

Associate Professor
Dept. of Computer Science and Engineering
BRAC University

Abstract

Gender classification based on voice analysis is one of the elemental tasks in speech and audio processing, with various applications such as speech recognition systems, voice assistants, call center analytics, etc. For speech synthesis, speaker identification, and human-computer interaction- gender recognition plays a vital role. Although extensive research on this topic has been done in various languages, any studies can hardly be found regarding gender classification in the Bangla language. Our research paper aims to recognize gender in the Bangla language using deep learning approaches and voice analysis. The core of our approach involves the use of CNN models (ResNet50, EfficientNetB0, InceptionV3, and DenseNet-121) for our data training. The Mel-Frequency Cepstral Coefficients (MFCC) and short-time Fourier transforms (STFT) were computed from audio recordings and used as input features to the neural network model. The system's excellent accuracy rate demonstrates its potential for use in practical settings. By providing light on the application of deep learning techniques in the context of the Bangla language, this study advances the area of gender identification. 95% accuracy was achieved in the InceptionV3 and EfficientNetB0 models with the MFCC input.

Keywords: Deep learning; Machine Learning; F1-score; Bangla Language; Prediction; Decision tree; ResNet50; EfficientNetB0; InceptionV3; DenseNet-121; STFT; MFCC

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis has been completed without any major interruption.

Secondly, to our supervisor Dr. Md. Ashrafal Alam sir for his kind support and advice in our work. He helped us whenever we needed help.

Thirdly, all the reviews they gave helped us a lot in our later work.

And finally to our parents without their support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vi
List of Tables	viii
Nomenclature:	1
1 Introduction	2
1.1 Thoughts behind Gender Identification	2
1.2 Problem Statement	3
1.3 Aims and Objectives	4
2 Literature Review	5
3 Methodology	10
3.1 Workplan	10
3.2 Dataset Discription	10
3.3 Dataset Preprocessing	11
3.4 Convolutional Neural Network Models	15
4 Experiment and Results	18
4.1 Measurements	18
4.2 Experiment with Models	20
5 Conclusion	47
Bibliography	50

List of Figures

3.1	workflow	10
3.2	STFT Architecture[2]	13
3.3	MFCC Architecture[14]	15
3.4	The-architecture of the deep residual network	15
3.5	The-architecture of the EfficientNetB0	16
3.6	InceptionV3	16
3.7	Densenet121 Architecture[23]	17
4.1	confusion matrix example	20
4.2	code of EfficientNetB0	21
4.3	Training and validation accuracy curves of ResNet50 from MFCC feature	22
4.4	Training and validation loss curves of ResNet50 from MFCC feature	22
4.5	Confusion matrix of ResNet50 from MFCC feature	23
4.6	Training and validation accuracy curves of EfficientNetB0 from MFCC feature	24
4.7	Training and validation loss curves of EfficientNetB0 from MFCC feature	24
4.8	Confusion matrix of EfficientNetB0 from MFCC feature	25
4.9	Training and validation accuracy curves of InceptionV3 with MFCC feature	26
4.10	Training and validation loss curves of InceptionV3 with MFCC feature	26
4.11	Confusion matrix of InceptionV3 with MFCC feature	27
4.12	Training and validation accuracy curves of DenseNet-121 with MFCC feature	28
4.13	Training and validation loss curves of DenseNet-121 with MFCC feature	28
4.14	Confusion matrix of DenseNet-121 with MFCC feature	29
4.15	Training and validation accuracy curves of ResNet50 from STFT feature	30
4.16	Training and validation loss curves of ResNet50 from STFT feature	30
4.17	Confusion matrix of ResNet50 from STFT feature	31
4.18	Training and validation accuracy curves of EfficientNetB0 from STFT feature	32
4.19	Training and validation loss curves of EfficientNetB0 from STFT feature	32
4.20	Confusion matrix of EfficientNetB0 from STFT feature	33
4.21	Training and validation accuracy curves of InceptionV3 with STFT Feature	34
4.22	Training and validation loss curves of InceptionV3 with STFT Feature	34
4.23	Confusion matrix of InceptionV3 with STFT Feature	35

4.24	Training and validation accuracy curves of DenseNet-121 with STFT	36
4.25	Training and validation loss curves of DenseNet-121 with STFT . . .	36
4.26	Confusion matrix of DenseNet-121 with STFT	37
4.27	Training and validation accuracy curves of ResNet50 from STFT & MFCC combination Feature	38
4.28	Training and validation loss curves of ResNet50 from STFT & MFCC combination Feature	38
4.29	Confusion matrix of ResNet50 from STFT & MFCC combination . .	39
4.30	Training and validation accuracy curves of EfficientNetB0 from STFT & MFCC combination Feature	40
4.31	Training and validation loss curves of EfficientNetB0 from STFT & MFCC combination Feature	40
4.32	Confusion matrix of EfficientNetB0 from STFT & MFCC combination	41
4.33	Training and validation accuracy curves of InceptionV3 with STFT & MFCC combination Feature	42
4.34	Training and validation loss curves of InceptionV3 with STFT & MFCC combination Feature	42
4.35	Confusion matrix of InceptionV3 with STFT & MFCC combination .	43
4.36	Training and validation accuracy curves of DenseNet-121 with STFT & MFCC combination Feature	44
4.37	Training and validation loss curves of DenseNet-121 with STFT & MFCC combination Feature	44
4.38	Confusion matrix of DenseNet-121 with STFT & MFCC combination	45

List of Tables

4.1	Classification report of ResNet50 from MFCC feature	23
4.2	Classification report of EfficientNetB0 from MFCC feature	25
4.3	Classification report of InceptionV3 with MFCC feature	27
4.4	Classification report of DenseNet-121 with MFCC feature	29
4.5	Classification report of ResNet50 from STFT feature	31
4.6	Classification report of EfficientNetB0 from STFT feature	33
4.7	Classification report of InceptionV3 with STFT Feature	35
4.8	Classification report of DenseNet-121 with STFT	37
4.9	Classification report of ResNet50 from STFT & MFCC combination .	39
4.10	Classification report of EfficientNetB0 from STFT & MFCC combination	41
4.11	Classification report of InceptionV3 with STFT & MFCC combination	43
4.12	Classification report of DenseNet-121 with STFT & MFCC combination	45
4.13	Comparison among all the models	46

Nomenclature:

- CNN Convolutional Neural Network
- STFT Short-Time Fourier Transform
- MFCC Mel-frequency cepstral coefficient
- RF Random Forest
- ASR Automatic speech Recognition
- TTS Text-to-Speech
- STT Speech-to-text
- SVM Support vector Machine
- HMM Hidden Markov Models
- LPC Linear Predictive Coding
- GMM Gaussian Mixture Model
- KNN K-Nearest Neighbors Algorithm
- MLP Multilayer Perception
- LDA Linear Discriminant Analysis
- ANN Artificial Neural Network
- CSV Comma Separated Values
- TSV Tab-Separated Values
- FFT Fast Fourier Transform
- DCT Discrete Cosine Transform
- VGG Visual Geometry Group
- ReLU Rectified Linear Unit
- TP true Positive
- TN True Negative
- FP False Positive
- FN False Negative
- CART Classification And Regression Tree
- ML Machine Learning
- RNN Recurrent Neural Network

Chapter 1

Introduction

1.1 Thoughts behind Gender Identification

Voice is the primary mode of human communication. Humans use different types of voices or tones according to situations, places, emotions, etc. Sometimes we use loud tones, whereas other times we use low-frequency tones. Also, some of the male voices are kind of similar to the female voices, which have a higher pitch than usual male voices, and some female voices have lower pitches than typical female voices. As a result, it becomes difficult to discern gender from the wide range of tones, accents, frequencies, and pitches.

The increasing popularity of virtual platforms for education and communication has led to significant advancements in the field of speech analysis. Applications like Text-to-Speech (TTS) and Automatic Speech Recognition (ASR), commonly referred to as Speech-to-Text (STT), have emerged as a result. These technologies are widely applied in fields including computerized language assessment for educational reasons, the diagnosis and treatment of linguistic disorders, better agricultural practices, and voice-activated assistants that make technology more accessible.

Unfortunately, the research and development of Bengali language technologies similar to those mentioned above have been hampered by the absence of sufficient datasets, despite significant theoretical and computational efforts focused on modeling Bengali phonology and the creation of potent deep learning networks. Bangla is one of the most widely spoken languages, with 234 million native speakers and 39 million speakers who use it as a second language. It is the sixth most spoken language in the world. A huge percentage of this group does not know any other language for communication. Therefore, we are trying to distinguish gender through speech in the Bangla Language.

Our dataset collection approach was split into two independent methodologies going forward. Firstly, we obtained a dataset from the Mozilla Common Voice platform. This platform contains a sizable collection of transcribed audio recordings that span more than 400 hours and contain spoken Bengali words. Through teamwork and community outreach initiatives, this dataset was compiled with contributions from many parts of Bangladesh and India.

Secondly, we crowdsourced an extensive dataset that span nearly 7 hours containing spoken Bangla words that went through pre-processing steps similar to training data. This independently created dataset was primarily intended for testing and assessing the effectiveness of the CNN models under consideration.

We used convolutional neural networks (CNN), a deep learning model, for gender classification in Bangla language. CNNs comprise a particular kind of neural network known for their effectiveness in voice and picture recognition applications. The dataset consisting of Bengali speech recordings serves as the training data for the CNN model, with each recording carrying information about the speaker's gender. The algorithm has been trained to recognize the distinctive speech characteristics of each gender. In our work, we used four different models to train our classification algorithm: ResNet50, EfficientNetB0, InceptionV3, and DenseNet-121.

The input data for these models was taken from the audio recordings and was converted into a 2D array using the Short-Time Fourier Transform (STFT) and Mel-frequency cepstral coefficients (MFCC). After that, we used this array as input to train each model. To study variances in model performance and identify the ideal configuration for our research, we also carried out independent experiments employing STFT and MFCC alone as well as in combination. This thorough examination made it easier to choose the best model for our gender classification task.

1.2 Problem Statement

The purpose of our research paper is to create a robust and highly accurate gender differentiation algorithm for the Bangla Language using the Deep Learning process. We used a large dataset of Bangla speech data which helped us to improve our algorithm performance of the system.

In the context of this research, the challenges encountered are:

1. It is hard to distinguish between male and female recognition systems in Bangla Language due to a lack of online data and practical approaches. The methods based on CNN which are available on the internet are insufficient for understanding the patterns and the voice nodes.
2. In the data set, the speakers speak in different accents and tones, making it more complex for the algorithm to understand the pattern and the files.
3. Additionally, the files of the audio clips are not the same length and size. Also, the speakers of the audio files did not follow any rules to speak. They spoke in different emotions consisting of different frequencies. So, the accuracy level became lower than expected at first.

In conclusion, it is mandatory to deal with these challenges and develop a Deep Learning-Based approach for gender recognition in Bangla Language. We aim to improve the accuracy and robustness of our research by collecting data and training our algorithm with the help of these.

1.3 Aims and Objectives

The objective of our research is to create a deep learning-driven gender classification system tailored for Bangla speakers, aiming to achieve the highest possible accuracy and resilience. Our goal is to analyze the Bangali Speaker to differentiate the gender using voice analysis techniques.

Our primary goals in this study are:

- Create a deep learning model to distinguish between speakers of Bangla who are male and female. The model will be trained to recognize the pattern and the key distinctions between the two genders with ease.
- Find a solution for different homophones, tones, and accents in Bangla language.
- Uncovering Gender Bias in Deep Learning Models for Bangla Language
- Deep Learning-Based Gender Classification in Bangla for different age groups.

Chapter 2

Literature Review

Speaker recognition, speech synthesis, and emotion recognition are just a few of the fields that have shown significant interest in gender categorization, the process of determining a speaker's gender based on their voice. The widespread use of digital media as a medium for instruction and communication has accelerated the development of voice analysis. Numerous studies based on speech classification and gender-inferred identification have been conducted for quite some time. Therefore, studies focusing on the detection of gender are not new. There is preliminary evidence that deep learning models and data mining can accurately identify a speaker's gender based on their voice. Here we provide a few practical results from these studies.

Parris and Carrey[19] specifically developed two techniques for gender recognition, combining acoustic analysis and pitch. A linear classifier is used to integrate the data from the acoustic analysis and pitch estimate to determine the gender of the voice. With two seconds of speech, the system was evaluated on three British English databases, which had an identification error rate of less than 1.0%. Additional testing using the OGI database's eleven languages without optimization produced error rates of less than 5.2% and an average of 2.0%.

In another study, gender recognition was done by comparing different classification algorithms[16]. At first, predictions were primarily made based on the dataset and algorithms, which were collected from 3000 male and female audio files. In order to determine which method produces superior results in detecting gender-derived particular parameters, results are compared with prior prediction results. Then the classification algorithms are used for estimation. From the previously discussed algorithms, a precise prediction of how the comparative algorithmic technique detects gender is obtained. The algorithms that were used here are the Gradient Boosting Algorithm, Decision Tree, Random Forest, Support Vector Machine (SVM), and Neural Network. It was derived from the comparison that Gradient Boosting Algorithm gave the best result with 90% accuracy, whereas Neural Network and Random Forest showed 89% accuracy.

Livieris, Pintelas, and Pintelas[15] opted for a semi-supervised algorithm called iCST-Voting, which includes self-labeled algorithms like Tri-training, Self-training, Co-training, Co-bagging, and Democratic-Co-Forest. The iCST-Voting model achieved

a remarkable 98.42% accuracy, with the downside of an increase in training time.

A study by Kotwal, Hassan, and their group in 2011 used Hidden Markov Models (HMM) to address gender effects in Bengali automatic speech recognition. Their goal was to use different HMMs for males and females to establish a method for gender impact suppression. They created a medium-sized Bengali speech corpus and split it into training sets for men and women. These sets had 3,000 phrases delivered by 30 speakers from different geographical locations, one of each gender. These training sets were used by the researchers to create Bengali triphone HMMs. Mel-frequency cepstral coefficients (MFCCs) were used in the experimental setup to extract features. In order to evaluate their methodology, the study used four experimental designs, which produced remarkable word accuracy scores of 90.78% and sentence correctness scores of

A study on gender recognition utilizing real-time audio processing techniques in LabVIEW was carried out in 2011 by Rakesh, Dutta, and Shama. By examining acoustic metrics, particularly F0 and F1, they sought to determine a speaker's gender. Framing, windowing, pre-emphasis, and Linear Predictive Coding (LPC) were some of the speech processing techniques used in the study. They also looked into using pitch detection to figure out the speaker's age and gender. The study illustrated the efficiency of LabVIEW in speech detection by examining F0 and F1 values, which are typically approximately 120 Hz for males and 210 Hz for females for F0 and 387 Hz for males and 432 Hz for females for F1. It did, however, point out several inaccuracies when vowels were absent.[22]

Yucesoy and Nabiyev[29] created a method that was centered on the classification of MFCC coefficients that were derived from speech signals by utilizing GMM. The process consists of two distinct stages. The initial step in the process is the training of the system with quotations from prominent speakers of both genders. The system is then put through its paces, utilizing comments made by speakers whose gender is hidden for the subsequent round of testing. For the purpose of this inquiry, the researchers accessed the TIMIT database to compile the speech data they needed. The TIMIT database is comprised of ten lines, only two of which are identical to one another while the remaining lines are all original. These statements were said by a total of 630 people, including 438 males and 192 women, all of whom are native speakers of one of the eight basic varieties of American English. In two separate accuracy tests, the system scored an impressive 100 and 97.76% respectively.

The model presented by Jalil, Stephan, and Najji[11] determines the gender of speech samples using voice recognition. The method employs the 12 most important characteristics extracted from each voice sample, as well as a number of voice parameters, such as Mean, Zero-Crossing, Standard Deviation, and Amplitude. Feature vectors for the voice are generated by combining these characteristics. Several machine learning and deep learning algorithms, including Random Forest, KNN, Logical Re-

gression, Decision Tree, and CNNs, are employed by the proposed method to classify speech vectors as Male or Female. By comparing the evaluation metrics of each classifier, the suggested method concludes that the CNN model is the most successful classifier. It attains a precision value of 1 for accurately identifying voice vectors.

To integrate gender-based user categorization, Jasuja, Rasul, and Hajela[12] created a deep learning model utilizing Multilayer Perception (MLP). As input, the model receives a set of acoustic characteristics collected from various products. The researchers compiled a dataset containing 3,168 data points from male and female voice samples. These vocal samples were created using methods of acoustic analysis. By training the network with a variety of parameter configurations, they ultimately developed an MLP model with an accuracy of 96 percent on the test dataset. In addition, the researchers highlighted the effectiveness of batch normalization and dropout in preventing the model from becoming overly specialized to the training data. Similar research was conducted by Buyukyilmaz and Cibikdiken[8] in which a Multilayer Perceptron (MLP) deep learning model achieved 96.74% accuracy on 3,168 male and female voice recordings.

Another research[20] used audio data to identify gender using five machine learning algorithms: Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Random Forest (RF), K-Nearest Neighbour (KNN), and Support Vector Machine (SVM). These methodologies were evaluated using eight distinct metrics, including Box, Density, Parallel, Dot, and Pair plots and statistical significance tests. From the voice dataset, vocal strength, pitch, frequency, q21, q25, and other metrics were extracted. Using these parameters, the numerous algorithms were then trained and validated. Based on several prediction algorithms, the study suggests a comparison model algorithm that divides individuals into gender categories. On a dataset constructed from voice data, predictions are built. The collected data is compared to other algorithms to determine which algorithm performs best in terms of gender categorization based on specific factors. The outcomes are contrasted to those of previous research. Also evaluated is the efficacy of the comparison model algorithm's gender detection. The results of this study indicate that the SVM algorithm performs better than other algorithms at classifying gender in the presence of pitch and frequency changes. SVM has superior gender classification accuracy and reduced error rates than other algorithms.

In their 2016 study, Pahwa and Aggarwal used speech feature extraction to create a gender recognition model for Hindi speech. They investigated a number of algorithms, including SVM and ANN, to improve gender recognition in phone applications and speech data from many languages. They developed a system that can identify gender from WAV speech files using a speech library of 50 speakers, Mel-frequency cepstral coefficients (MFCC), and other speech parameters. The model's accuracy increased to 93.48% when the initial MFCC coefficient was taken into account, while it remained highly accurate (91.3%) even when it wasn't. The study also emphasized the value of the MFCC 1 algorithm, which greatly increased overall

accuracy by 2.18% and led to recognition rates of 92.33% for females and 95.22% for males.[18]

Wang et al.[28] use deep neural networks (DNNs) to combine the activations from the final hidden layer over time in order to transform each utterance into a fixed-length vector. This feature encoding procedure is trained alongside the utterance-level classifier to enhance classification accuracy. To further enhance utterance-level categorization, they used encoded vectors to train a kernel extreme learning machine (ELM). The results of the research indicate significant gains in accuracy. In the task of emotion recognition, it improves weighted and unweighted accuracy by 3.8% and 2.94%, respectively, compared to a robust DNN-ELM strategy. In the age/gender recognition challenge, this method performs as well as or better than human evaluators. However, this study is more concerned with emotion detection than with gender classification. This research, however, utilized a Mandarin dataset with varying utterance levels.

Gupta, Goel, and Purwar (2018) proposed a novel method for gender identification through speech analysis, utilizing 20 different acoustic variables and stacked machine-learning approaches. Their study used a dataset split into three sections for training and testing with the goal of improving gender detection accuracy. Unexpectedly, despite identical inputs, different outcomes were produced by different models, including SVM, CART, and a neural network. This led the investigators to integrate these models for better performance. The study increased accuracy by 2% and noticed significant gains in precision, specificity, and F-score by gathering audio from 3160 people. The accuracy was increased by the stacked model to 96.74%, although acknowledging possible difficulties with larger and more varied datasets.[9]

Artificial neural networks were used in another research project[27] as a standard method for classifying speech data. 3168 voice samples from men and females made up the dataset. Voice samples were analyzed acoustically using the R programming language's seewave and tuneR utilities. In order to increase classification accuracy, the dataset was divided into ten pieces, with each portion used for testing and retesting. The average classification success was calculated using the arithmetic mean of the findings. Using artificial neural networks, a remarkable success rate of 97.9% was achieved in precisely distinguishing between male and female audio.

Bangla, often known as Bengali, is one of the most widely spoken languages on the Indian subcontinent. However, research on gender classification in the Bangla language using speech analysis is not as extensive as it is for other languages. To meet the needs of the Bangla-speaking community, there is a demand for specialized research and development in this area.

A method for identifying gender from Bengali voice data was published in a study[21],

which involved establishing a corpus, generating input feature vectors, and employing a classifier. What makes this research for Bengali unique is the application of machine learning algorithms for gender detection and the focus on the extraction of certain features from the input speech. The training and testing corpus comprised 8,000 words from 80 male and 80 female speakers, selected using 10-fold cross-validation. The results of the experiments showed that the vast majority of the classifiers achieved perfect accuracy.

In a separate investigation[6], the Mel-frequency cepstral coefficient (MFCC) was utilized to categorize Bangla voice samples by gender. Gradient boosting, random forest, and logistic regression were utilized for the mapping and selection processes. The researchers Badhon et al. [12] used a straightforward method to get their findings. The approach involved reading the audio file, performing some pre-processing steps, extracting features (including MFCC), writing out a CSV file containing the feature extractions, training the model, and putting it through its paces on test data. The model achieved 99.13% accuracy when tested on a dataset of 1652 instances from over 250 speakers. The test data included the voices of 400 men and 400 women.

When conducting research, having access to a large and diversified dataset is crucial. Data from the Bengali Common Voice Speech Dataset[4], collected via crowdsourcing and retrievable using sentence-level automatic speech recognition, was utilized in this regard. Using the Mozilla Common Voice platform, the dataset was collected as part of an ongoing study. In just two months, more than 400 hours of data were collected, and it is continuing to expand rapidly. OpenSLR Bengali ASR is the largest publicly accessible dataset for automatic speech recognition. However, this dataset has been found to have more variety than others in terms of speakers, phonemes, and contextual factors. Insights from the dataset are shared, and major linguistic concerns that will need fixing in future iterations are highlighted.

There have been improvements in gender classification with the use of deep learning-based voice analysis, but many issues still need to be resolved. The lack of a large-scale dataset designed specifically for gender categorization in the Bangla language is a major problem. There is also a need for greater research into how variations in Bangla speakers' accents, dialects, and speech patterns influence their gender assignment.

For our research, we use four different models: ResNet50, EfficientNetB0, InceptionV3 and DenseNet-121. The InceptionV3 model outperforms the others in terms of gender recognition. Additionally, we can quickly identify features and patterns in our audio files that have been converted to .tsv files with the use of these models.

Chapter 3

Methodology

3.1 Workplan

The model training was conducted using the audio files extracted from the validation.tsv and train.tsv files sourced from the Mozilla Common Voice Dataset[17]. Our dataset from Mozilla Common Voice is divided into an 80% training set and a 20% validation set. The audio from these files is used in the training process.

A separate dataset was used that we have curated for testing and evaluation. We utilized the Short-Time Fourier Transform (STFT) and Mel-frequency cepstral coefficients (MFCC) to transform the audio recordings into 2D arrays. Our convolutional neural network (CNN) models are then evaluated using these altered data representations.

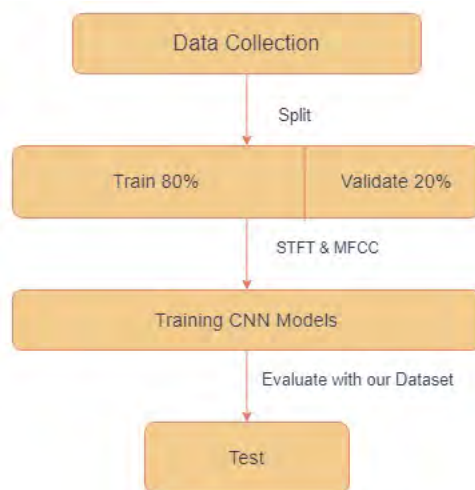


Figure 3.1: workflow

3.2 Dataset Description

For our training data, we are utilizing a dataset from Mozilla Common Voice version 13.0 [5] The over 400 hours of Bangla speech recordings that make up this dataset are the result of continuing work. By reading specified prompts, contributors to the Mozilla Common Voice platform recorded single-channel audio data at a rate of

48kHz. With an average length of 7.12 words, the text database contains 135,625 unique sentences. Each word contains 3.24 graphemes and 4.95 Unicode characters on average. 399 hours of speech recordings totalling 231,120 samples from 19,817 contributors make up the collection. A "sentence" annotation and additional meta-data, such as "up votes," "down votes," "age," "gender," and "accents," are linked to each audio clip.

With a 4.8% margin of error and 99% confidence, A survey was conducted on 720 recordings randomly chosen from the full dataset (containing samples from both validated and invalidated portions) for qualitative analysis, with a 4.8% margin of error and 99% confidence. These recordings exhibited a 0.417 % incomplete rate, a 0.833% too fast to understand rate, a 0.417% considerable background noise rate, a 4.444% muffled rate, a 0.417% incomprehensible rate, a 1.25% stuttering rate, and a 0.694% no speech data rate.

About 80% of the training data and 20% of the validation data were taken from the 56 hours of speech that made up the validated portion of this dataset. The dataset primarily had an unequal distribution of data of male and female speakers. We balanced the dataset by including 2,846 voice samples from both male and female speakers in order to address this. Each voice sample was divided into segments of two seconds. As a result, we were able to collect 9,506 male and 9,249 female voice clips.

For our testing data, we have crowdsourced a dataset of our own with a total of 70 voice data consisting of 35 male audio files and 35 female audio files which adds up to 7 hours. We asked data contributors to record single-channel audio data by reading prompts that we wrote ourselves. The text corpus contains 147 unique sentences. Each sentence has 8.39 words on average and each word contains 3.25 Unicode characters on average. We also segmented our own data by a range of 2 seconds. By this, we have 4518 male audio clips and 5407 female audio clips.

3.3 Dataset Preprocessing

The crucial component of the machine learning workflow that significantly influences model performance is data preparation. This crucial phase helps to improve the model's accuracy, decrease the amount of time and computational resources needed for training, reduce overfitting problems, and make the model easier to understand. We thoroughly analyzed and conducted preprocessing on the dataset which follows as:

1. Segmentation: Due to the varying lengths of the audio recordings in our dataset, we standardized them by dividing each audio into 2-second segments and classifying them according to gender in distinct folders.
2. Labeling: We gave labels to the segmented audio files, designating males as "1" and females as "0," to help discriminate between different data points.
3. Data Splitting: We split our dataset into an 80:20 ratio, allocating 80% to training and 20% to validation.

4. Conversion: All audio files' sampling rates were consistently changed to 16,000 Hz. The audio files were then converted into 1D arrays using the 'librosa.load()' method, which were then stored as numpy arrays.
5. Padding: We added padding to the 1D arrays to account for the different audio file sizes. As a guide, we established a maximum length of 40,000 (the average length of sample audio files). Any audio that was longer than this was trimmed, while shorter audio files were padded with zeros to keep their length constant. This made sure that consistent 2D matrices were produced that could be used with CNN models.
6. STFT and MFCC: At first, we converted the 1D arrays to create 2D arrays by applying the Short-Time Fourier Transformation (STFT). When using CNN models, this method produced results that were pretty accurate, although we ran into overfitting problems. These problems were related to noise in the audio data, which decreased the STFT process's dependability. As a result, we resorted to using the MFCC method. We extract critical audio features using MFCC, then use STFT to further enhance CNN model training.

STFT

STFT (Short-Time Fourier Transform) is utilized in signal processing, notably in the analysis and representation of signals in the time-frequency domain. It is a useful tool for figuring out how a signal's frequency content changes over time.

Features of STFT:

- Time-Frequency Representation: A signal is concurrently represented in the time and frequency domains by STFT. It demonstrates how a signal's frequency components alter as a function of time.
- Localised Analysis: A signal's individual time windows can be examined using STFT. It divides the signal into brief, overlapping pieces rather than examining it as a whole, giving insights into localized fluctuations.
- Conversion from Complex to Real: STFT transforms a complex signal into a real-valued representation, making further analysis easier.

When computing its Fourier transform, the STFT only takes into account a short-duration slice of a longer signal. Typically, this is done by multiplying a window function with a short duration, by a longer time function. The rectangular window, which effectively extracts only the necessary brief sequence without additional modification, and the Hamming window, which applies a taper to the ends to better the representation in the frequency domain, are two often used finite-duration windows.

The formula of STFT:

$$X[t, \omega] = \sum_{-\alpha}^{+\alpha} x[m] \omega[t - m] e^{-j\omega m}$$

Here,

1. STFT(t, ω) represents the STFT of the signal at time t and frequency ω
2. $x(m)$ is the input signal.
3. $\omega(t-m)$ is a window function that determines the time window over which the Fourier.
4. $e^{-j\omega m}$ is the complex exponential representing the frequency domain analysis.

Benefits of STFT:

Time-Frequency Analysis: STFT is suited for analyzing non-stationary signals because it gives a clear knowledge of how a signal's frequency content changes over time.

Localized Information: It is useful for applications like voice analysis, audio processing, and music analysis because it enables researchers to concentrate on particular time intervals of interest within a signal.

Visualization: STFT representations are often shown as spectrograms, which are frequently utilized in applications like voice recognition and audio analysis, they are frequently visually intuitive.

Efficient Computation: Fast Fourier Transform (FFT)-based STFT computations are more efficient than those of the continuous Fourier transform, making them suitable for real-time and massively parallel signal processing tasks.

In conclusion, STFT is a useful tool for studying signals in the time-frequency domain because it sheds light on how a signal's frequency components evolve over time. It is a vital technique in numerous signal-processing applications due to its capacity to provide localized information and efficient computation.[1]

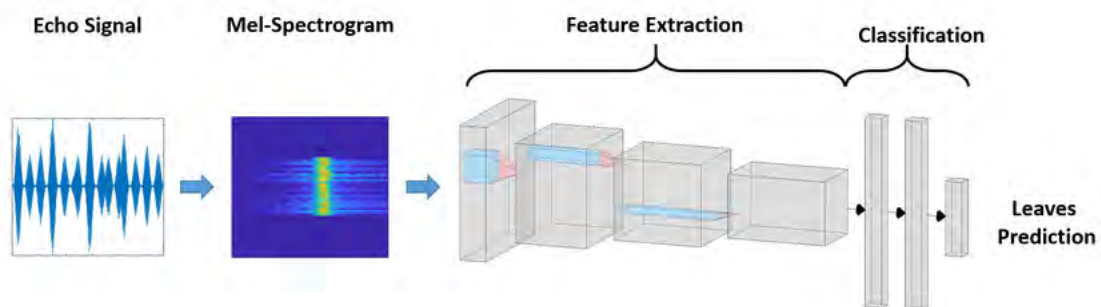


Figure 3.2: STFT Architecture[2]

MFCC

The Fast Fourier Transform (FFT) is frequently used to derive the Fourier spectrum from an audio source in the field of audio signal processing. Two steps make up this

procedure. First, the logarithm of the Fourier spectrum's magnitude is calculated. The cepstrum within the signal then emerges when this logarithmic result is transformed using a cosine function.

The mathematical formula for the link between the frequency (denoted as 'f') and the Mel-frequency (denoted as 'm') is:

$$\text{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Understanding the perceptual features of audio transmissions depends on this relationship. Particularly, it has been found that the usage of Mel-frequency-spaced spectral windows exhibits similar sensitivities to human aural perception. This encourages the use of the Mel-frequency cepstral coefficients (MFCCs) for the analysis of audio signals.

There are several phases involved in the calculation of MFCCs:

- Making use of the Fourier transform to determine the power spectrum, or $|X(f)|^2$, of the sound signal, $x(t)$.
- Projecting a collection of Mel-frequencies with equal spacing, m_k , $k = 1, 2, 3, \dots$, into the frequency domain to produce f_k , $k = 1, 2, 3, \dots$.
- Obtaining the weighted sum of the power spectrum using triangular windows centered at f_k , then calculating the logarithm of the power integral for each Mel-frequency.
- Using the discrete cosine transform to change the values of the logarithmic power into MFCCs.

Due to its nonlinear structure, the term "cepstrum" is famous for its odd language construction. It is frequently referred to as a "reverse" or "spectrum of a spectrum." The resulting cepstrum coefficients play a crucial part in characterizing the changes in the audio stream across distinct spectrum bands.

It is useful to represent these coefficients in terms of the mel scale rather of a linear scale for specific applications, especially in speech and audio analysis. The cepstral coefficients become mel-frequency cepstral coefficients (MFCCs) as a result of this modification, which creates the mel-frequency cepstrum. By using the Mel scale in this context, these coefficients' discrimination skills in audio analysis tasks are improved, bringing them closer to the characteristics of human auditory perception.[3][25]

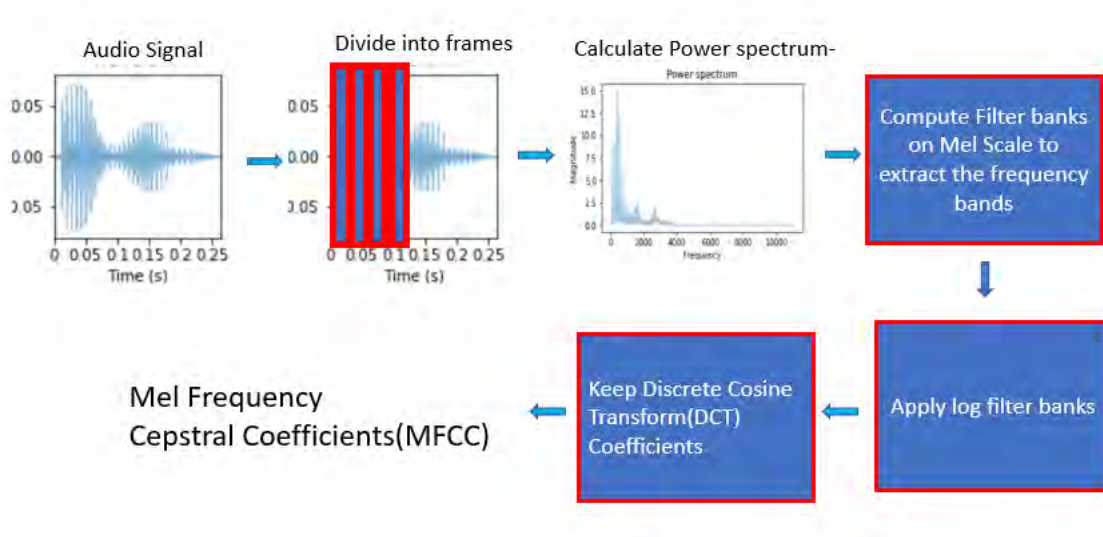


Figure 3.3: MFCC Architecture[14]

3.4 Convolutional Neural Network Models

- **Resnet50**

The term Resnet improves Residual network. Resnet50 is the modern architecture for Resnet, which has 48 convolutional layers. The basic and very first Resnet Architecture is Resnet34 which consists of 34 convolutional networks. It was an efficient architecture to provide layers to a CNN without having the vanishing gradient issue. The modern network was established on VGG which had a frame of 3x3. But still, Resnet is faster as it has fewer filters and is much easier than VGG. Figure 1 describes a block diagram for Resnet50[13]. In the ImageNet challenge, it beats the other data models very efficiently. That is why Resnet50 is better than other models[10]. Figure 1 describes a block diagram for Resnet50[13].

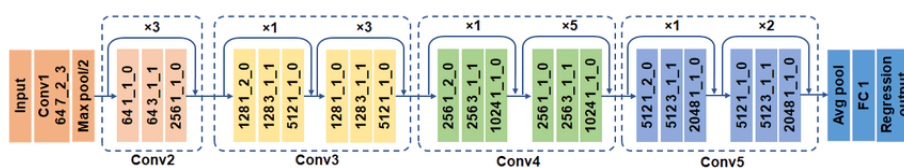


Figure 3.4: The-architecture of the deep residual network

- **EfficientnetB0**

Efficientnet adds new degrees to scale up convolutional networks. It scales up convolutional networks by their depth or width. In the depths, the network catches more complex features but it is very hard to train. Whereas, in-width network catches more fine-grained features and is easy to train. Moreover, in high-resolution pictures, the network also captures fine-grained pictures but the accuracy gain diminishes[26].

Width Scaling Increases the number of channels in each layer, effectively increasing the model's capacity to learn more complex features. Depth Scaling Increases the number of layers in the network, allowing for more fine-grained

features to be captured. Resolution Scaling Increases the input image resolution, enabling the model to process higher quality and more detailed images. The goal is to find a balance between width, depth, and resolution scaling factors to maximize the model performance.

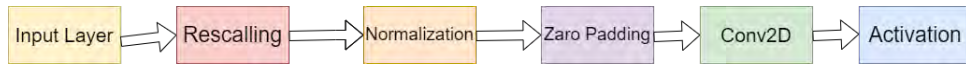


Figure 3.5: The-architecture of the EfficientNetB0

- **InceptionV3**

The fundamental building block in InceptionV3 is the inception module, which is designed to process input data using multiple convolutional operations of different sizes (1x1, 3x3, 5x5) and max pooling simultaneously. This allows the model to capture features at various spatial scales within the same module. InceptionV3 is the upgraded model of Inception V1. It has a higher frequency than its predecessors. It is built up of 42 layers which is higher than Inception V1 and V2. Furthermore, this model is efficient in grid size reduction. For example, if we have a $d \times d$ grid with k filters after reduction it results in a $d/2 \times d/2$ grid with $2k$ filters. The activation dimension of the network filters is expanded so that the grid size gets reduced efficiently. Moreover, to work with a 5x5 convolutional matrix is pretty expensive and to overcome this problem the inceptionV3 model modifies the 5x5 layers into 3x3 which both cuts down the costs and computational error[24].

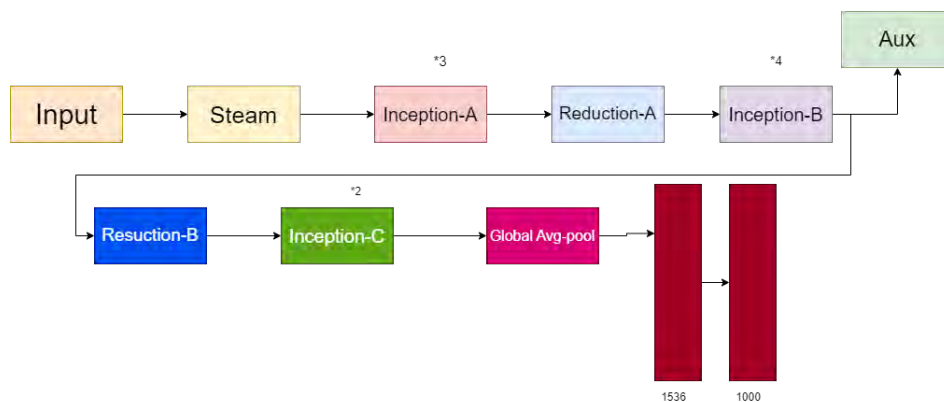


Figure 3.6: InceptionV3

- **DenseNet-121**

Densely Connected Convolutional Network is a deep neural network architecture known for its dense connectivity pattern between layers where each layer is directly connected to every other layer in a feed-forward fashion. This model promotes feature reuse, reduces the number of parameters, and alleviates vanishing gradient issues. They employ growth rates to control information addition, direct layer connections, feature map concatenation, and layer concatenation. DenseNet-121 is composed of three primary types of layers: Convolutional Layer, Batch Normalization layers, and Rectified Linear Unit (ReLU) Activation. An input is sent through each layer of a model being trained with DenseNet-121. Layers are tightly connected, ensuring that

feature maps move fluidly from one to the next. This makes it easier to learn complex feature representations and enhances gradient flow throughout the training process, leading to effective deep learning. [23]

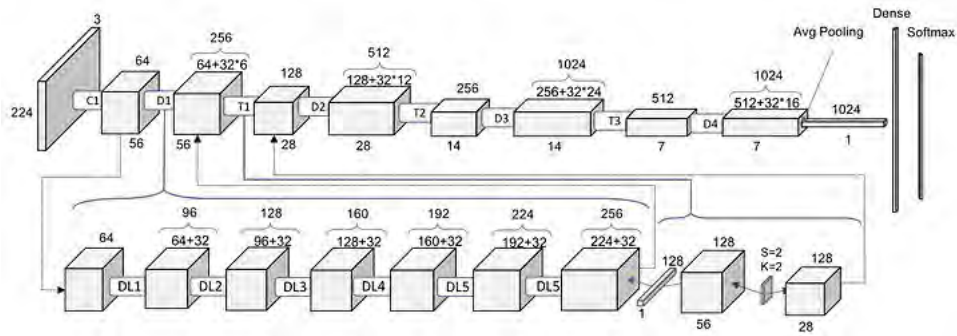


Figure 3.7: Densenet121 Architecture[23]

Chapter 4

Experiment and Results

4.1 Measurements

Essential performance indicators for assessing the effectiveness of classification models include precision, recall, F1 score, and confusion matrices. These metrics are essential for evaluating a model's performance in particular circumstances since they provide insightful information about many different aspects of a model's success in classifying data.[7].

- True Positives (TP): These are situations in which the model successfully detected positive cases. To put it another way, the model correctly identified good outcomes as such.
- True Negatives (TN): Positive cases that the model properly recognized as negatives are known as True Negatives (TN). It indicates that the model correctly identified negative occurrences as such.
- False Positives (FP): These are instances where the model projected mistakenly that positive outcomes would occur when they actually occurred negatively. The model occasionally misclassifies negative events as positive ones.
- False Negatives (FN): These are situations where the model incorrectly anticipated negative results when they were really positive. The model sometimes misclassifies positive cases as negative ones.

Precision: It indicates how well the model can identify the positive events among those that it anticipated to be positive. Reducing false positives is its goal. High precision means that it's unlikely that the model will classify negative events as positive. For instance, in a scenario involving a medical diagnosis, precision is the proportion of correctly identified positive instances among all cases that were expected to be positive. In essence, accuracy evaluates the model's dependability in positively labeling occurrences. It is a crucial statistic, particularly when false positives can have serious repercussions or when we wish to reduce the possibility of

making falsely positive forecasts.

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

Recall: In academic contexts, "Recall," also known as "Sensitivity" or "True Positive Rate," measures how well a model can detect all occurrences of positivity. Its main goal is to reduce false negatives. The model shows a decreased chance of misclassifying positive examples as negative when memory is high. This indicator, also known as sensitivity or the true positive rate, assesses how well the model performs in correctly identifying positive events. It mathematically denotes the percentage of positively predicted occurrences that really occurred out of all positively occurring outcomes (TP + FN). The model's memory essentially offers information on how well it recognizes actual positive experiences.

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

F1 Score: It is created by taking the harmonic mean of recall and precision, combining these two essential components into a single measurement. The harmonic mean gives more weight to the lower value when precision or recall considerably lags behind the other, which lowers the F1 score overall. The F1 Score recognizes and rewards models that demonstrate both high precision and high recall concurrently, as opposed to favoring one statistic over the other. A balanced evaluation tool, the F1 Score, is provided by this combined metric, particularly when there is a trade-off between precision and recall. It has a scale from 0 to 1, with 1 being the highest possible performance. The F1 Score is a crucial metric in model evaluation and comparison since it essentially captures the thorough examination of classification models, taking into account their capacity to reduce both false positives and false negatives.

- F1 Score = $\frac{2TP}{TP+FP+TN+FN}$

Accuracy: A key performance statistic for model training is accuracy, which quantifies the percentage of instances from a dataset that were correctly predicted. It is a crucial indicator for assessing a machine learning model's efficacy, particularly in classification tasks. It provides information on the model's accuracy in classifying situations. The proportion of true positives and true negatives among all cases is measured in binary classification. It is frequently combined with other metrics to give a more thorough picture of a model's performance.

- Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$

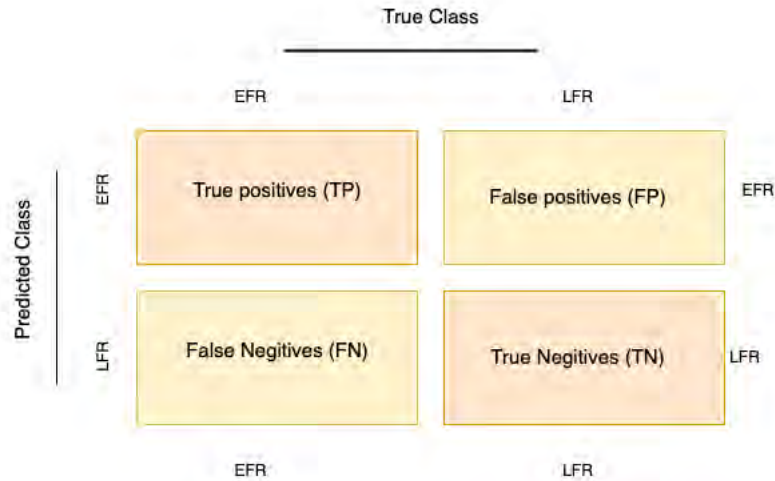


Figure 4.1: confusion matrix example

4.2 Experiment with Models

We can see the EfficientNetB0 model’s training procedure and performance metrics in the provided snippet of code. The important components are highlighted in this description:

- Epochs: The model goes through 50 epochs, each of which involves a full iteration of the training dataset.
- Batch Size: The dataset is divided into batches for training, with each batch containing 32 samples. After processing each batch, the parameters of the model are updated.
- Learning Rate: The learning rate is initially set to 0.001 (1e-3) per second. The size of the step at which the model’s parameters are changed during training is determined by this learning rate.
- The Adam optimizer: It is renowned for its adaptable learning rates and momentum and contributes to effective model refinement, is used to optimize the model.
- Model Compilation: During compilation, the Adam optimizer, a binary cross-entropy loss function, and accuracy as the evaluation measure are all set up for the model.
- Training: Training data (labeled "X train" and "Y train") and validation data (labeled "X val" and "Y val") are provided. Each epoch of the training process is tracked, and loss and accuracy scores for the training and validation

datasets are displayed.

- **Loss and Accuracy Plotting:** Following training, graphs displaying the loss and accuracy curves for both the training and validation datasets are produced using Matplotlib.

```
[9]: model_backbone = tf.keras.applications.EfficientNetB0(
      include_top=False,
      weights=None,
      input_shape=(128, 128, 1),
      pooling=None,
    )
    model_backbone.summary()

    model = Sequential()
    model.add(model_backbone)
    model.add(GlobalAveragePooling2D())
    model.add(Dense(units=2, activation='softmax'))

    loss = tf.keras.losses.SparseCategoricalCrossentropy(
        from_logits=False, reduction='auto', name='sparse_categorical_crossentropy'
    )

    metrics = ['accuracy']
    optimizer = Adam(0.001)

    media_start = './'

    filepath = media_start + model_name + ".hdf5"

    checkpoint = ModelCheckpoint(filepath, save_freq="epoch", save_best_only=True,
        save_weights_only=True, monitor='val_accuracy', verbose=1)

    csv_logger = CSVLogger(media_start + model_name + "-log.csv")

    model.compile(optimizer = optimizer, loss=loss, metrics=metrics)

    train_steps = len(X_train)//batch_size
    val_steps = len(X_val)//batch_size

    history = model.fit(train_loader,
        steps_per_epoch=train_steps,
        epochs=epochs,
        verbose=1,
        validation_data=val_loader,
        validation_steps=val_steps,
        callbacks=[checkpoint, csv_logger]
    )
```

Figure 4.2: code of EfficientNetB0

In conclusion, the code coordinates the training of a model using specified configurations for epochs, batch size, learning rate, and optimizer on the Mozilla Common Voice dataset.

Final Accuracy: The model correctly identifies 95% of the training samples, achieving a training accuracy of 95%.

Accuracy of Validation: The validation accuracy is 91.266%, demonstrating the model's ability to generalize successfully to new data.

Final Loss: The model's ability to minimize the discrepancy between predicted and actual values during training is indicated by the ultimate loss value on the training data, which equals 0.3046.

Validation Loss: At the end of the training, the validation loss is 0.3344. Lower values indicate better generalization, and this metric quantifies how well the model performs on unobserved data.

We have used 4 models to train and test our models which are ResNet50, EfficientNetB0, InceptionV3, and DenseNet-121. Initiatives are taken such as MFCC and

STFT separately then combining STFT and MFCC together to extract the best features of data to train the CNN models. A brief discussion is given below with each model with each combination

We have run 50 epochs where we use the Adam optimizer with the 0.001 learning rate.

In **ResNet50** with MFCC, figures 4.3 and 4.4, the graph produced by training the ResNet50 model with MFCC features. Our val accuracy is 0.9220 which was found in epoch 50.

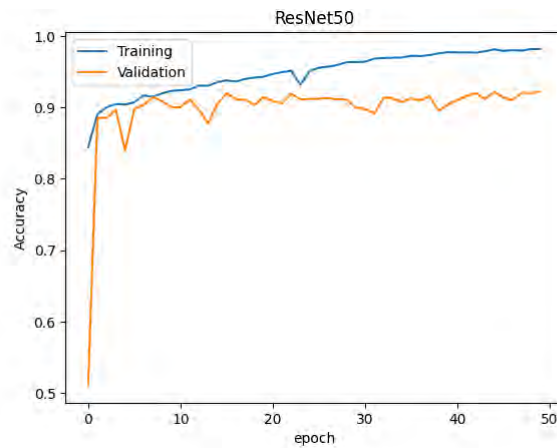


Figure 4.3: Training and validation accuracy curves of ResNet50 from MFCC feature

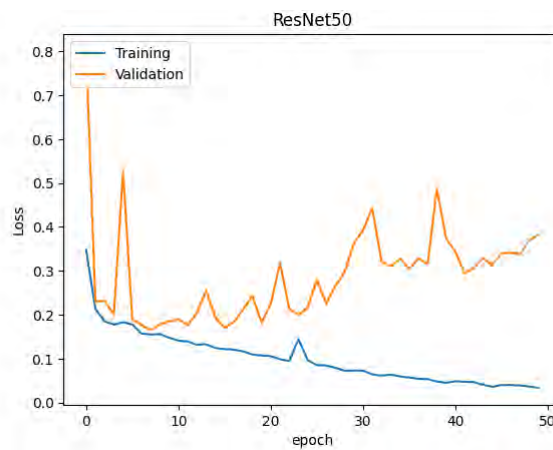


Figure 4.4: Training and validation loss curves of ResNet50 from MFCC feature

Table 4.1 states the precision, recall, f1-score, and support training of the Mozilla common voice dataset with the ResNet50 model. We got an accuracy of 0.91 by training this dataset with ResNet50. Here for Females, the precision value, Recall value & the F1-score are 0.97, 0.84, and 0.94 respectively. Also, for males, the precision value, Recall value & the F1-score are 0.86, 0.98, 0.91 respectively.

Class	Precision	Recall	F1-Score
Female	0.97	0.84	0.90
Male	0.86	0.98	0.91

Table 4.1: Classification report of ResNet50 from MFCC feature

The confusion matrix generated by ResNet50 is given below



Figure 4.5: Confusion matrix of ResNet50 from MFCC feature

In **EfficientNetB0** with MFCC, Figure 4.6 and Figure 4.7, we found the best accuracy rate. Our val accuracy improved is 0.91266. So this is our best model for EfficientNetB0 and we have tested this model.

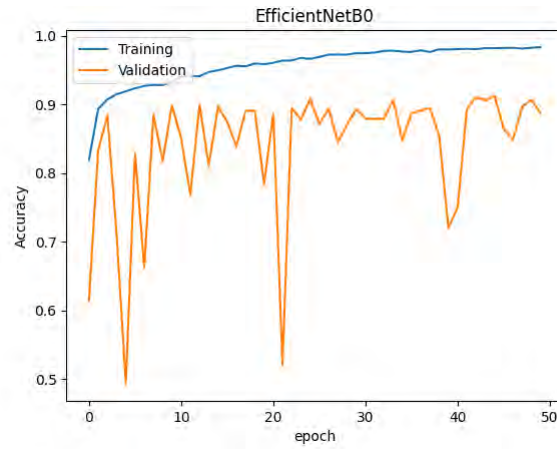


Figure 4.6: Training and validation accuracy curves of EfficientNetB0 from MFCC feature

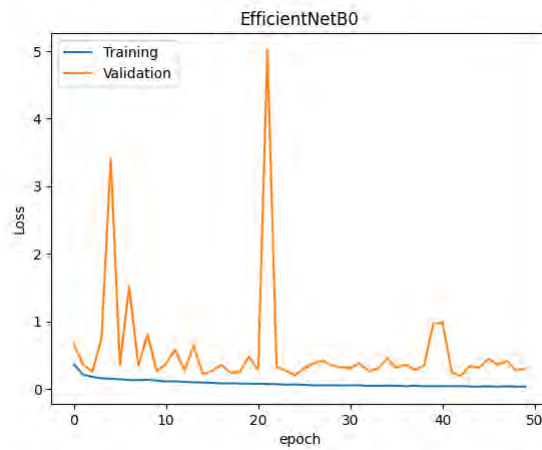


Figure 4.7: Training and validation loss curves of EfficientNetB0 from MFCC feature

Table 4.2 states the precision, recall, f1-score, and support training of our dataset with the EfficientNetB0 model. We got an accuracy of 0.95 by training our dataset with EfficientNetB0. Here for females, we got the score of 0.97, 0.93, and 0.95 and for males, we got 0.93, 0.98, 0.95 as the value of Precision, Recall, and F1-score accordingly.

Class	Precision	Recall	F1-Score
Female	0.97	0.93	0.95
Male	0.93	0.98	0.95

Table 4.2: Classification report of EfficientNetB0 from MFCC feature

The confusion matrix generated by is EfficientNetB0 given below



Figure 4.8: Confusion matrix of EfficientNetB0 from MFCC feature

In **InceptionV3** with MFCC, Figure 4.9 and Figure 4.10, our val-accuracy was also 0.92012 in epoch 48. So this is our best model for InceptionV3 and we have tested through this model.

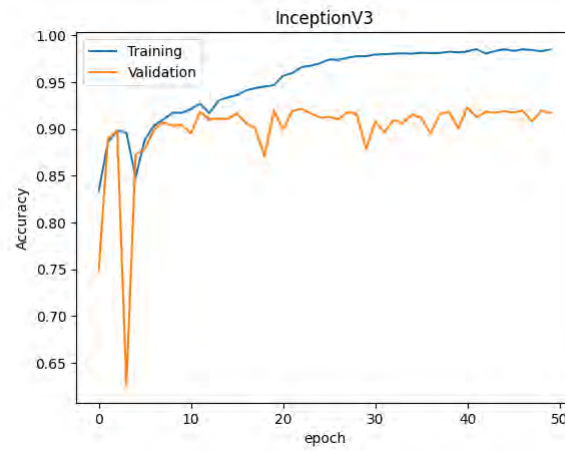


Figure 4.9: Training and validation accuracy curves of InceptionV3 with MFCC feature

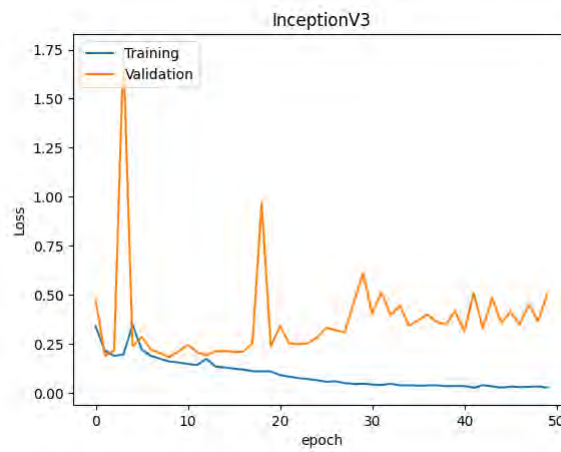


Figure 4.10: Training and validation loss curves of InceptionV3 with MFCC feature

Table 4.3 states the precision, recall, and F1-score, support training our dataset with the InceptionV3 model. We got an accuracy of 0.95 by training our dataset with InceptionV3. Here for females, we got the score of 0.94, 0.98, and 0.96 and for males, we got 0.98, 0.93, 0.95 as the value of Precision, Recall, and F1-score accordingly.

Class	Precision	Recall	F1-Score
Female	0.94	0.98	0.96
Male	0.98	0.93	0.95

Table 4.3: Classification report of InceptionV3 with MFCC feature

The confusion matrix generated by is InceptionV3 given below:



Figure 4.11: Confusion matrix of InceptionV3 with MFCC feature

In **DenseNet-121** with MFCC, Figure 4.12 and Figure 4.13, our val-accuracy was also 0.89021 in the 49th epoch. So this is our best model for DenseNet-121 and we have tested through this model.



Figure 4.12: Training and validation accuracy curves of DenseNet-121 with MFCC feature

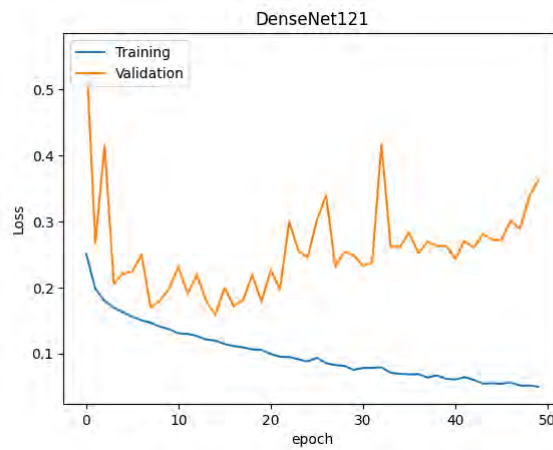


Figure 4.13: Training and validation loss curves of DenseNet-121 with MFCC feature

Table 4.4 states the precision, recall, and F1-score, support training our dataset with the DenseNet-121 model. We got an accuracy of 0.88 by training our dataset with DenseNet-121. Here for females, we got the score of 0.98, 0.79, and 0.87 and for males, we got 0.82, 0.98, 0.90 as the value of Precision, Recall, and F1-score accordingly.

Class	Precision	Recall	F1-Score
Female	0.98	0.79	0.87
Male	0.82	0.98	0.90

Table 4.4: Classification report of DenseNet-121 with MFCC feature

The confusion matrix generated by is DenseNet-121 given below:



Figure 4.14: Confusion matrix of DenseNet-121 with MFCC feature

In **ResNet50** with STFT, figures 4.15 and 4.16, the graph produced by training the ResNet50 model. Our val accuracy is 0.87260 which was found in epoch 43.

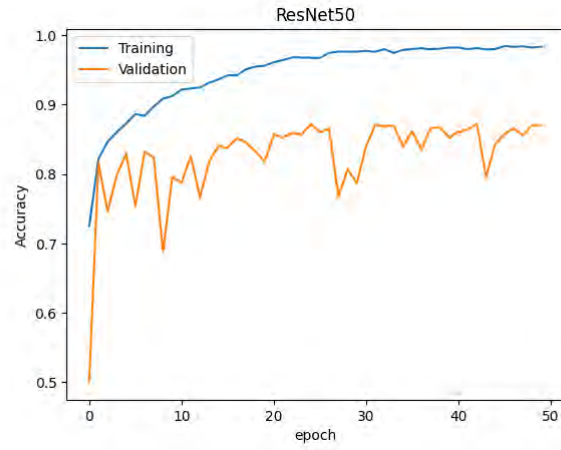


Figure 4.15: Training and validation accuracy curves of ResNet50 from STFT feature

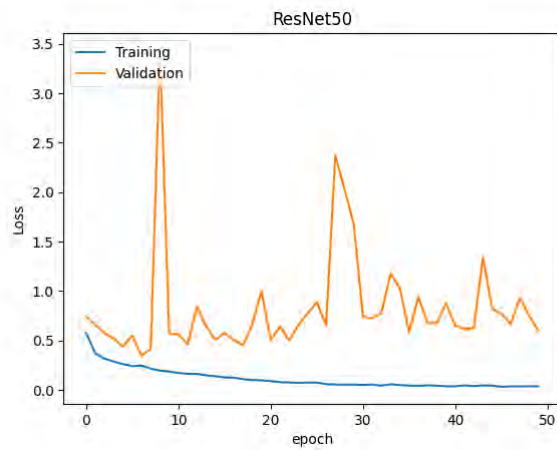


Figure 4.16: Training and validation loss curves of ResNet50 from STFT feature

Table 4.5 states the precision, recall, and f1-score, supporting training our dataset with ResNet50 model. We got an accuracy of 0.91 by training our dataset with ResNet50. Here for females, we got the score of 0.97, 0.85, and 0.91 and for males, we got 0.87, 0.97, 0.92 Precision, Recall, and F1-score accordingly.

Class	Precision	Recall	F1-Score
Female	0.97	0.85	0.91
Male	0.87	0.97	0.92

Table 4.5: Classification report of ResNet50 from STFT feature

The confusion matrix generated by ResNet50 is given below



Figure 4.17: Confusion matrix of ResNet50 from STFT feature

In **EfficientNetB0** with STFT, Figure 4.18 and Figure 4.19, we found the best accuracy rate. Our val accuracy improved is 0.83734 from epoch 48. So this is our best model for EfficientNetB0.

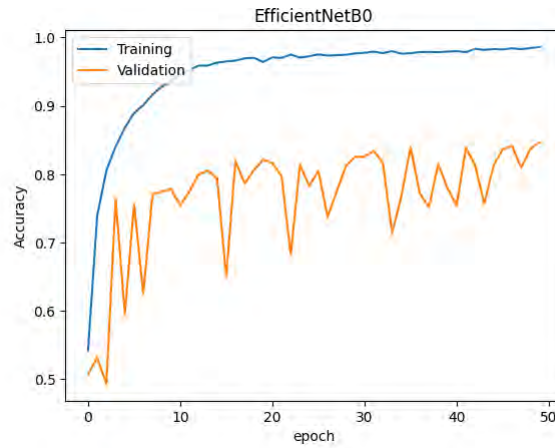


Figure 4.18: Training and validation accuracy curves of EfficientNetB0 from STFT feature

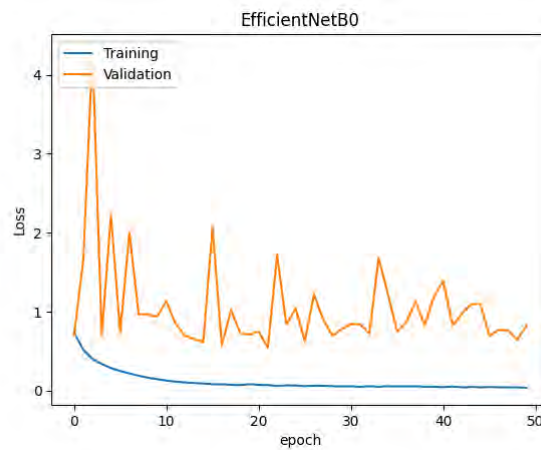


Figure 4.19: Training and validation loss curves of EfficientNetB0 from STFT feature

Table 4.6 states the precision, recall, f1-score, and support training of our dataset with the EfficientNetB0 model. We got an accuracy of 0.89 by training our dataset with EfficientNetB0. Here for females, the values of Precision, Recall, and F1-score, we got 0.96, 0.82, and 0.88 and for males, we got 0.84, 0.96, and 0.90 respectively.

Class	Precision	Recall	F1-Score
Female	0.96	0.82	0.88
Male	0.84	0.96	0.90

Table 4.6: Classification report of EfficientNetB0 from STFT feature

The confusion matrix generated by is EfficientNetB0 given below



Figure 4.20: Confusion matrix of EfficientNetB0 from STFT feature

In **InceptionV3** with STFT, Figure 4.21, and Figure 4.22, our val-accuracy was also 0.87607 in epoch 46. So this is our best model for InceptionV3.

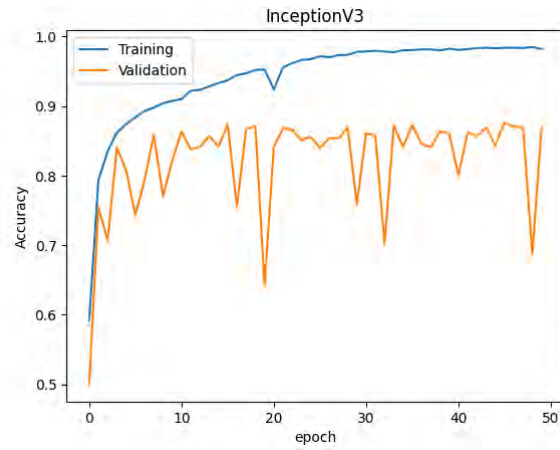


Figure 4.21: Training and validation accuracy curves of InceptionV3 with STFT Feature

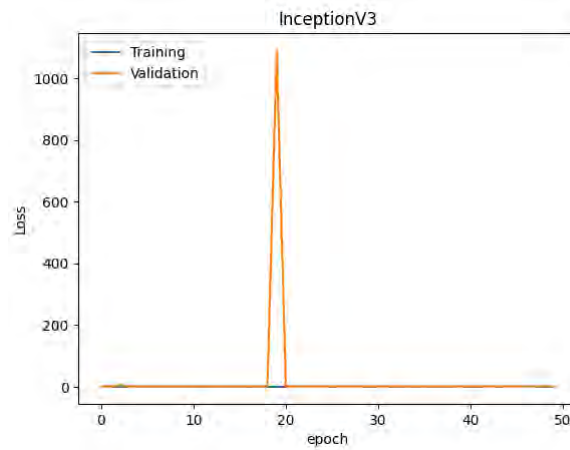


Figure 4.22: Training and validation loss curves of InceptionV3 with STFT Feature

Table 4.7 states the precision, recall, and F1-score, support training our dataset with the InceptionV3 model. We got an accuracy of 0.94 by training our dataset with InceptionV3. Here for females, the values of Precision, Recall, and F1-score, we got 0.96, 0.92, and 0.94 and for males, we got 0.92, 0.96, and 0.94 respectively.

Class	Precision	Recall	F1-Score
Female	0.96	0.92	0.94
Male	0.92	0.96	0.94

Table 4.7: Classification report of InceptionV3 with STFT Feature

Confusion matrix generated by is InceptionV3 given below:

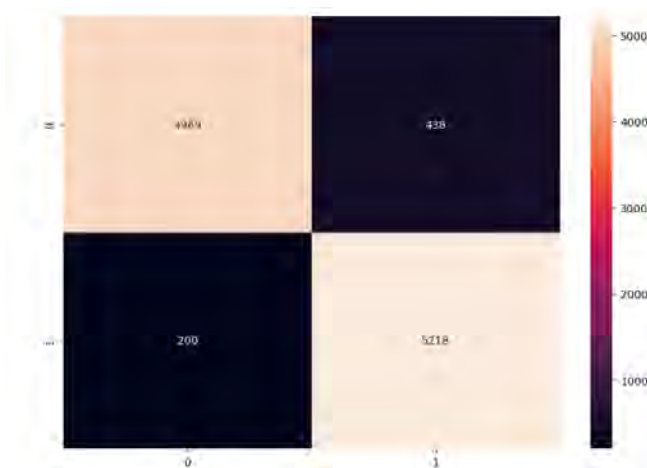


Figure 4.23: Confusion matrix of InceptionV3 with STFT Feature

In **DenseNet-121** with STFT, Figure 4.24 and Figure 4.25, our val-accuracy was also 0.87340 in the 44th epoch. So this is our best model for DenseNet-121 and we have tested through this model.

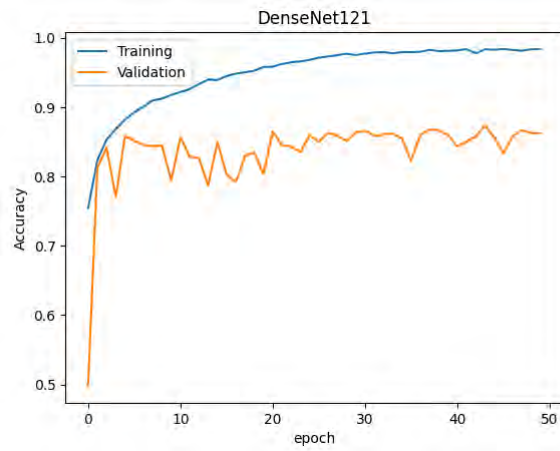


Figure 4.24: Training and validation accuracy curves of DenseNet-121 with STFT

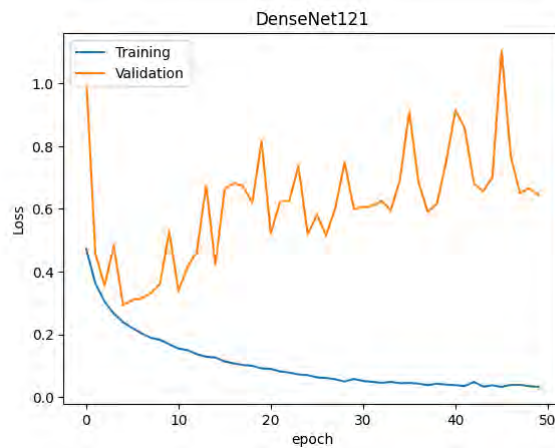


Figure 4.25: Training and validation loss curves of DenseNet-121 with STFT

Table 4.8 states the precision, recall, and F1-score, support training our dataset with the DenseNet-121 model. We got an accuracy of 0.84 by training our dataset with DenseNet-121. Here for females, the values of Precision, Recall, and F1-score, we got 0.96, 0.72, and 0.82 and for males, we got 0.77, 0.97, and 0.86 respectively.

Class	Precision	Recall	F1-Score
Female	0.96	0.72	0.82
Male	0.77	0.97	0.86

Table 4.8: Classification report of DenseNet-121 with STFT

Confusion matrix generated by is DenseNet-121 given below:



Figure 4.26: Confusion matrix of DenseNet-121 with STFT

In **ResNet50** with STFT & MFCC combo, figures 4.27 and 4.28, the graph produced by training the ResNet50 model. Our val accuracy is 0.90652 which was found in epoch 47.

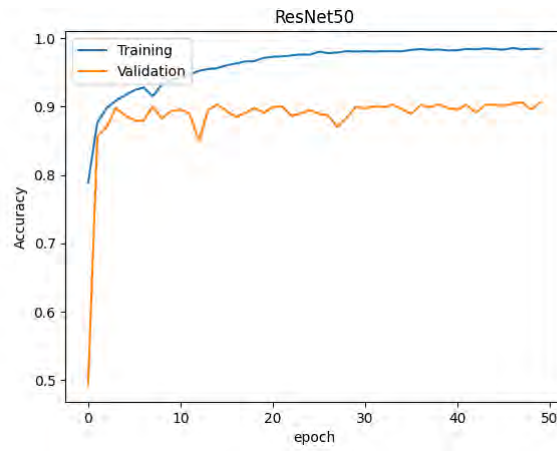


Figure 4.27: Training and validation accuracy curves of ResNet50 from STFT & MFCC combination Feature

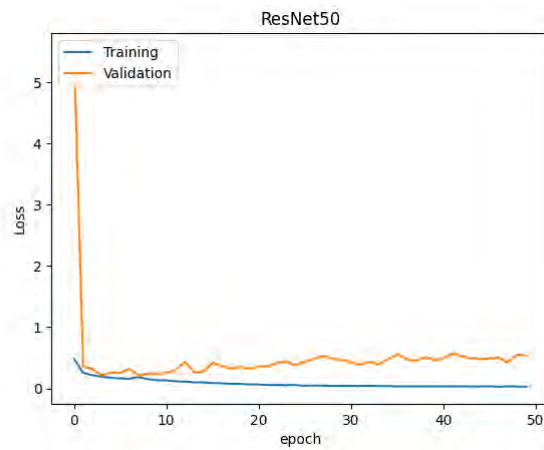


Figure 4.28: Training and validation loss curves of ResNet50 from STFT & MFCC combination Feature

Table 4.9 states the precision, recall, f1-score, and support training of our dataset with ResNet50 model. We got an accuracy of 0.95 by training our dataset with ResNet50. Here for females, the values of Precision, Recall, and F1-score, we got 0.96, 0.94, and 0.95 and for males, we got 0.94, 0.96, and 0.95 respectively.

Class	Precision	Recall	F1-Score
Female	0.96	0.94	0.95
Male	0.94	0.96	0.95

Table 4.9: Classification report of ResNet50 from STFT & MFCC combination

The confusion matrix generated by ResNet50 is given below



Figure 4.29: Confusion matrix of ResNet50 from STFT & MFCC combination

In **EfficientNetB0** with STFT & MFCC combination, Figure 4.30 and Figure 4.31, we found the best accuracy rate. Our val accuracy improved is 0.90438 from epoch 42. So this is our best model for EfficientNetB0 and we have tested this model.

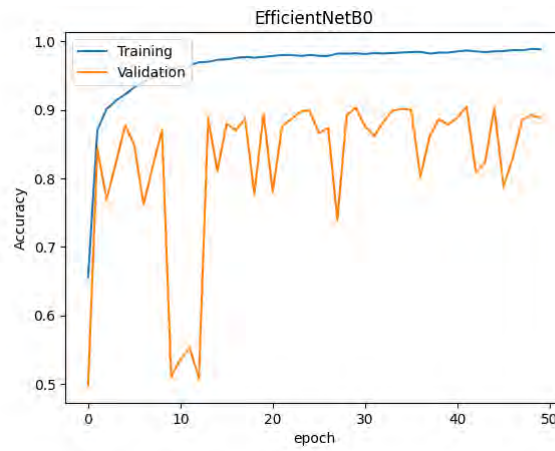


Figure 4.30: Training and validation accuracy curves of EfficientNetB0 from STFT & MFCC combination Feature

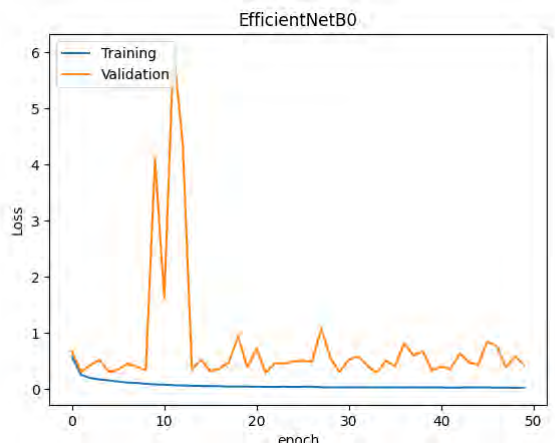


Figure 4.31: Training and validation loss curves of EfficientNetB0 from STFT & MFCC combination Feature

Table 4.10 states the precision, recall, f1-score, and support training of our dataset with the EfficientNetB0 model. We got an accuracy of 0.91 by training our dataset with EfficientNetB0. Here for females, the values of Precision, Recall, and F1-score, we got 0.96, 0.86, and 0.91 and for males, we got 0.88, 0.96, and 0.92 respectively.

Class	Precision	Recall	F1-Score
Female	0.96	0.86	0.91
Male	0.88	0.96	0.92

Table 4.10: Classification report of EfficientNetB0 from STFT & MFCC combination

The confusion matrix generated by is EfficientNetB0 given below



Figure 4.32: Confusion matrix of EfficientNetB0 from STFT & MFCC combination

In **InceptionV3** with STFT & MFCC combo, Figure 4.33 and Figure 4.34, our val-accuracy was also 0.91560 in epoch 41. So this is our best model for InceptionV3 and we have tested through this model.

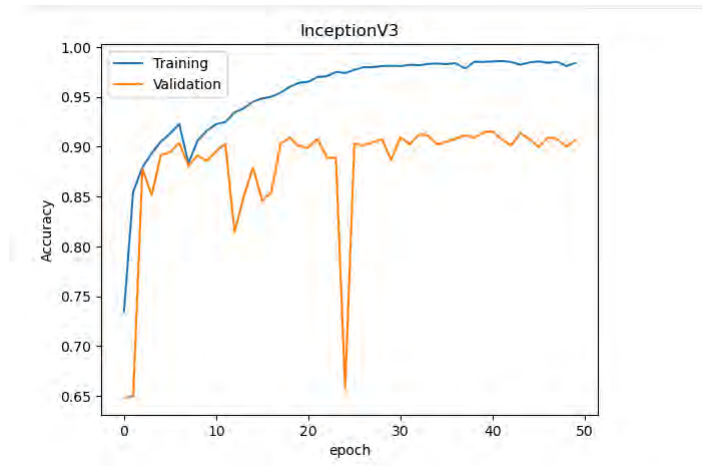


Figure 4.33: Training and validation accuracy curves of InceptionV3 with STFT & MFCC combination Feature

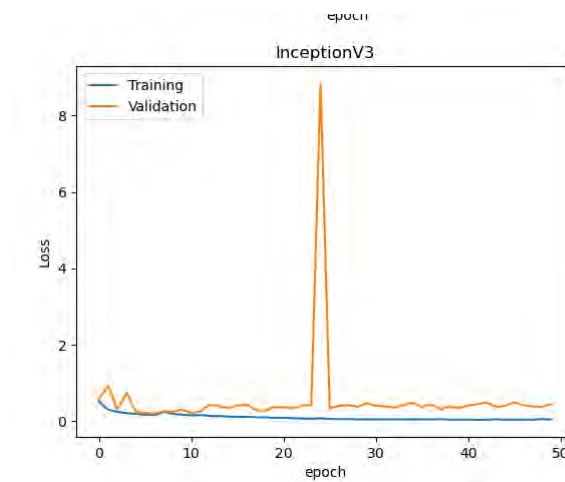


Figure 4.34: Training and validation loss curves of InceptionV3 with STFT & MFCC combination Feature

Table 4.11 states the precision, recall, and F1-score, support training our dataset with the InceptionV3 model. We got an accuracy of 0.94 by training our dataset with InceptionV3. Here for females, the values of Precision, Recall, and F1-score, we got 0.95, 0.93, and 0.94 and for males, we got 0.93, 0.94, and 0.94 respectively.

Class	Precision	Recall	F1-Score
Female	0.95	0.93	0.94
Male	0.93	0.94	0.94

Table 4.11: Classification report of InceptionV3 with STFT & MFCC combination

The confusion matrix generated by is InceptionV3 given below:

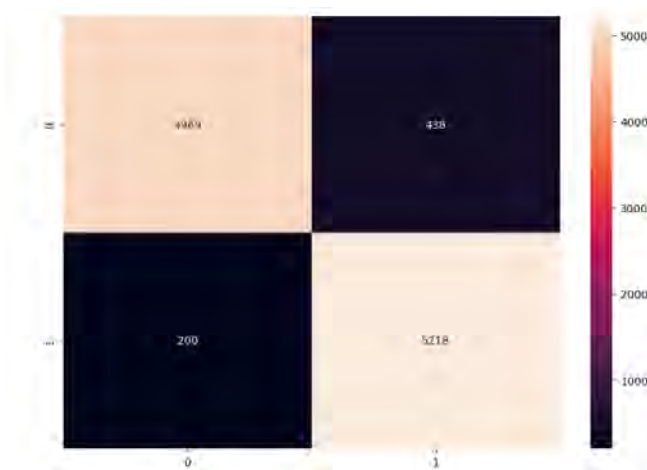


Figure 4.35: Confusion matrix of InceptionV3 with STFT & MFCC combination

In **DenseNet-121** with STFT & MFCC combo, 4.36 and Figure 4.37, our val-accuracy was also 0.91480 in the 44th epoch. So this is our best model for DenseNet-121 and we have tested through this model.

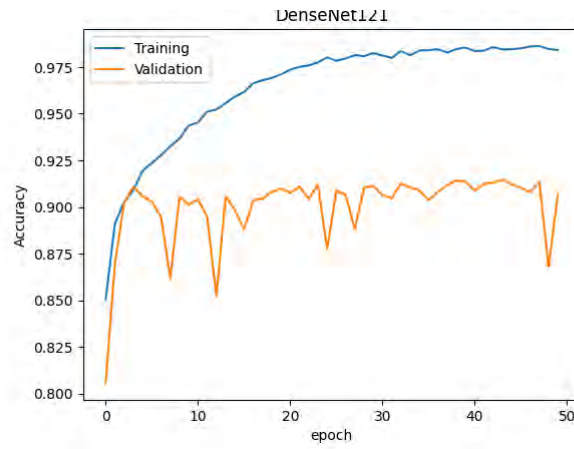


Figure 4.36: Training and validation accuracy curves of DenseNet-121 with STFT & MFCC combination Feature

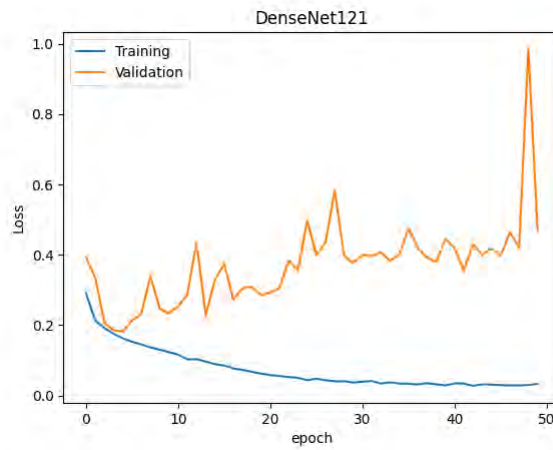


Figure 4.37: Training and validation loss curves of DenseNet-121 with STFT & MFCC combination Feature

Table 4.12 states the precision, recall, and F1-score, supporting training our dataset with the DenseNet-121 model. We got an accuracy of 0.92 by training our dataset with DenseNet-121. Here for females, the values of Precision, Recall, and F1-score, we got 0.95, 0.88, and 0.92 and for males, we got 0.89, 0.95, and 0.92 respectively.

Class	Precision	Recall	F1-Score
Female	0.95	0.88	0.92
Male	0.89	0.95	0.92

Table 4.12: Classification report of DenseNet-121 with STFT & MFCC combination

The confusion matrix generated by is DenseNet-121 given below:



Figure 4.38: Confusion matrix of DenseNet-121 with STFT & MFCC combination

Here in Table 4.13, we tried to compare all the values that we found from the models. For this step, we calculate the weighted average of all models with their inputs.

Models	Inputs	Accuracy	Precision	Recall	F1-Score
ResNet50	MFCC	0.91	0.915	0.91	0.905
ResNet50	STFT	0.91	0.92	0.91	0.915
ResNet50	STFT&MFCC	0.95	0.95	0.95	0.95
EfficientNetB0	MFCC	0.95	0.95	0.955	0.95
EfficientNetB0	STFT	0.89	0.90	0.89	0.89
EfficientNetB0	STFT&MFCC	0.91	0.92	0.91	0.915
InceptionV3	MFCC	0.95	0.96	0.955	0.955
InceptionV3	STFT	0.94	0.94	0.94	0.94
InceptionV3	STFT&MFCC	0.94	0.94	0.935	0.94
DenseNet-121	MFCC	0.88	0.90	0.885	0.885
DenseNet-121	STFT	0.84	0.865	0.845	0.84
DenseNet-121	STFT&MFCC	0.92	0.92	0.915	0.92

Table 4.13: Comparison among all the models

In the above table, we can see that the **InceptionV3** with MFCC feature makes the best score among all the models with their inputs which is 0.95, 0.96, 0.955, 0.955 of Accuracy, Precision, Recall & F1-Score accordingly.

Chapter 5

Conclusion

In the final stretch, our study on gender categorization in Bangla utilizing deep learning-based voice analysis provided light on the complexities of vocal communication and its prospective applications. Our research has emphasized the difficulties presented by the vast variety of tones, emotions, and similarities between male and female voices in the Bangla language. However, our deep learning models, such as ResNet50, EfficientNetB0, DenseNet-121, and InceptionV3, have proved to be effective in identifying gender-specific speech patterns. Model InceptionV3 and Efficient Net B0 have the highest validation accuracy of 95% over other models. Our discoveries have far-reaching ramifications in a variety of fields. Voice analysis' capacity to reliably identify gender opens the possibility to customized offerings, targeted marketing strategies, and enhanced human-computer connection. Furthermore, our research advances the area of speech and language research by expanding our grasp of the intricacies of the Bangla language. We gained insight into the many facets of communication by investigating the link between gender and voice features. We realize that our research has limitations. Our deep learning models' performance may be improved further by continuing to investigate advanced network designs, feature engineering approaches, and data augmentation strategies. As the area evolves, our findings lay the groundwork for future research, promoting additional inquiry and innovation and opening the path for the creation of gender identification systems in other languages and fields.

Bibliography

- [1] URL: https://course.ece.cmu.edu/~ece491/lectures/L25/STFT_Notes_ADSP.pdf.
- [2] URL: https://www.researchgate.net/figure/The-proposed-pipeline-Mel-spectrogram-converts-the-audio-signal-to-an-equivalent-image_fig1_349284934.
- [3] URL: <https://www.semanticscholar.org/paper/A-Comparative-Study-of-Crossover-Operators-for-to-Magalh%C3%A3es-Mendes-Almeida/3bdacac99759ca1c71e241b815ea50226b05af70>.
- [4] Samiul Alam et al. “Bengali Common Voice Speech Dataset for Automatic Speech Recognition”. en. In: *arXiv.org* (2022).
- [5] Rosana Ardila et al. “Common voice: A massively-multilingual speech corpus”. In: *arXiv preprint arXiv:1912.06670* (2019).
- [6] S M Saiful Islam Badhon, Md Habibur Rahaman, and Farea Rehnuma Rupon. “A machine learning approach to automating Bengali voice based gender classification”. In: *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*. Moradabad, India: IEEE, Nov. 2019.
- [7] Sebastian Bittrich et al. “Application of an interpretable classification model on Early Folding Residues during protein folding”. In: *BioData mining 12* (2019), pp. 1–16.
- [8] Mucahit Buyukyilmaz and Ali Osman Cibikdiken. “Voice gender recognition using deep learning”. In: *Proceedings of 2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016)*. Xiamen, China: Atlantis Press, 2016.
- [9] Pramit Gupta, Somya Goel, and Archana Purwar. *A Stacked Technique for Gender Recognition Through Voice*. Aug. 2018. DOI: <https://doi.org/10.1109/IC3.2018.8530520>. URL: <https://ieeexplore.ieee.org/document/8530520>.
- [10] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [11] Hanan Qassim Jaleel, Jane Jaleel Stephan, and Sinan A Naji. “Gender identification from speech recognition using machine learning techniques and convolutional neural networks”. In: *Webology* 19.1 (Jan. 2022), pp. 1666–1688.
- [12] Lakhan Jasuja, Akhtar Rasool, and Gaurav Hajela. “Voice gender recognizer recognition of gender from voice using deep neural networks”. In: *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. Trichy, India: IEEE, Sept. 2020.

- [13] Shaowei Jiang et al. “Transform- and multi-domain deep learning for single-frame rapid autofocusing in whole slide imaging”. In: *Biomedical Optics Express* 9 (Apr. 2018), p. 1601. DOI: 10.1364/BOE.9.001601.
- [14] Renu Khandelwal. *Deep Learning Audio Classification*. Dec. 2020. URL: <https://medium.com/analytics-vidhya/deep-learning-audio-classification-fcbed546a2dd>.
- [15] Ioannis Livieris, Emmanuel Pintelas, and Panagiotis Pintelas. “Gender recognition by voice using an improved self-labeled algorithm”. en. In: *Mach. Learn. Knowl. Extr.* 1.1 (Mar. 2019), pp. 492–503.
- [16] Raz Mohammad Sahar et al. “Performance analysis of ML algorithms to detect gender based on voice”. In: *Recent Trends in Intensive Computing*. Advances in parallel computing. Amsterdam, NY: IOS Press, Dec. 2021, pp. 163–171.
- [17] *Mozilla common voice*. en. <https://commonvoice.mozilla.org/en/datasets>. Accessed: 2023-5-22.
- [18] Anjali Pahwa and Gaurav Aggarwal. “Speech Feature Extraction for Gender Recognition”. In: *International Journal of Image, Graphics and Signal Processing* 8.9 (Sept. 2016), pp. 17–25. DOI: <https://doi.org/10.5815/ijigsp.2016.09.03>.
- [19] E S Parris and M J Carey. “Language independent gender identification”. In: *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. Atlanta, GA, USA: IEEE, 2002.
- [20] A Raahul et al. “Voice based gender classification using machine learning”. In: *IOP Conf. Ser. Mater. Sci. Eng.* 263 (Nov. 2017), p. 042083.
- [21] Shahriar Rahman, Fasihul Kabir, and Mohammad Nurul Huda. “Automatic gender identification system for Bengali speech”. In: *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*. Khulna: IEEE, Dec. 2015.
- [22] Kumar Rakesh, Subhangi Dutta, and Kumara Shama. “GENDER RECOGNITION USING SPEECH PROCESSING TECHNIQUES IN LABVIEW”. In: *International Journal of Advances in Engineering Technology* 51.2 (2011), pp. 51–63. URL: <https://www.ijaet.org/media/0001/6GENDER-RECOGNITION-USING-SPEECH-PROCESSING-TECHNIQUES-IN-LABVIEW-Copyright-IJAET.pdf>.
- [23] Pablo Ruiz. *Understanding and visualizing DenseNets - Towards Data Science*. Oct. 2018. URL: <https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a>.
- [24] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.
- [25] Ramya. T. “Speech emotion recognition”. In: *International Journal for Research in Applied Science and Engineering Technology* 9.1 (2021), pp. 746–749. DOI: 10.22214/ijraset.2021.32740.
- [26] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

- [27] Yavuz Selim Taşpınar et al. “Gender determination using voice data”. In: *Int. J. Appl. Math. Electron. Comput.* (Dec. 2020), pp. 232–235.
- [28] Zhong-Qiu Wang and Ivan Tashev. “Learning utterance-level representations for speech emotion and age/gender recognition using deep neural networks”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, LA: IEEE, Mar. 2017.
- [29] Ergun Yucesoy and Vasif V Nabiyev. “Gender identification of a speaker using MFCC and GMM”. In: *2013 8th International Conference on Electrical and Electronics Engineering (ELECO)*. Bursa, Turkey: IEEE, Nov. 2013.