Spoken Language Identification Using Convolutional Neural Network

BY

SADEQ BIN JAFAR
ID: 201-15-14341
AND

ASIF AL MOTTAKIM
ID: 201-15-14346

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

MOHAMMAD JAHANGIR ALAM
Lecturer (Sr. Scale)
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

July 2024

APPROVAL [CAPITAL LETTER, Bold, Font-14, Alignment-Center]

This Project titled **"Spoken Language Identification Using Convolutional Neural Network"**, submitted by **Sadeq Bin Jafar** and **Asif Al Mottakim** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **July 2024.**

BOARD OF EXAMINERS

Dr. Md. Taimur Ahad (MTA) — **Board Chairman**
Associate Professor & Associate Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mohammad Monirul Islam (MMI) — **Internal Examiner 1**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Jakaria
14.07.2024
Md. Jakaria Zobair (MJZ) — **Internal Examiner 2**
Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

14.07.2024
Nazibur Rahman — **External Examiner**
Technical Lead - Database Administrator
Telenor - Grameen Phone Account

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mohammad Jahangir Alam, Lecturer (Sr. Scale), Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Mohammad Jahangir Alam
Lecturer (Sr. Scale)
Professor & Associate Head
Department of CSE
Daffodil International University

Submitted by:

Sadeq Bin Jafar

Sadeq Bin Jafar
ID: 201-15-14341
Department of CSE
Daffodil International University

Asif Al Mottakim

Asif Al Mottakim
ID: 201-15-14346
Department of CSE
Daffodil International University

©Daffodil International University, Bangladesh *iii*

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Mohammad Jahangir Alam, Lecturer (Sr. Scale)**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Computer Vision*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to the **Head of the Department of CSE**, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Language, the core of human civilization, is all about communication. This research explores the intricacies of Automatic Speech Recognition (ASR) systems, particularly Speech-to-Language Identification (SLID), in multilingual environments. It focuses on data collection, feature extraction, and language classification, using techniques such as Mel spectrograms, MFCC, and deep learning architectures like CNNs and RNNs. Our dataset is from Kaggle and the rest of them are real-time. Deep learning models, particularly CNNs and hybrid architectures like CRNN, show promising results in language identification. An audio speech recognition model is developed using deep learning techniques. Data is stored in Google Drive and accessed via Colab for model training. The model was trained on Mel spectrograms of audio data using deep learning techniques. Pre-trained models like VGG16, Efficientnet, GoogleNet, and DenseNet offer viable alternatives for audio recognition tasks. We are going to propose them on our pre-processed dataset. The performance of all models will be compared using standard metrics like accuracy, precision, recall, and F1 score. This research offers an in-depth analysis of SLID systems, showing their importance in communication across languages and proposing technologies that may be able to increase their accuracy and allow wider acceptance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ASR	Automatic Speech Recognition
SLID	Speech-to-Language Identification
LID	Language Identification
CNN	Convolutional Neural Network
DNN	Deep Neural Network
RNN	Recurrent Neural Network
SVM	Support Vector Machine
MFCC	Mel-Frequency Cepstral Coefficients
GTCC	Gammatone Cepstral Coefficient
ANN	Artificial Neural Network
FFBPNN	Feed-forward Back Propagation Neural Network
CRNN	Convolutional Recurrent Neural Networks
PLDA	Probabilistic Linear Discriminant Analysis
ASC	Acoustic Scene Classification

CHAPTER 1
Introduction

1.1 Overview

The basis of human civilization, that language shapes our interactions and achievements. Speech-to-Language Identification (SLID) is an essential component of Automatic Speech Recognition (ASR) systems, particularly in multilingual environments. While methods like deep learning models and Mel-Frequency Cepstral Coefficients (MFCC) have improved language classification performance, issues with ASR adaptation for different languages and dialects still exist. This points out how important SLID is to ASR systems, which are essential for modern, globally interconnected communication. Even with these improvements, it is still difficult to identify languages accurately, which calls for more study and creativity. While pre-trained Convolutional Neural Network (CNN) models and deep neural networks have the potential to improve accuracy, further research is necessary to improve ASR systems in multilingual contexts. It requires more research in ASR and SLID because it is motivated by the need to address pronunciation variations and enable cost-effective adoption across linguistic contexts. Beyond simple speech recognition, ASR technology can be used for voice assistance, language learning, and improving communication in various environments. Overall, the summary highlights the continued importance of language in human civilization and the critical function that SLID plays in ASR systems. It draws attention to the need for more developments in deep learning and feature extraction methods to solve current problems and open the door to more efficient multilingual communication in the digital age.

1.2 Background & Present State

Various algorithms and methodologies were explored to develop language identification systems for audio data. These included CNN-based SLID models using mel spectrograms, achieving 88% accuracy; a CRNN approach combining MFCC and GTCC features for 92.81% accuracy; hybrid feature extraction using FFBPNN with 93% accuracy; ensemble methods with MFCC and delta features achieving 86%

accuracy; rhythmic feature-based GMM with 70% accuracy; and LSTM RNNs for improved language discrimination. Additionally, DNNs showed promise in language detection, along with i-vector and x-vector techniques coupled with logistic regression or PLDA for enhanced accuracy. Transfer learning with pre-trained CNNs was also effective for sound classification, while ASC systems evolved towards hierarchical approaches for improved precision.

1.3 Problem Statement

The problem statement revolves around the necessity and challenges of Spoken Language Identification (SLID) within Automatic Speech Recognition (ASR) systems, particularly in multilingual environments. Despite advancements in techniques like feature extraction and deep learning models, significant challenges persist, such as adapting ASR for diverse languages and dialects. The problem further extends to the lack of adoption of advanced language classification techniques in regions like India, and Bangladesh, and among local languages due to pronunciation difficulties and linguistic differences. Additionally, the inefficiency of traditional machine learning models compared to deep learning models poses a challenge in achieving higher accuracy in SLID. The study aims to address these challenges and improve the adaptability and accuracy of ASR systems in multilingual environments through innovative methodologies and advancements in deep learning and feature extraction techniques.

1.4 Motivation and Objectives

The motivation behind this thesis is to delve into the realm of Audio Speech Recognition (ASR) and Spoken Language Identification (SLID) research, recognizing their pivotal role as the cornerstone for language-related advancements. ASR not only facilitates transcription and language conversion but also underpins the development of voice-related technologies such as voice commands and assistance. Despite advancements primarily in Western countries, there's a significant gap in adoption across diverse linguistic landscapes like India and Asia. Bridging this gap entails addressing challenges such as accent variations, pronunciation nuances, and computational resource requirements. Deep learning models offer promise, yet

accuracy remains challenging due to linguistic intricacies. Overcoming these hurdles is key to automating language learning, enhancing communication, and facilitating knowledge acquisition globally. This thesis seeks to contribute to refining ASR and SLID systems, driving toward a future where language barriers dissolve, communication transcends boundaries, and learning becomes accessible to all.

1.5 Scope & Limitations

Speech recognition datasets are scarce, especially in the multilingual space. Detecting language out of audio speech requires substantial effort for different languages. Creating a specialized dataset for each language is needed for extensive research. Nevertheless, this is a highly demanding procedure and exceedingly challenging to attain a high-quality dataset. Additionally, the data-gathering process will incur substantial costs for each language. The majority of models likewise fail to achieve high accuracy across all datasets, with most of the accuracy scores being rather low. To start, I preprocess the supplied audio data. Created mel spectrograms for feature extraction due to their optimal compatibility with CNN-based architectures. I converted those Mel Spectrograms features into images. Ultimately, I identified the CNN-based design that yielded superior accuracy compared to previous research.

1.6 Report Organization

This report is structured into some sections. The first section outlines the project, its purpose, its scope, and the expected results.

In the second part, we have done some comparisons where we are pointing out the shortcomings of the current edu-tech platforms and what the gaps are that we identified and that we want to overcome through our project.

The third part is the most important, where we have discussed methodologies, and requirements and done some ground and financial analysis to showcase how we can manage the project and what our goal is to enable every learner to reach their maximum potential by embracing technology and rethinking the conventional educational model. In the fourth its described about implementation & the fifth part is all about results. The

last two parts are all about impact on society, environment and sustainability and future works.

1.7 Summary

The summary outlines the significance of language in human civilization, emphasizing its role in communication and the necessity of Speech to Language Identification (SLID) within Automatic Speech Recognition (ASR) systems. Challenges in multilingual environments and the need for accurate language identification are highlighted, alongside the potential of advanced techniques like deep learning models for improving ASR accuracy. The motivation section underscores the importance of ASR and SLID research, particularly in addressing pronunciation variations and enabling cost-effective adoption in diverse linguistic contexts. The summary also points out the potential applications of ASR technology in voice assistance, language learning, and improving communication in various environments. Overall, the summary underscores the importance of continued research in ASR and SLID for advancing language-related technologies and facilitating effective communication.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

Various algorithms and methodologies were explored to develop language identification systems for audio data. These included CNN-based SLID models using mel spectrograms, achieving 88% accuracy; a CRNN approach combining MFCC and GTCC features for 92.81% accuracy; hybrid feature extraction using FFBPNN with 93% accuracy; ensemble methods with MFCC and delta features achieving 86% accuracy; rhythmic feature-based GMM with 70% accuracy; and LSTM RNNs for improved language discrimination. Additionally, DNNs showed promise in language detection, along with i-vector and x-vector techniques coupled with logistic regression or PLDA for enhanced accuracy. Transfer learning with pre-trained CNNs was also effective for sound classification, while ASC systems evolved towards hierarchical approaches for improved precision

2.2 Related Works

The author developed a machine-learning system to detect language from audio clips by unknown speakers using a Convolutional Neural Network (CNN). They used the Kaggle spoken language detection dataset and converted audio speech files into mel spectrogram images. The CNN model achieved an accuracy of 88%[1], demonstrating the effectiveness of image formats and the Bernoulli Naive Bayes approach.Adal A. A. et al. developed a hybrid convolutional Recurrent Neural Network (CRNN)-based architecture for spoken language identification (SLID) using deep learning advancements. The system uses sequential feature vectors and combines Mel Frequency Cepstral Coefficient (MFCC) and Gammatone Cepstral Coefficient (GTCC) features for improved accuracy. [2]The study examines the effectiveness of a language identification system using hybrid characteristics and artificial neural network (ANN) learning techniques. It uses RASTA-PLP features and merges advanced MFCC features with RASTA-PLP features. The study achieves classification accuracy of 93%, emphasizing the need for suitable feature sets to improve LID system

performance.[3]The authors advance SLID by using MFCC and delta features in audio data. They used ensemble methods and four ML algorithms. The model achieved an accuracy rate of 86% across five languages. However, there's room for improvement with minimal data preprocessing and using parameter and regularization techniques for higher accuracy.[4]The study introduces a language identification system using acoustic features of audio data to capture rhythmic features of vowels and consonants. The authors used a Gaussian Mixture Model (GMM) with 70% accuracy, and developed different language identification approaches, including SVM and i-vector, to ensure the best use of the feature in classification. [5]This study explores the use of deep learning in audio signal classification. It surveys five deep learning models: CNN, RNN, and hybrid models. RNN is used for temporal pattern recognition, while CNN is used for speaker identification and speech recognition. The hybrid model combines CNN and RNN and CNN-SVM. Accuracy varies based on the dataset and model, with CNN often providing better results. The study guides further research on deep learning. [6]The paper discusses the importance of communication techniques and the human-computer interface in voice recognition, analyzing technical advancements and determining the best approach for building a Marathi language-based human-computer interaction system, highlighting MFCC as the widely used technique.[7]The study addresses the challenge of machine learning and AI-based speech recognition in multilingual environments. It proposes a new algorithm combining binary-bat-algorithm (BBA) with late-acceptance-hill-climbing (LAHC) to extract the best features from audio data. The algorithm achieved a 92.35% accuracy, outperforming meta-heuristic FS algorithms. Future research could explore different transfer and cost functions, apply the algorithm to more languages, and address long-term speech signal dependencies. [8]A. Draghici et al. conducted a study on Spoken Language Identification (SLID) using deep neural network architectures. They found that both CNN and CRNN can learn specific patterns from speech audio, with CNN achieving 86% accuracy on YouTube News and CRNN 83% [9]accuracy on EU Repo. The authors introduce a novel method for automatic language identification (LID) using long short-term memory (LSTM), an upgraded type of Recurrent Neural Network. The LSTM RNNs outperform baseline i-vector and feed-forward Deep Neural Network systems on the NIST Language Recognition Evaluation 2009 dataset, achieving higher accuracy with fewer parameters. The study suggests further integration and dataset

investigation for further development. [10]The authors use Deep Neural Networks for language classification, comparing their model with traditional methods. They show gains of up to 70%[11] in Cavg over baseline systems using a fully connected feed-forward neural network with rectified linear activation functions. The study concludes that DNNs are a powerful tool for improving automatic language detection, especially when large amounts of training data are available. The authors used i-vectors and x-vectors to capture audio data features, evaluating them using probabilistic linear discriminant analysis (PLDA) and logistic regression. They used a dataset from Indonesia, including Bahasa Sunda, Jawa, and Minang, and divided the data into segments. The study found that logistic regression improved x-vector accuracy, while i-vector performance was better when using the LR classifier. [12]The study uses spoken language to classify audio data from European radio broadcasts using VPM and LD discriminating systems. Training pairs are generated using annotated and segmented audio samples. The authors propose using various supervised and unsupervised classification methods simultaneously, achieving a commendable discrimination rate for speech signals. [13]The study evaluates the performance of three image-trained CNNs and two sound-trained CNNs in sound classification across datasets like UrbanSound8K, ESC-10, and Air Compressor. Results show Sound CNNs outperform Image CNNs, achieving high accuracy. Future work may involve fusion techniques and addressing sound data challenges, demonstrating the effectiveness of transfer learning in sound classification. [14]Acoustic Scene Classification (ASC) has evolved from RASTA analysis to recurrent neural networks, k-nearest neighbor, and hidden Markov models.[15] Early systems used the 'bag-of-frames' method, while modern systems prioritized hierarchical classification. This study examines the progression of ASC research, from initial techniques to current hierarchical approaches, focusing on high-level generic categories and correct classifications within broad categories.

2.3 Comparison between existing works

I evaluated prior research about our objective. I have studied both different machine learning algorithms as well as different deep learning algorithms to find out which one gives better accuracy. This field of identifying language out of audio speech can be correlated to NLP, but I am taking the deep learning approach for this. For this review,

I have examined many research works that investigate the application of different machine learning techniques in audio speech recognition systems. Some of the crucial aspects for comparison encompass:

In feature extraction, Mel spectrograms are commonly employed for their ability to visually depict frequency and temporal data and have demonstrated impressive accuracy in convolutional neural network (CNN) models, obtaining an 88% accuracy rate [1].MFCCs and Gammatone Cepstral Coefficients are highly effective in collecting both spectral and phonetic information. When used with CRNNs, they achieve an accuracy of 92.81%, as reported in [2]. The characteristics of RASTA-PLP include utilizing MFCCs in conjunction with feed-forward backpropagation neural networks, leading to enhanced performance and achieving an accuracy of 92% [3].Characteristics of a hybrid: By combining Mel-frequency cepstral coefficients (MFCCs) with delta features (DFCCs) and utilizing ensemble methods, such as random forests or bagging, an accuracy of 86% was obtained for various languages [4]. Examining vowel and consonant patterns using Gaussian Mixture Models resulted in a 70% accuracy rate in terms of rhythmic aspects. There is potential for improvement by employing SVMs and i-vectors, as suggested by [5].

In model preparation, For model training, different machine-learning approaches were taken based on different feature extraction techniques. Different features yield different accuracy based on the machine learning technique applied. For mel-spectrograms, CNN and CRNN are mostly used to achieve higher accuracy. Deep learning can provide up to 70% better accuracy than normal machine learning algorithms. For MFCC, DFCC, and their combined features, CRNN is mostly used. Various pre-trained models like VGGish, Google-net, and Shuffle Net can also produce high accuracy on both mel spectrograms and MFCC. Different ensemble methods, like random forests and bagging, are also shown to produce higher accuracy. SVM and GMM are also used in many cases. The accuracy based on different models and feature extraction techniques is stated below in the table based on previous studies:

Table 2.3.1 Comparison of previous studies along with results.

Year	Authors	Features	Approach	Best Algorithm	Accuracy
2021	Gundeep Singh , Sahil Sharma ,Vijay Kumar , Manjit Kaur , MohammedBaz , and Mehedi Masud [1]	Mel spectrograms	CNN	CNN	88%
2022	Adal A. Alashban *, Mustafa A. Qamhan, Ali H. Meftah and Yousef A. Alotaibi [2]	MFCC, GTCC, and Combination of GTCC MFCC	CRNN	CRNN	92.81%
2021	Pardeep Sangwan , Deepti Deshwal , Naveen Dahiya [3]	MFCC features with RASTA-PLP features	Feed-forward Back Propagation Neural Network (FFBPNN)	Feed-forward Back Propagation Neural Network (FFBPNN)	92.6%
2020	Dilip Singh Sisodia,Saragadam Nikhil, G. Sai Kiran, P. Sathvik [4]	MFCC and DFCC	Ensemble	Ensemble method	86%
2020	Hwamin Kim 1 and Jeong-Sik Park [5]	r-vector	Gaussian Mixture Model (GMM)	Gaussian Mixture Model (GMM)	70%

2020	ANKIT DAS, SAMARPAN GUHA ,PAWAN KUMAR SINGH ,ALI AHMADIAN NORAZAK SENU, AND RAM SARKAR [8]	MFCC and LPC	Support Vector machine (svm)	Support Vector machine (svm)	92%
2020	Alexandra Draghici,Jakob Abeßer,Hanna Lukashevich [9]	Mel spectrograms	CNN, CRNN	CNN	86%
2020	P. Rangan, S. Teki, and H. Misra [20]	log-Mel	CNN-LSTM	CNN-LSTM	79.02%
2021	E. Salesky, B. M. Abdullah, S. Mielke et al. [19]	MFCC	CNN ResNet50 RNN	CNN	53%
2021	Andrei Shcherbakov,Liam Whittle,Ritesh Kumar, Siddharth Singh,Matthew Coleman,Ekaterina Vylomova [17]	MFCC	CNN LSTM	CNN	74%
2020	R. van der Merwe [22]	Log-Mel	CRNN ResNet50 DenseNet121	CRNN	89%

2.4 Open Issues

Despite significant advancements in Automatic Speech Recognition (ASR) and Speech-to-Language Identification (SLID) systems, several open issues remain. One major challenge is the integration of open-source technologies with proprietary solutions to create robust, adaptable SLID systems. Leveraging open-source datasets, such as those available on Kaggle, along with real-time data collection, presents opportunities and challenges in ensuring data diversity and representativeness across various languages and dialects. Additionally, the effective integration of newly gained deep learning techniques with traditional methods like Mel-Frequency Cepstral Coefficients (MFCC) is essential for improving the accuracy and adaptability of SLID systems.Identifying real-life complex engineering problems, such as ASR adaptation for diverse linguistic contexts, requires a holistic approach. It involves understanding pronunciation variations and developing cost-effective, scalable solutions. This necessitates economic evaluation and cost estimation, particularly in terms of computational resources required for training deep learning models like CNNs, RNNs, and CRNNs on large datasets. The use of pre-trained models such as VGG16, EfficientNet, GoogleNet, and DenseNet can reduce development costs but requires careful evaluation to ensure they meet the specific needs of multilingual SLID applications. Addressing these open issues is critical for enhancing SLID systems' performance and facilitating their broader adoption in various real-world applications.

2.5 Summary

Researchers have explored various methodologies for SLID, audio signal classification, and acoustic scene classification (ASC), achieving accuracy ranging from 86% to 93%. They also studied feature extraction techniques, deep learning models, and transfer learning methodologies. The studies contribute to advancements in these areas, offering insights into feature extraction, deep learning model architectures, and transfer learning methodologies. Further research is needed to improve model accuracy and address challenges in multilingual environments and real-world applications. audio data categorization and Automatic Speech Recognition (ASR) performance. The research will analyze the advantages and disadvantages of machine learning and deep learning approaches, providing valuable information for future model development.

CHAPTER 3
METHODOLOGY

3.1 Overview

The process of training a deep learning model to identify spoken languages from audio data involves several steps. The first step is data gathering, which involves collecting audio data from files. The data is then pre-processed, converting the files from FLAC to WAV format using tools like ffmpeg and PyChi. Feature extraction is performed, creating Mel spectrograms and saving them as PNG images. The data is then converted into a torch format for use in a deep learning model. The model is trained using various hyperparameters, such as VGG16, GoogleNet, EfficientNet and DenseNet. The model is evaluated on its accuracy in classifying spoken languages, but challenges such as pre-processing can negatively affect its accuracy. The summary effectively summarizes the key steps of the methodology while maintaining essential information about spoken language identification using deep learning. To effectively conduct audio speech classification research, researchers need a diverse dataset with multiple languages, speakers, accents, and recording environments. Data preprocessing and feature extraction are crucial, and a powerful CPU and GPU are needed for data handling and model training.

3.2 Proposed Methodology/System Design

Upon gathering the data, I converted all the audio files from the flac format to the wav format. I employed the pydub library in conjunction with ffmpeg to accomplish this task. Following that, the process of extracting features takes place. I transformed all the audio files into Mel spectrograms and stored them as images in the png format. Subsequently, I performed picture preprocessing to enhance accuracy. It is essential to ensure that preprocessing improves accuracy since it can sometimes have a detrimental effect on the model. Additionally, the dataset is translated into a torch in a parallel manner. Subsequently, the converted dataset was incorporated into the model for the purpose of training. I experimented with several hyperparameters in order to get

improved accuracy. For model, I have used VGG16, GoogleNet, Densenet EfficientNet.

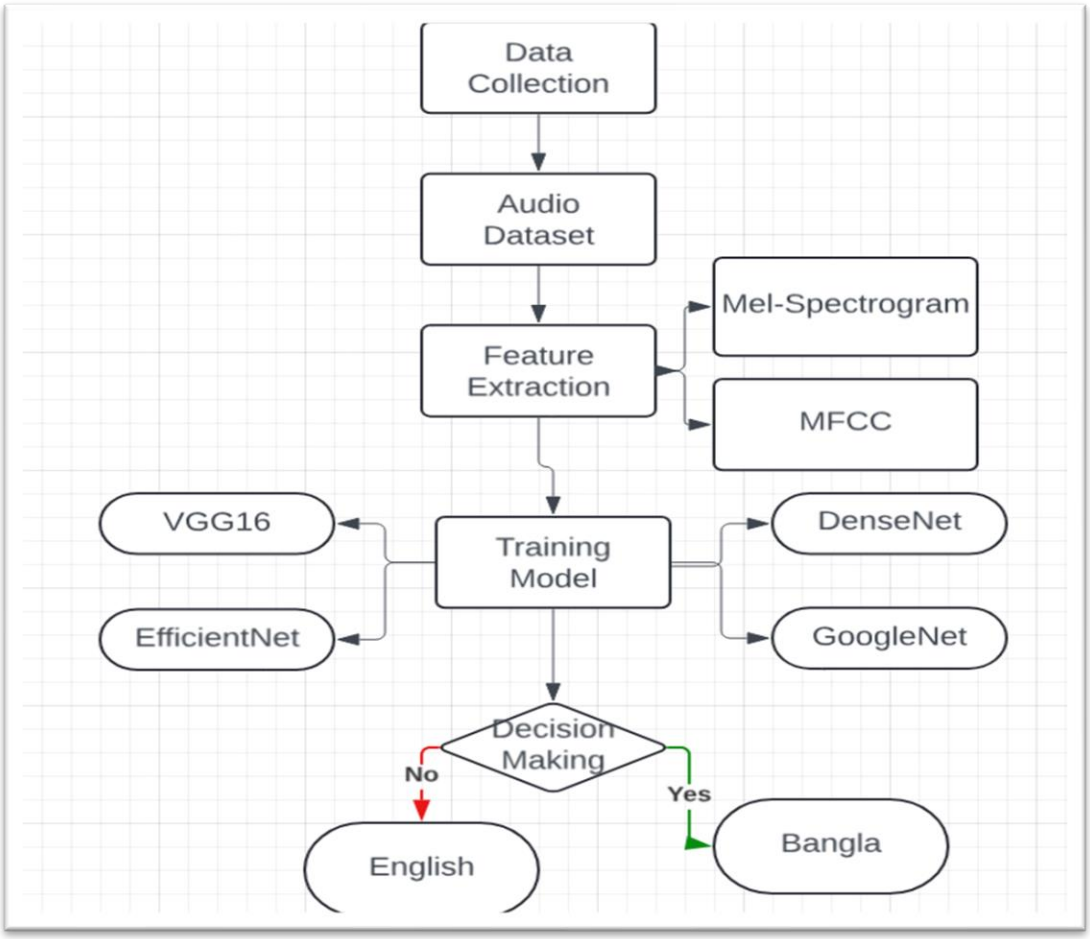


Figure 3.2.1: Proposed Methodology

3.3 Requirement Analysis (Hardware/ Software Requirement)

Dataset Acquisition: Collect an appropriate dataset of audio recordings covering multiple languages, diverse speakers, accents, and recording environments. Ensure the dataset is sufficiently large and varied to train and evaluate the classification model effectively.

Data Preprocessing and Feature Extraction:

- Utilize FFMPEG for audio file manipulation and conversion if necessary.
- Implement preprocessing techniques to clean and normalize audio data.

- Extract features such as Mel Frequency Cepstral Coefficients (MFCC), Delta features, or other relevant acoustic features.

Hardware Requirements:

- Powerful CPU: To handle the computational tasks involved in data preprocessing, feature extraction, and model training.
- Powerful GPU: Accelerate deep learning model training processes, particularly if utilizing frameworks like PyTorch.
- Adequate RAM: A minimum of 8 GB RAM is required to handle large datasets and computational tasks efficiently.

Software Requirements:

- FFMPEG: For audio file manipulation, conversion, and processing.
- Python: A programming language for implementing the classification model and related tasks.

Essential Libraries and Frameworks:

- NumPy: For numerical computing and array operations.
- Matplotlib: For data visualization and plotting.
- Seaborn: For statistical data visualization, if required.
- Scikit-Learn: For implementing machine learning algorithms and evaluation metrics.
- Pandas-data analysis and manipulation library
- ImageNet-ImageNet is a large-scale dataset of labeled images designed for training and benchmarking computer vision algorithms.
- OpenCV-OpenCV is an open-source computer vision and image processing library used for tasks such as object detection, image enhancement.

By fulfilling these implementation requirements, the research on audio speech classification can proceed effectively, ensuring proper data handling, preprocessing, feature extraction, model implementation, and evaluation.

3.4 Project Management and Financial Analysis

Project management for the machine learning project involved setting clear goals and objectives, chosen after studying various topics and algorithms with my supervisor. We utilized software like Google Calendar, Gmail, and Zoom for organization and communication. Regular discussions with my supervisor addressed potential risks and challenges, allocating tasks and time effectively. After achieving our objectives, we proceeded to work on the research paper. Financial planning included securing investments and grants, allocating resources, and recognizing supplementary needs like specialized datasets. Throughout the project, we maintained a focus on data privacy, bias mitigation, and intellectual property rights

Project Objectives: Using audio data to use a model for language classification is more accurate, faster, and uses less hardware resources.

- developing a model to classify audio data.
- comparison between different deep learning architectures to find out the best one.
- challenges and limitations of language classification from audio data.

Project Timeline:

- Phase 1: Planning and Research (2-4 weeks)
- Phase 2: Data collection and conversation through Mel-spectrogram (3-5 weeks)
- Phase 3: Data labeling, and preprocessing (8-10 weeks)
- Phase 4: Model training and applying algorithm (10-12 weeks)
- Phase 5: Model test and output analysis (4-6 weeks)
- Phase 6: Front-End Development (5-6 weeks)
- Phase 7: Back-End Development (6-8 weeks)
- Phase 8: Testing (4-6 weeks)
- Phase 9: Deployment (2-4 weeks)
- Phase 10: Post-Launch and Marketing (Ongoing)

Resource Planning: Effectively allocating human, financial, and technological resources to ensure the timely completion of project tasks is essential. Assigning roles and responsibilities to team members based on their expertise and availability is key.

Risk Management: Identifying potential risks and uncertainties that may impact project success and developing mitigation strategies to address them is important. Regularly monitoring and evaluating project risks throughout the project lifecycle is necessary.

Communication Plan: Establishing clear communication channels and protocols to facilitate collaboration and information exchange among project stakeholders is vital. Holding regular meetings, progress reviews, and status updates to keep stakeholders informed is recommended.

Quality Assurance: Implementing quality assurance processes and standards to ensure that project deliverables meet or exceed predefined quality criteria is imperative. Conducting regular quality checks and audits to identify and rectify any deviations from standards is essential.

Financial Analysis: Financial analysis is a key component to complete the theses paper The first factors to consider when investing are the cost of producing films, the technical infrastructure, and the development of AI-driven features. Operational expenses, including hosting and maintenance, are carefully estimated. Possible cooperative partnerships, course sales, and subscription fees are a few examples of revenue streams. The most effective way to distribute resources and reduce risks is to do a thorough cost-benefit analysis.

A general table showing the approximate costs of the various spoken language identification project components is provided below. We have to choose based on our requirements.

Table 3.4.1. Estimated Cost Of The Project

SL No.	Expense Description	Amount(BDT)
01	Domain and Hosting	4,500-6,000
02	Data set collection(raw)	25,000-30,000
03	GPU (minimum 1650 SUPER 4 GB Ram)	21,000-25,000
04	RAM 16 GB (3200 BUS) x 2	8,000-10,000
05	CPU AMD Ryzen 5 5600	13,000-15,000
06	Documentation and Report Writing	2000-2500
07	Transportation Cost	5000-6000
08	Yearly Maintenance Cost	60,000-70,000
09	Miscellaneous (e.g., licenses, tools)	3,000-4,000
10	Contingency (10% of total)	14,150-16,850
	Total	1,41,500-1,68,500

3.5 Summary

The methodology for training a deep learning model for spoken language identification starts with data gathering, involving the acquisition of diverse audio recordings. Preprocessing follows, utilizing FFMPEG for file manipulation and extracting features like MFCC. Hardware requirements include a powerful CPU and GPU, while software necessities involve FFMPEG and Python. Libraries like PYdub, OpenCV, ImageNet and Scikit-learn aid in manipulation and model building. The proposed methodology involves converting audio to WAV, extracting Mel spectrograms, and preprocessing images. Data are transformed into torch format for model training, experimenting with various hyperparameters and employing models like VGG16. Data collection prioritizes quality, sourced from platforms like Kaggle, and organized into folders by language. This comprehensive approach ensures a robust pipeline from data acquisition to model evaluation, enhancing the efficacy of spoken language identification systems.

CHAPTER 4
IMPLEMENTATION

4.1 Overview

In data preparation, where audio files in FLAC format are converted to WAV format using the pydub library and ffmpeg. This conversion is necessary because the WAV format is more widely supported and less data-heavy. After converting the audio files, feature extraction is performed, transforming the audio into mel spectrograms, which are then saved as PNG images. This step allows the audio data to be processed by convolutional neural networks (CNNs).

- During the data preprocessing phase, efforts were made to create a high-quality dataset. The audio data collected needed to be clear and varied in pitch and speed to account for differences in how people speak. The initial conversion from FLAC to WAV facilitated easier handling and reduced data size.
- Feature extraction techniques included mel spectrograms, mel-scaled filter banks, and Mel-Frequency Cepstral Coefficients (MFCCs). Mel spectrograms were chosen for this project as they effectively capture the audio characteristics in a visual format suitable for CNNs.
- The waveform representation, which visualizes sound wave movement, was discussed as having key attributes: wavelength, amplitude, frequency, and velocity. Mel spectrograms use a frequency-domain filter bank to process audio inputs over time, producing a visual representation that highlights frequency components. MFCCs provide a short-term power spectrum of sound through a logarithmic power spectrum and a linear cosine transform. Mel-scale filter banks analyze frequency characteristics in a way that mimics human hearing sensitivity.
- For model training, a supervised learning approach was employed, leveraging various CNN architectures. The models used included VGG16, EfficientNet,GoogleNet, and GoogleNet. These models were compared to evaluate their performance and accuracy in processing the transformed audio data.

Overall, this project demonstrates a comprehensive workflow from data preparation and feature extraction to model selection and training, aiming to achieve high accuracy in audio classification tasks using advanced deep learning techniques.

4.2 Train Model/ Prototype Design

Data Collection Procedure

I have extensively looked for datasets on the internet to find out the best set of datasets. Creating a machine learning system for the automatic speech detection dataset plays a crucial role, especially in the quality of the data. The higher the quality of data, the more accurate the model can perform. So, Optimizing the balance and reliability of the dataset is a must for the enhancement of the efficacy of machine learning algorithms, resulting in impeccable machine training. I gave paramount importance to data collection so that I could construct a dataset with more careful consideration for better equilibrium and dependability. I have collected my dataset from famous data science resources and tools platform Kaggle. The dataset name is spoken language identification. Then, the dataset is plotted into two different folders, Bangla and English, for my dataset. I have taken 1030 data out of the original folder to create my own dataset.

Table 4.2.1: Glimpse at the Dataset

Audio files	Labels
de_f_5d2e7f30d69f2d1d86fd05f3bbe120c2.fragment1	English
de_f_5d2e7f30d69f2d1d86fd05f3bbe120c2.fragment2	English
de_f_5d2e7f30d69f2d1d86fd05f3bbe120c2.fragment3	English
de_f_5d2e7f30d69f2d1d86fd05f3bbe120c2.fragment4	English

es_f_1d27c6d589eeff17973ffd0b7a77a70a.fragment1	Bangla
es_f_1d27c6d589eeff17973ffd0b7a77a70a.fragment2	Bangla
es_f_1d27c6d589eeff17973ffd0b7a77a70a.fragment3	Bangla
es_f_1d27c6d589eeff17973ffd0b7a77a70a.fragment4	Bangla

Here, in English and Bangla, two different class files are saved into two different folders. Then, the data is converted into Mel spectrograms and saved in Google Drive. From then on, the drive-specific folder will work as a further dataset for my research and code in Colab. There is 180 data in the dataset, which is used for testing purposes.

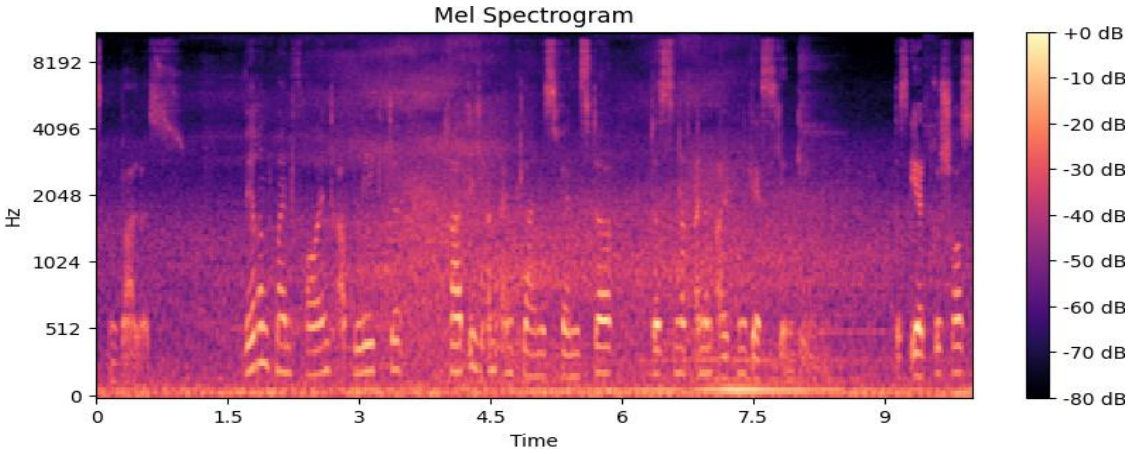


Figure 4.2.1: Mel Spectrogram of the Audio Speech

Data Preprocessing

Here, at first, I tried to create a dataset that is good. I have searched through various websites to collect the best data to create a better model. The dataset's audio should be clear. Besides that, the same audio data should be in different pitches and different speeds because different people have different talking speeds and pitches. The audio was in the FLAC format. Then, I converted it into WAV format because FLAC format is not supported by most audio players and many Python libraries. Besides that, it is

quite heavy and takes up more data. After converting the audio data into WAV format then, I started working on the feature extraction.

```
from IPython.display import Audio
Audio(audio_path)
```

Figure 4.2.2 Audio File

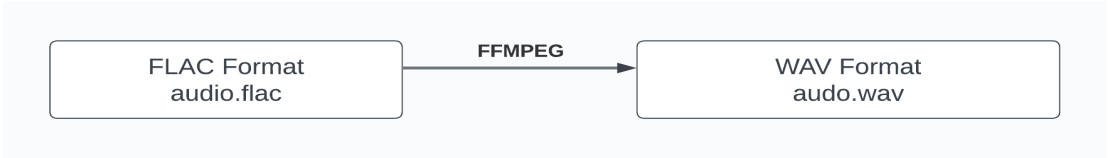


Figure 4.2.3: Audio converting from FLAC to WAV

Feature Extraction

There are a lot of feature extraction techniques out there. Among them, Mel spectrogram, Mel scaled filter bank, and MFCC are some of the most popular ones. After the extraction of the features, I converted them into image format using Matplotlib so that we could easily fit the image into different CNN-based deep learning algorithms to get better accuracy. For this paper, I have used the Mel spectrogram. Audio can be represented in many different forms.

Waveform: A waveform is a visual depiction of a sound wave's movement across a medium over some time. Every waveform has four essential attributes: Wavelength, amplitude, frequency, and velocity.

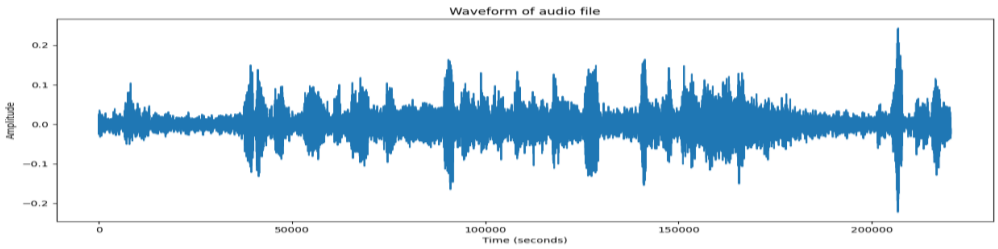


Figure 4.2.4: Visual Representation of the Audio

Mel-Spectrogram: A frequency-domain filter bank is used by the Mel Spectrogram function to handle audio inputs that have been divided into time frames. The second and third output arguments from the Mel Spectrogram function can be used to retrieve the center frequencies of the filters and the time instants corresponding to the analysis windows.

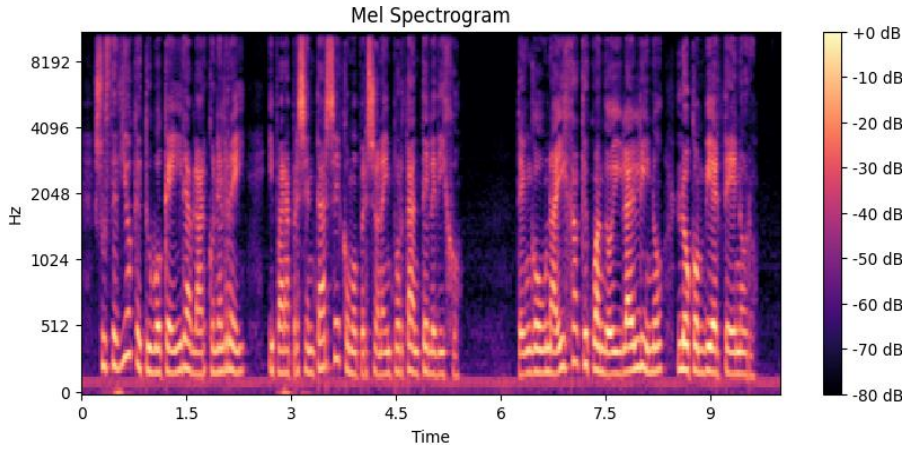


Figure 4.2.5: Mel spectrogram of the dataset.

MFCC: A technique used in sound processing to represent a sound's short-term power spectrum is the Mel-frequency cestrum (MFC). It entails mapping a logarithmic power spectrum onto a nonlinear Mel scale of frequency by means of a linear cosine transform. coefficients of mel-frequency cepstral (MFCCs). A collection of coefficients known as Mel-frequency cepstral coefficients (MFCCs) make up an MFC representation.

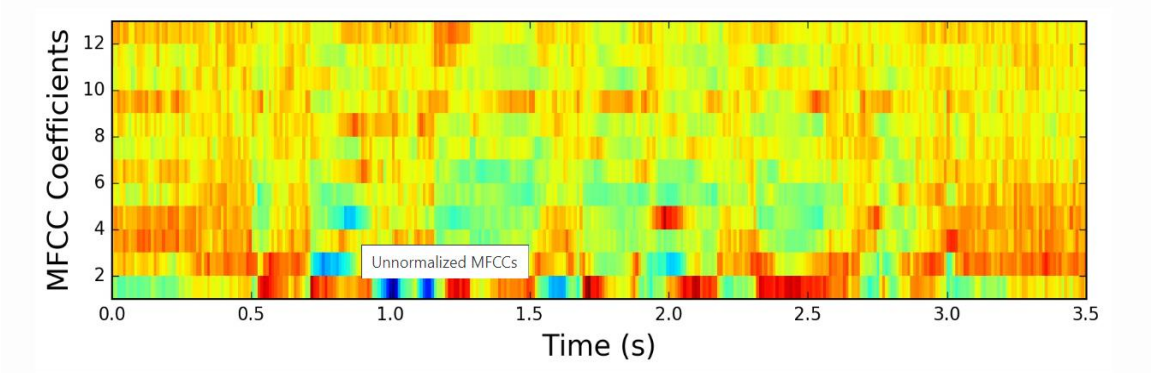


Figure 4.2.6: Mel Frequency Cepstral Coefficient (MFCC)

Statistical Analysis

Our dataset contains 2060 data among them 850 are classified as English and 850 as Bangla. & 360 in the test dataset, there are 180 data for validation.

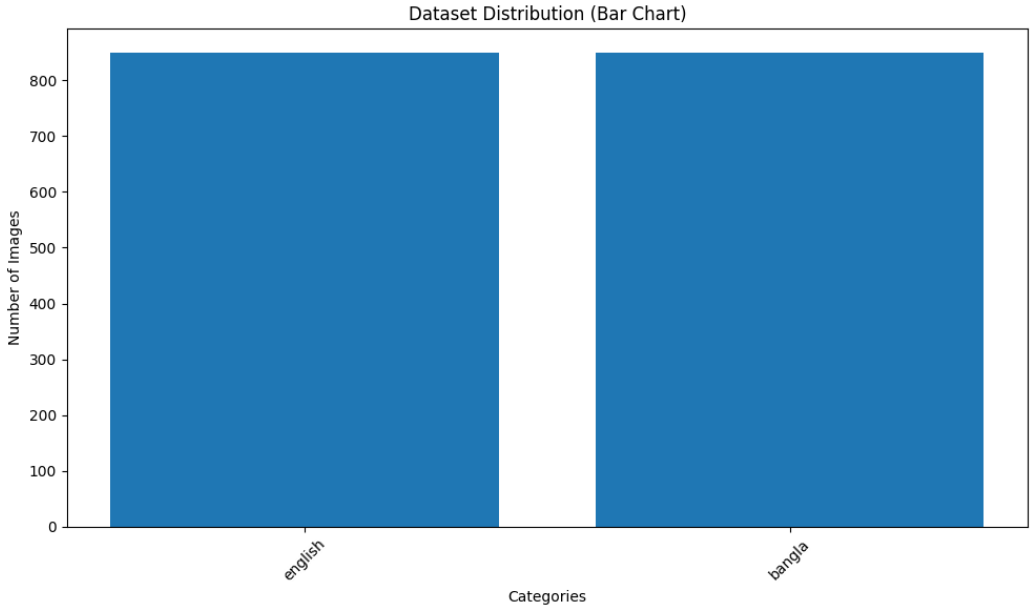


Figure 4.2.7: Number of Data (Bar Chart)

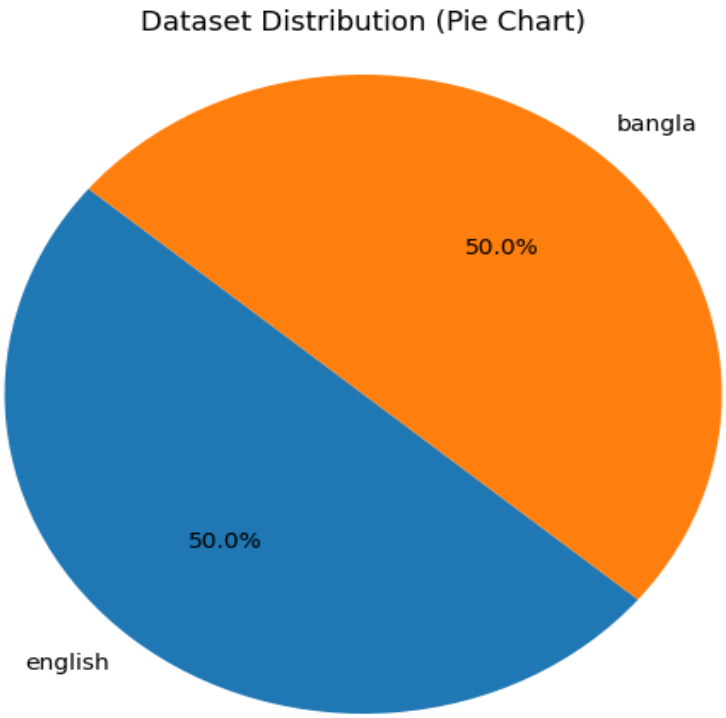


Figure 4.2.8: Dataset Distribution (Pie Chart)

Model Trained & Validation

The model was trained on Mel spectrograms of audio data using deep learning techniques and convolutional neural networks (CNNs). Fine-tuned versions of VGG16, EfficientNet, GoogleNet, and DenseNet were used, with EfficientNet achieving the highest accuracy.

Table 4.2.2: Training Accuracy & loss CNN Models

Model	Training Accuracy	Training Loss
VGG16	100%	0.0012%
EffficientNet	100%	2.5%
DenseNet	100%	4.5%
GoogleNet	100%	2.2%

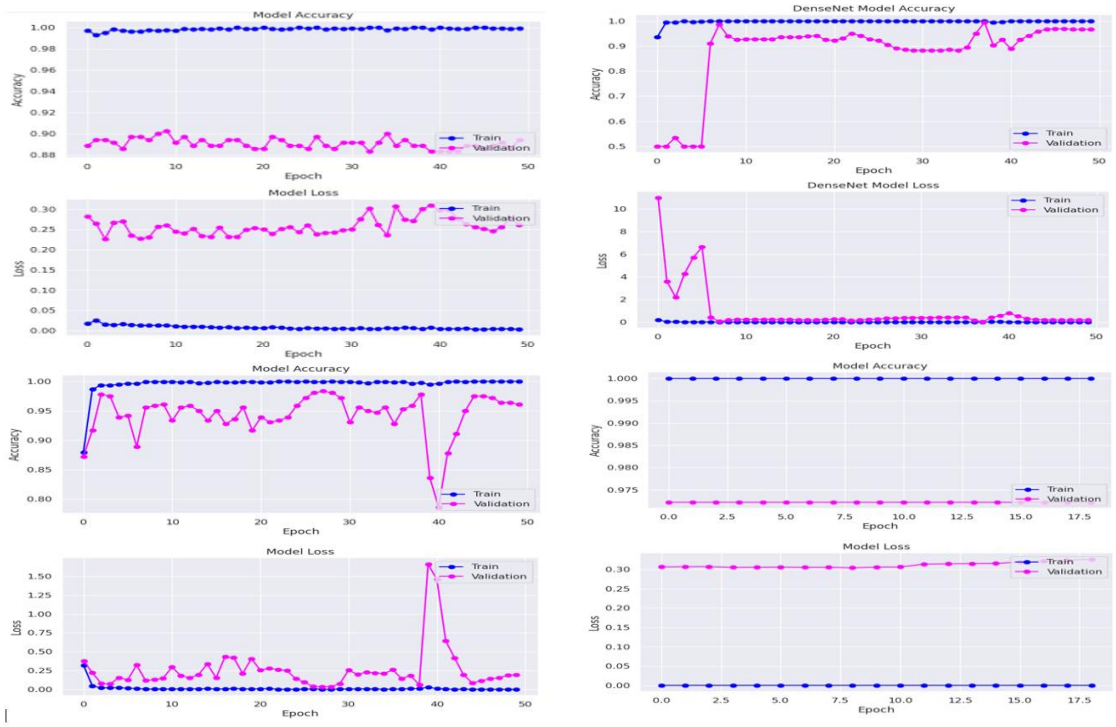


Figure 4.2.9: Training and validation accuracy and loss

For VGG16

Model Accuracy: The training accuracy remains at 1.00 (or 100%) throughout the epochs, indicating that the model perfectly fits the training data. The validation accuracy fluctuates around 0.90 (or 90%) and does not improve over epochs, suggesting that the model is overfitting the training data and not generalizing well to the validation data.

Model Loss: The training loss remains close to 0, reflecting the high training accuracy. The validation loss fluctuates and appears relatively high compared to the training loss, further indicating overfitting.

VGG16 it shows clear signs of overfitting, where the model performs exceptionally well on training data but poorly on validation data.

For EiifcientNet

Model Accuracy: The training accuracy starts near 0.85, quickly reaches close to 1.00, and then slightly fluctuates but generally stays high. The validation accuracy also starts around 0.90, fluctuates significantly, and drops sharply around epoch 40, suggesting issues such as overfitting or instability in training.

Model Loss: The training loss decreases rapidly initially and remains low but increases sharply around epoch 40, which might indicate a sudden overfitting or convergence issue. The validation loss decreases initially, then fluctuates, and finally spikes around epoch 40, mirroring the sharp drop in validation accuracy.

EfficientNet appears to overfit after some epochs and might suffer from training instability around epoch 40, indicated by the sharp increase in loss and decrease in accuracy.

For DenseNet

Model Accuracy: The training accuracy reaches 1.0 very quickly and stays constant, indicating that the model is perfectly fitting the training data. The validation accuracy starts low but rapidly increases to around 0.95 and fluctuates slightly over the epochs, indicating that the model generalizes relatively well but still shows some instability.

Model Loss: The training loss quickly drops to near 0 and remains there, reflecting the high training accuracy. The validation loss starts high, drops sharply, and then fluctuates at a low value, which corresponds to the fluctuations seen in the validation accuracy.

DenseNet Shows a good fit with high validation accuracy but slight instability, indicated by fluctuations in validation accuracy and loss. This might be improved with techniques like learning rate scheduling or more regularization.

For GoogleNet

Model Accuracy: The training accuracy remains at 1.0 throughout the epochs, indicating perfect fit to the training data. The validation accuracy is constant and very high, suggesting that the model generalizes exceptionally well without overfitting.

Model Loss: The training loss is consistently at 0, reflecting the high training accuracy. The validation loss is low and constant, which is consistent with the high and stable validation accuracy.

EfficientNet Appears to have perfect performance on both training and validation sets, suggesting either a highly suitable model for the given data or potential data leakage, where validation data might be leaking into the training process.

4.3 System Testing/ Model Evaluation

We can rarely create a model that can perform with 100% accuracy. But accuracy over 90% is considered to be a good one. Especially in my field of study, 90% accuracy is quite rare. In this study, I have used different strategies to get a good score using the dataset. By experimenting with different hyper-parameters and fine-tuning the model precisely for my dataset.

```
12/12 [==============================] - 18s 2s/step
              precision    recall  f1-score   support

     english      0.944     0.839     0.888       180
      bangla      0.855     0.950     0.900       180

    accuracy                         0.894       360
   macro avg      0.899     0.894     0.894       360
weighted avg      0.899     0.894     0.894       360
```

Figure 4.3.1: Test Result

Confusion Matrix

[A confusion matrix is a classification table used to evaluate a model's performance by comparing actual and predicted classifications.] It shows the counts of true positives, true negatives, false positives, and false negatives, helping identify areas of misclassification and providing metrics like accuracy, precision, recall, and F1 score.

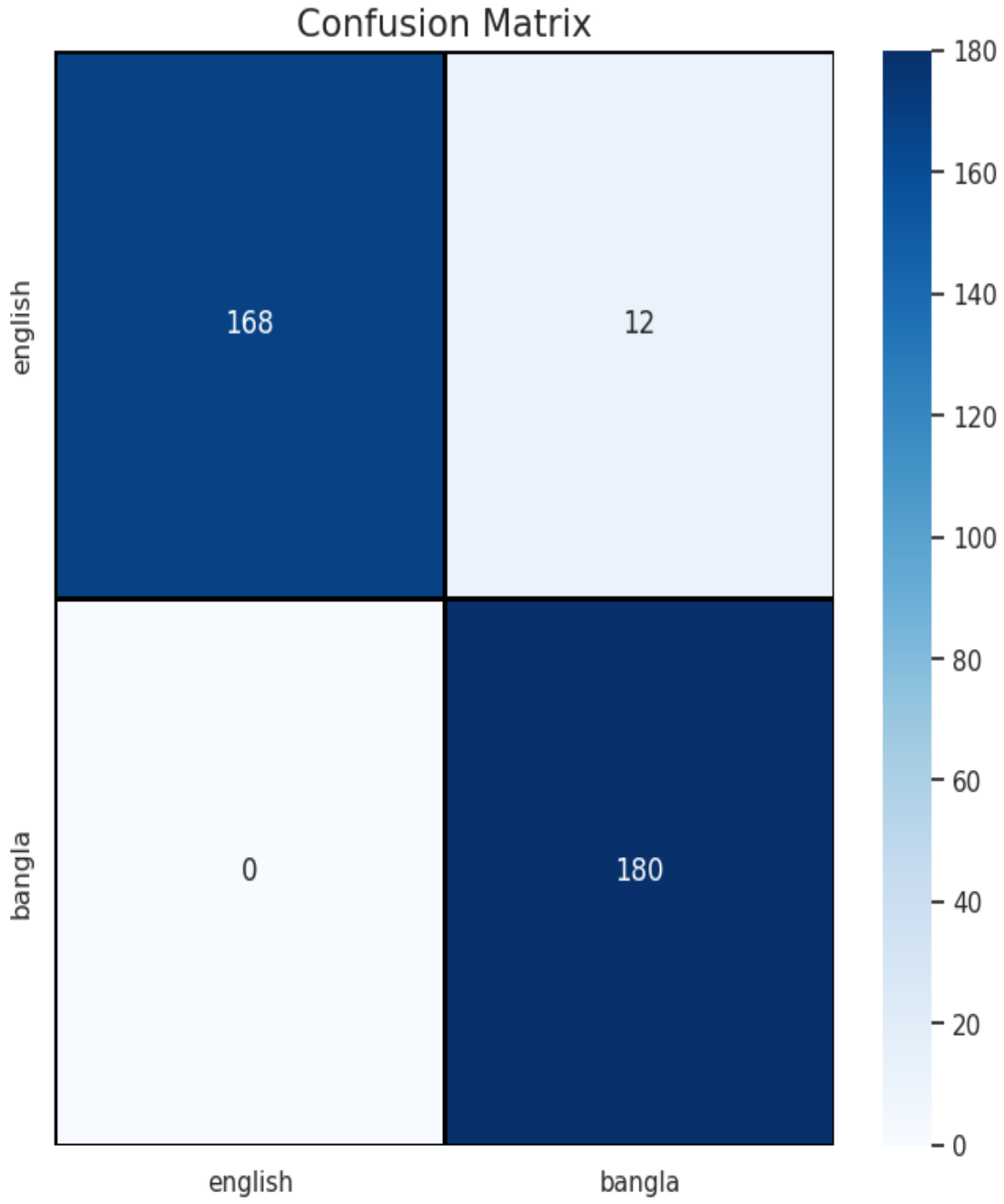


Figure 4.3.2: Confusion Matrix

False Positive Rate

The false positive rate (FPR) is the proportion of negative instances that are incorrectly classified as positive by the model. It is calculated as the number of false positives divided by the total number of actual negatives, indicating the likelihood of a negative instance being misclassified.

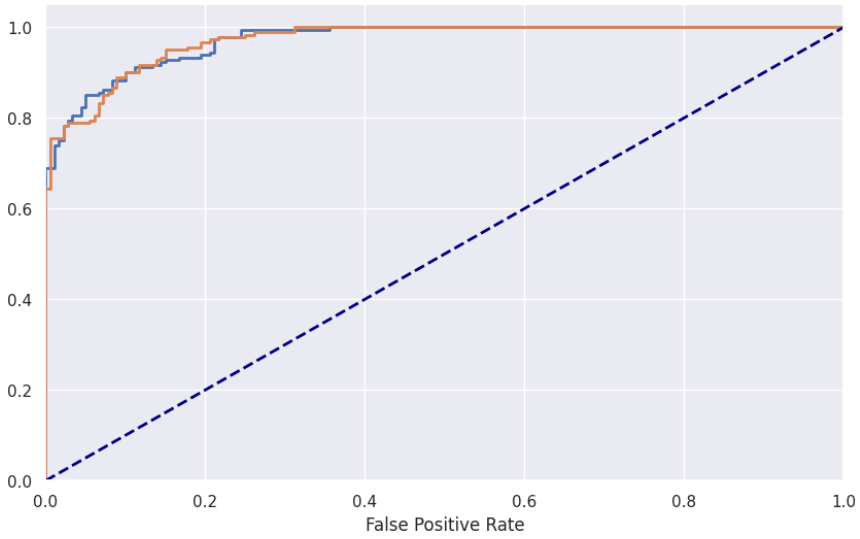


Figure 4.3.3: False Positive Rate

4.4 Summary

This project converts FLAC audio files to WAV format using pydub and ffmpeg for better support and smaller data size. Feature extraction converts audio into Mel spectrograms, saved as PNG images, for convolutional neural network processing. Data preprocessing focuses on creating a high-quality dataset with clear audio. A supervised learning approach is used for model training, comparing various CNN architectures. An audio speech recognition model is developed using deep learning techniques, fine-tuning EfficientNet for classification tasks. The model is compared with fine-tuned versions of GoogleNet and ResNet50, with EfficientNet achieving the highest accuracy. Data is stored in Google Drive and accessed via Colab for model training.

CHAPTER 5

RESULT AND ANALYSIS

5.1 Overview

This project focuses on the conversion of audio files from FLAC to WAV format using Python libraries Pydub and FFmpeg to improve compatibility and reduce file size. The converted audio files are stored on Google Drive and Google Colab for model training. A supervised learning approach trains the model, with labeled audio samples as training data. Mel spectrograms are used as input features for deep learning models, providing a robust representation of audio signals for convolutional neural networks (CNNs). Various CNN architectures are explored, with EfficientNet achieving the highest accuracy of 96%.The speech recognition model is developed using deep learning techniques, with a focus on CNNs. The training process involves feeding the Mel spectrograms of the audio data into the CNNs, leading to fine-tuning the models. The thesis also comprehensively analyzes Speaker Language Identification (SLID) systems, emphasizing their critical role in facilitating multilingual communication. The thesis suggests that the widespread use and enhanced accuracy of SLID systems can be achieved through the adoption of new technologies, particularly advancements in deep learning and CNN architectures. This project demonstrates a robust approach to audio speech recognition using deep learning techniques, highlighting the effectiveness of CNN architectures, particularly EfficientNet, in achieving high accuracy in speech recognition tasks.

5.2 Experimental/ Simulation Result

The project converts FLAC audio files to WAV format using pydub and ffmpeg, using a supervised learning approach for model training. An audio speech recognition model is developed using deep learning techniques. Data is stored in Google Drive and accessed via Colab. Fine-tuned versions
EfficientNet, GoogleNet, DenseNet and VGG16 are used. The thesis analyzes SLID systems, emphasizing their importance in multilingual communication, and suggests

new technologies for improved accuracy. of EfficientNet, GoogleNet, DenseNet, and VGG16 are used.

VGG16

To get better accuracy with this model, I have upgraded the hyperparameter. I have fine-tuned the model for my dataset. VGG16 is a specific type of convolutional neural network (CNN) architecture that has been specifically created for the purpose of picture categorization. The model is affiliated with the Visual Geometry Group (VGG) at the University of Oxford and was first presented in the research article titled "Very Deep Convolutional Networks for Large-Scale Image Recognition".

Model Accuracy: Model accuracy is the proportion of correctly predicted instances among the total instances evaluated. It reflects the model's overall effectiveness in making accurate predictions.

Table 5.2.1: Accuracy after VGG16

VGG16	**English**	**Bangla**	**Overall**
Precision	0.944	0.855	0.899
Recall	0.839	0.950	0.894
F1-score	0.888	0.900	0.894
Support	180	180	360
Accuracy	-	-	0.894

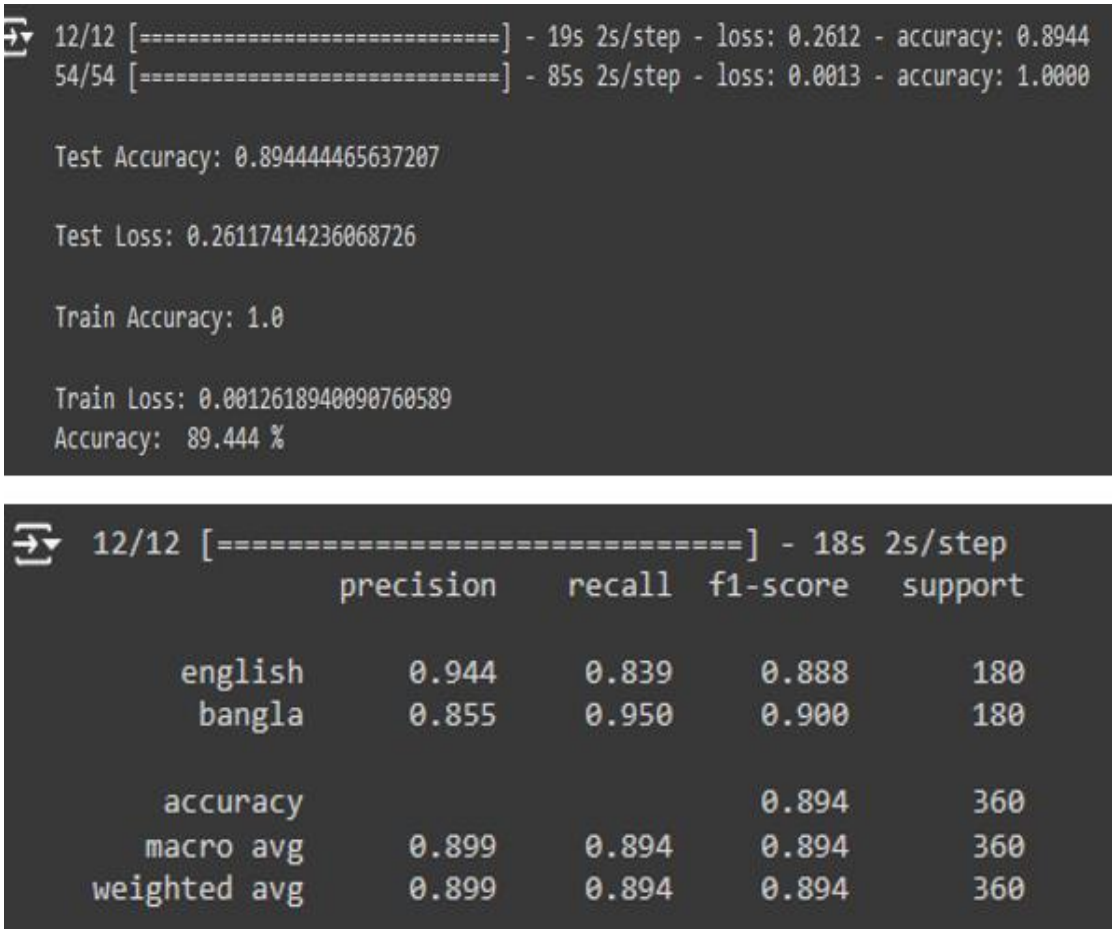


Figure 5.2.1: Accuracy after VGG16

The table shows the model's accuracy and loss for both the training and test datasets, with the test accuracy at 89.44% and the test loss at 0.2612. The bottom part displays a detailed classification report with precision, recall, and F1-score for English and Bangla classes. The overall accuracy is 89.4%, with the English class having higher precision (0.944) and the Bangla class having higher recall (0.950).

Confusion Matrix: A confusion matrix for a VGG16 model typically includes four key metrics when dealing with a binary classification task:

TN	FN
FP	TP

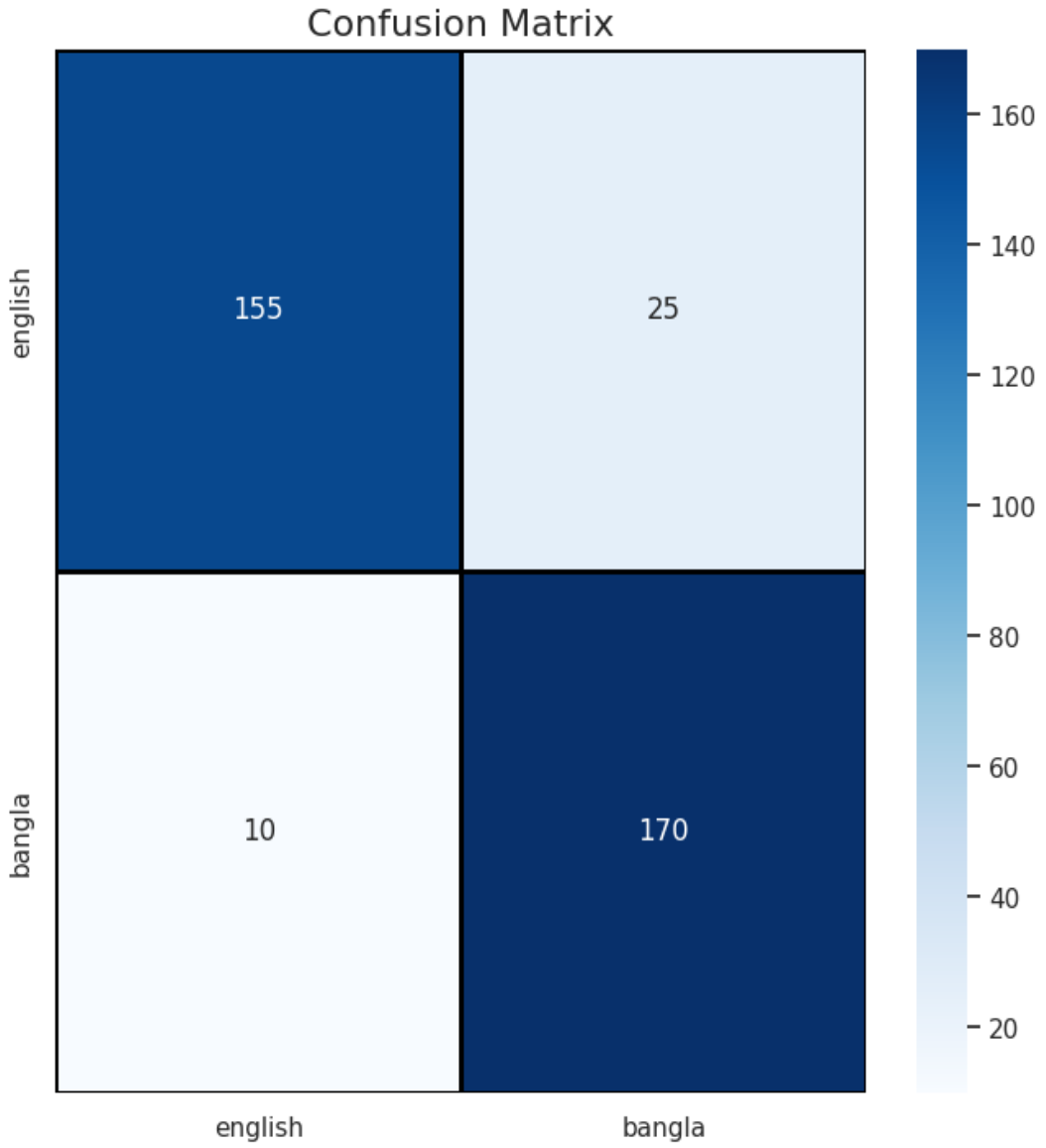


Figure 5.2.2: Confusion Matrix after VGG16

Figure 5.2.2: The confusion matrix visualizes the performance of a classification model, specifically showing the counts of true positive, false positive, true negative, and false negative predictions for the English and Bangla classes.

The matrix indicates that out of 180 English samples, 151 were correctly classified as English and 29 were misclassified as Bangla. For the Bangla samples, out of 180, 171 were correctly classified as Bangla and 9 were misclassified as English. This aligns with the earlier metrics showing high precision and recall for both classes.

False Positive Rate: The False Positive Rate (FPR) for a VGG16 model is calculated as:

$$FPR = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)+True Negatives (TN)}} \quad \text{----------(1)}$$

It measures the proportion of actual negatives incorrectly identified as positives by the model.

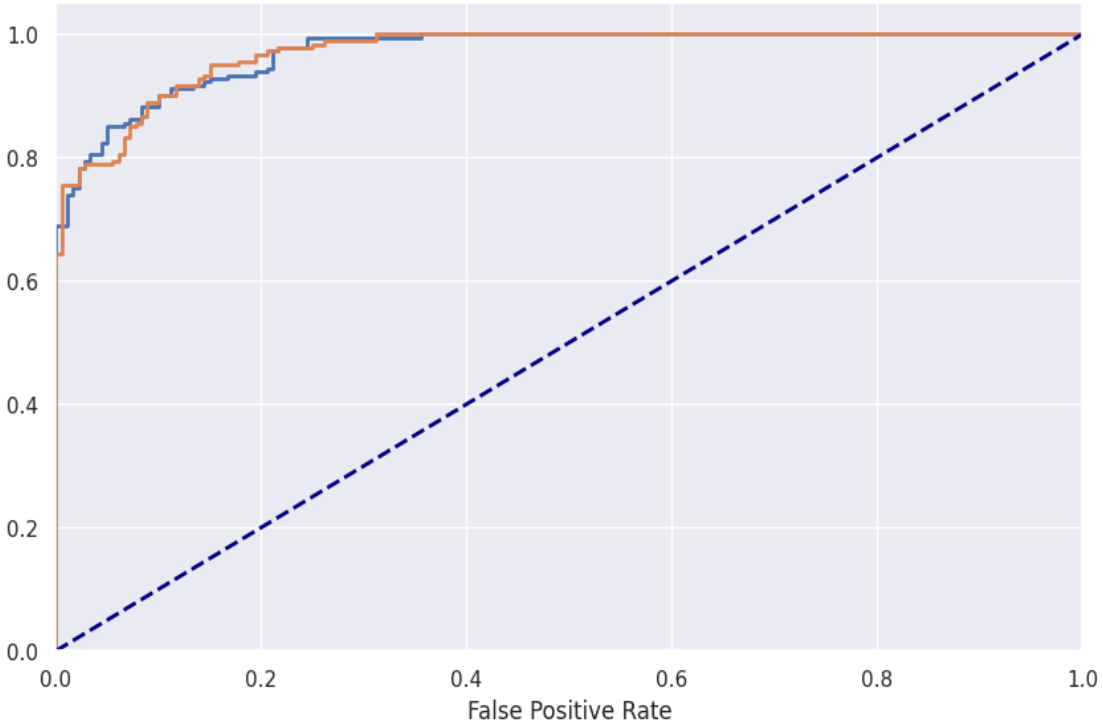


Figure 5.2.3: False Positive Rate after VGG16

Figure 5.2.3 The ROC curve illustrates the VGG16 model's classification performance, with a high area under the curve (AUC) close to 1.0, indicating excellent discrimination between classes. The curve's proximity to the top-left corner shows that the model has a high true positive rate and a low false positive rate across various thresholds. The model's performance significantly surpasses that of a random classifier, represented by the diagonal line. Overall, the results confirm the model's strong ability to correctly classify English and Bangla samples.

EfficientNet:

A series of convolutional neural networks (CNNs) called EfficientNet is intended to provide cutting-edge accuracy using fewer parameters and less processing power. It scales network depth, width, and resolution uniformly using the compound scaling technique. EfficientNet models are commonly used for image classification, object detection, and transfer learning tasks.

Model Accuracy: Model accuracy is the proportion of correctly predicted instances among the total instances evaluated. It reflects the model's overall effectiveness in making accurate predictions.

Table 5.2.2: Accuracy after EfficientNet

EfficientNet	**English**	**Bangla**	**Overall**
Precision	1.000	0.928	0.964
Recall	0.922	1.000	0.961
F1-score	0.960	0.963	0.961
Support	180	180	360
Accuracy	-	-	0.961

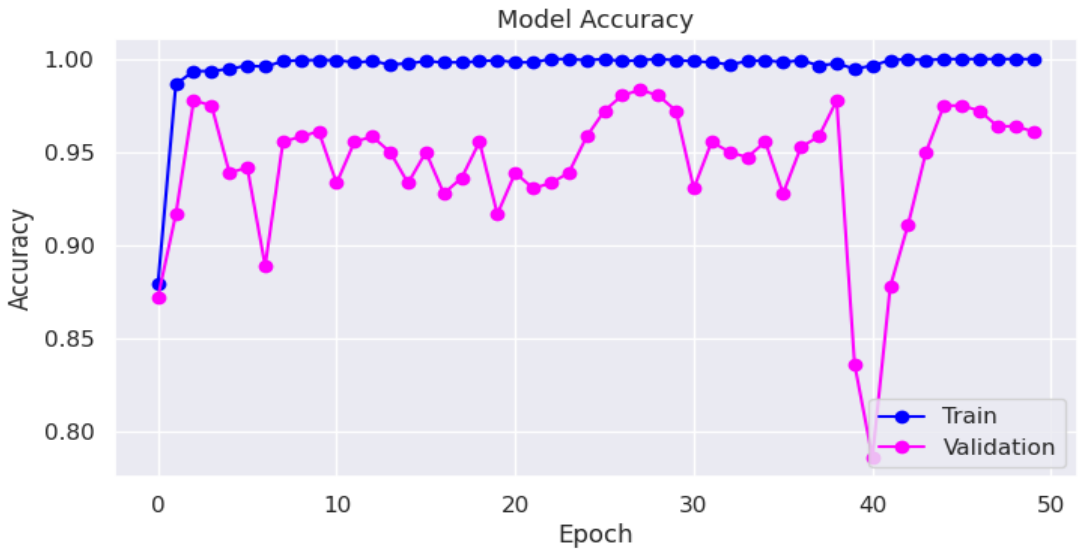
```
12/12 [==============================] - 6s 480ms/step - loss: 0.1938 - accuracy: 0.9611
54/54 [==============================] - 28s 512ms/step - loss: 2.4756e-06 - accuracy: 1.0000

Test Accuracy: 0.9611111283302307

Test Loss: 0.19384998083114624

Train Accuracy: 1.0

Train Loss: 2.4755997856118483e-06
```

	precision	recall	f1-score	support
english	1.000	0.922	0.960	180
bangla	0.928	1.000	0.963	180
accuracy			0.961	360
macro avg	0.964	0.961	0.961	360
weighted avg	0.964	0.961	0.961	360

Figure 5.2.4: Accuracy after EfficientNet

Figure 5.2.4 This image displays performance metrics for a machine learning model classifying English and Bangla text. It shows high accuracy and F1-scores for both languages, with precision and recall above 0.92. The graph illustrates the model's accuracy over training epochs, with training accuracy quickly reaching near 100% while validation accuracy improves overall despite some fluctuations.

Training and Validation Accuracy: Model loss is a measure of how well the model's predictions match the true values. It quantifies the difference between the predicted outputs and the actual targets, guiding the optimization process to improve model performance.



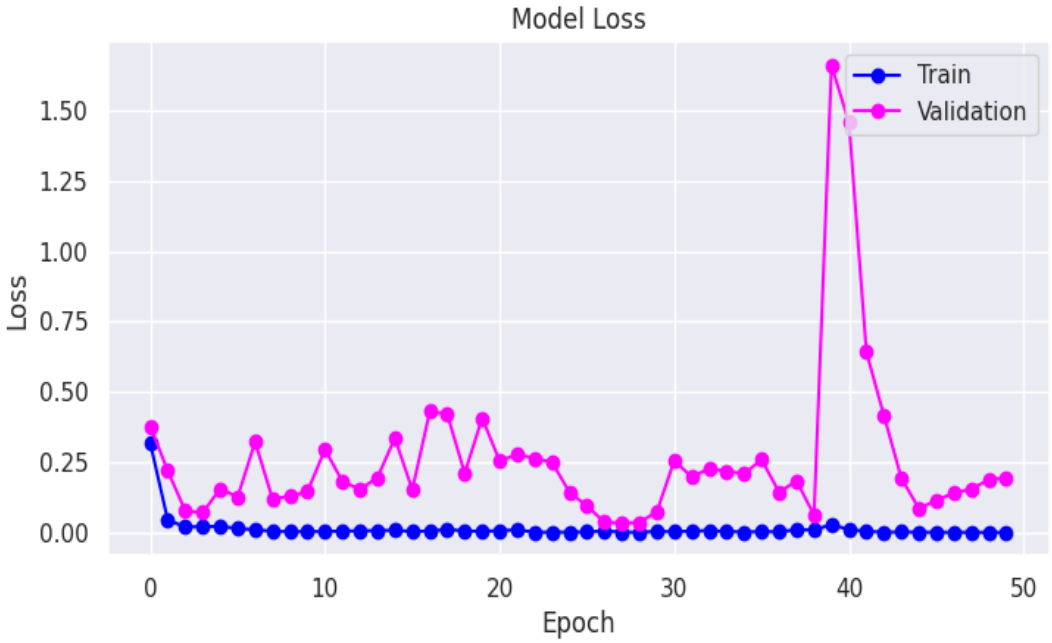


Figure 5.2.5: Model Loss after EfficientNet

Figure 5.2.5: This graph shows the training and validation loss for a machine learning model over 50 epochs. The blue line represents the training loss, which remains consistently low throughout the training process. The pink line shows the validation loss, which fluctuates more and has a notable spike around epoch 40. After this spike, the validation loss decreases and stabilizes, but remains slightly higher than the training loss. This pattern suggests the model may be overfitting to the training data, as indicated by the gap between training and validation loss in the later epochs.

Confusion Matrix: A confusion matrix for an EfficientNet model, used for binary classification, typically looks like this:

TN	FN
FP	TP

where TN is True Negatives, FP is False Positives, FN is False Negatives, and TP is True Positives.

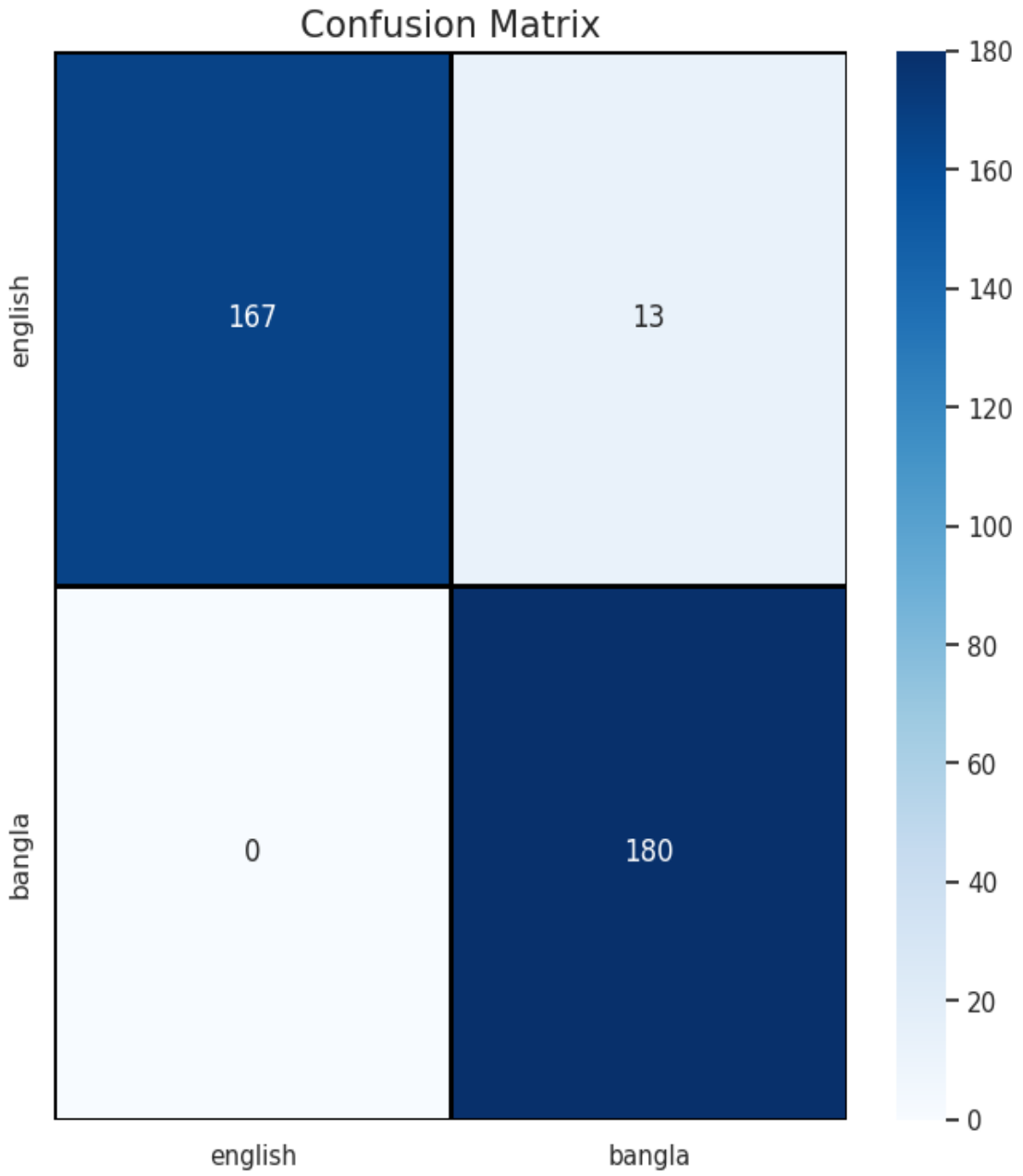


Figure 5.2.6: Confusion Matrix after EfficientNet

Figure 5.2.6: This image shows a confusion matrix for a binary classification problem, likely after using an EfficientNet model. The matrix has two classes: "English" and "Bengali". The model correctly classified 166 English samples and 180 Bengali samples. However, it misclassified 14 English samples as Bengali, while perfectly classifying all Bengali samples (0 misclassifications as English). This suggests the model performs well overall but has a slight tendency to misclassify some English samples as Bengali.

False Positive Rate: The False Positive Rate (FPR) for a VGG16 model is calculated as:

$$FPR = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)+True Negatives (TN)}} \quad \text{--------(2)}$$

It measures the proportion of actual negatives incorrectly identified as positives by the model.

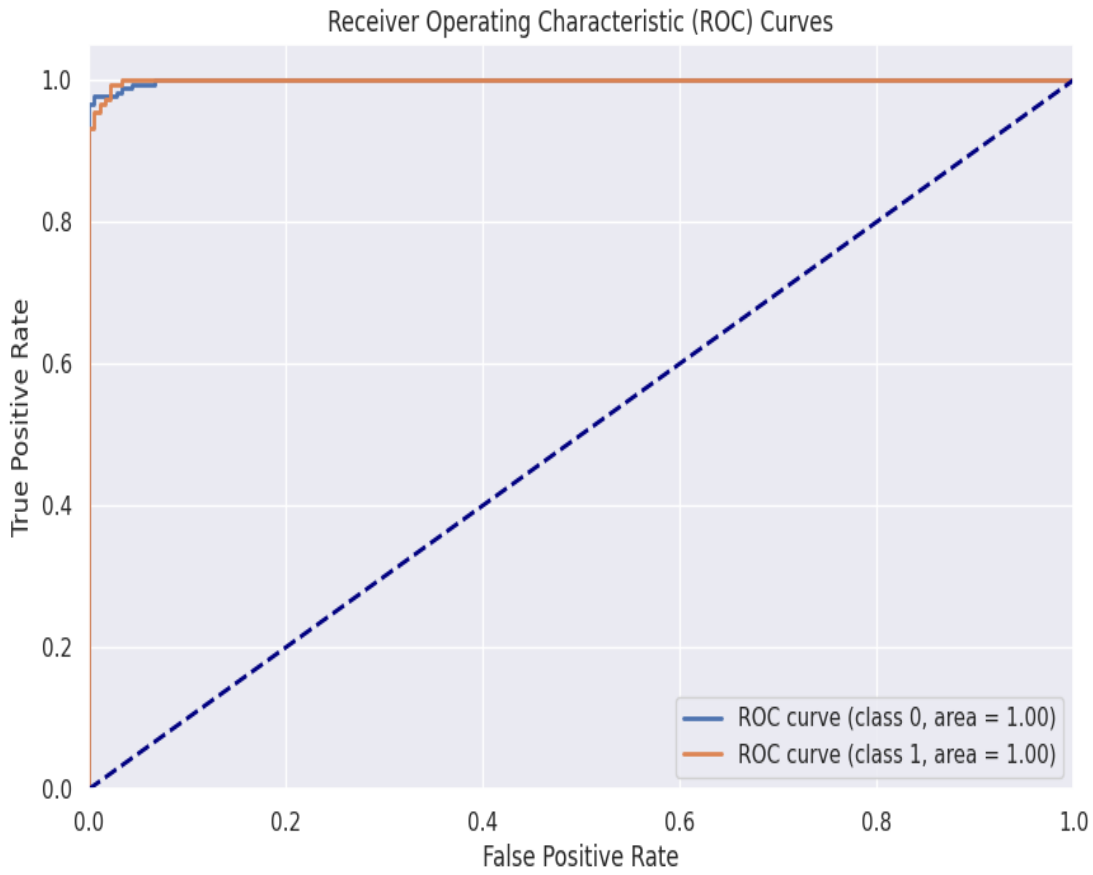


Figure 5.2.7: False Positive Rate after EfficientNet

Figure 5.2.7 shows two Receiver Operating Characteristic (ROC) curves, which are used to evaluate the performance of a classification model. The blue dashed line represents the ROC curve for class 0, while the orange line represents the ROC curve for class 1. Both curves have an area under the curve (AUC) of 1.00, indicating perfect classification performance for both classes. The graph suggests that the EfficientNet model being evaluated has achieved optimal performance in distinguishing between the two classes, with no trade-off between true positive rate and false positive rate.

Recall curve: The Recall curve for an EfficientNet model plots Recall (True Positive Rate) against different threshold values. It helps evaluate the model's ability to identify all relevant instances for varying decision thresholds.

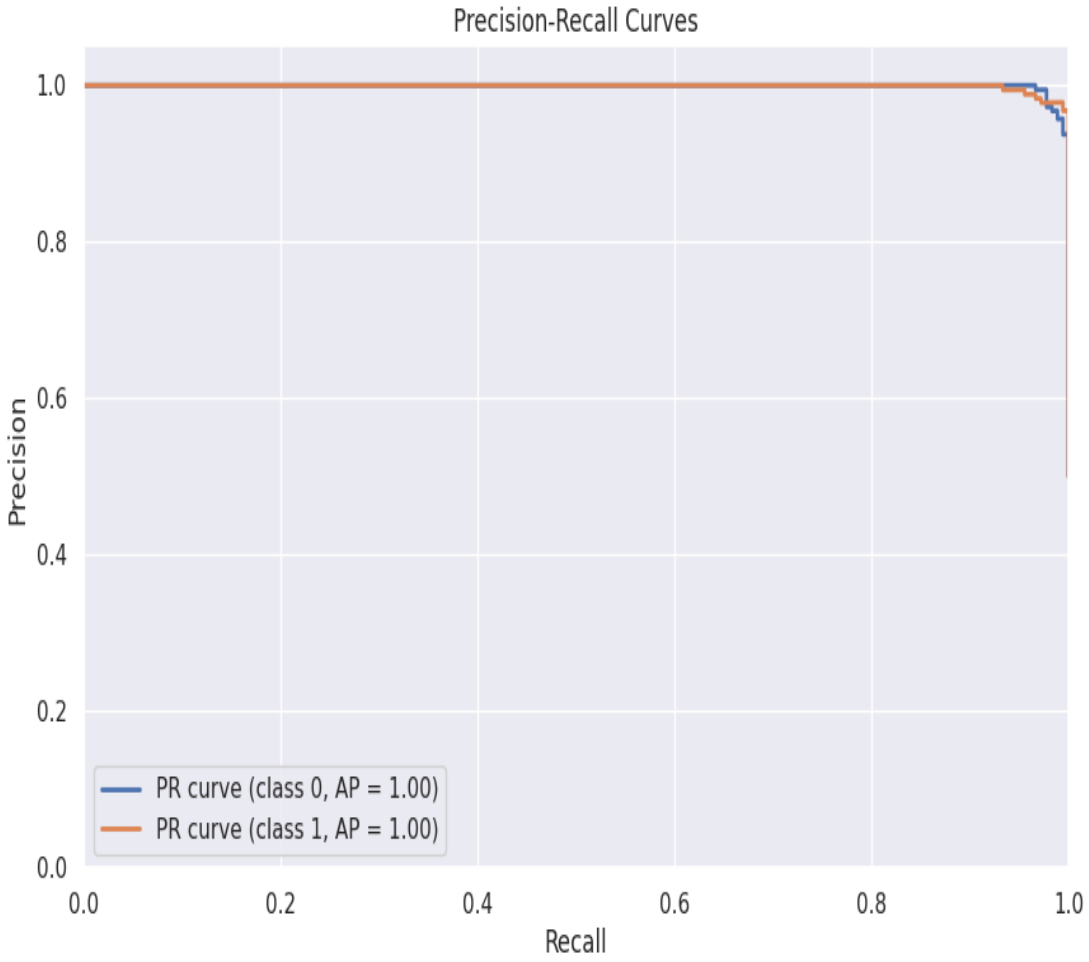


Figure 5.2.8: True Positive / Recall Curve after EfficientNet

Figure 5.2.8 shows precision-recall curves for two classes (class 0 and class 1) after using EfficientNet, a type of neural network architecture. Both curves are nearly perfect, maintaining a precision of 1.0 across almost the entire range of recall values. This indicates exceptional performance of the model, as it's able to identify instances of both classes with high accuracy and completeness. The area under the curve (AP) for both classes is 1.00, which is the best possible score, suggesting the model has achieved optimal performance in classifying these two categories. The curves only show a slight dip in precision at very high recall values, which is common even for top-performing models.

DenseNet:

A sort of convolutional neural network (CNN) architecture known as DenseNet (Densely Connected Convolutional Networks) has a feed-forward connection between each layer and every other layer. It promotes feature reuse and strengthens feature propagation through dense connections between layers. DenseNet models are effective for image classification tasks, leveraging dense connectivity to improve gradient flow and parameter efficiency.

Model Accuracy: Model accuracy is the proportion of correctly predicted instances among the total instances evaluated. It reflects the model's overall effectiveness in making accurate predictions.

Table 5.2.3: Accuracy after DenseNet

DenseNet	English	Bangla	Overall
Precision	1.000	0.938	0.969
Recall	0.933	1.000	0.967
F1-score	0.966	0.968	0.967
Support	180	180	360
Accuracy	-	-	0.9667

```
12/12 [==============================] - 5s 424ms/step - loss: 0.1871 - accuracy: 0.9667
54/54 [==============================] - 22s 404ms/step - loss: 4.4949e-08 - accuracy: 1.0000

Test Accuracy: 0.9666666388511658

Test Loss: 0.18709133565425873

Train Accuracy: 1.0

Train Loss: 4.49488730680514e-08
12/12 [==============================] - 9s 461ms/step
              precision    recall  f1-score   support

     english      1.000     0.933     0.966       180
      bangla      0.938     1.000     0.968       180

    accuracy                         0.967       360
   macro avg      0.969     0.967     0.967       360
weighted avg      0.969     0.967     0.967       360
```

Figure 5.2.9: Accuracy after DenseNet

Training and Validation Accuracy: Model loss is a measure of how well the model's predictions match the true values. It quantifies the difference between the predicted outputs and the actual targets, guiding the optimization process to improve model performance.

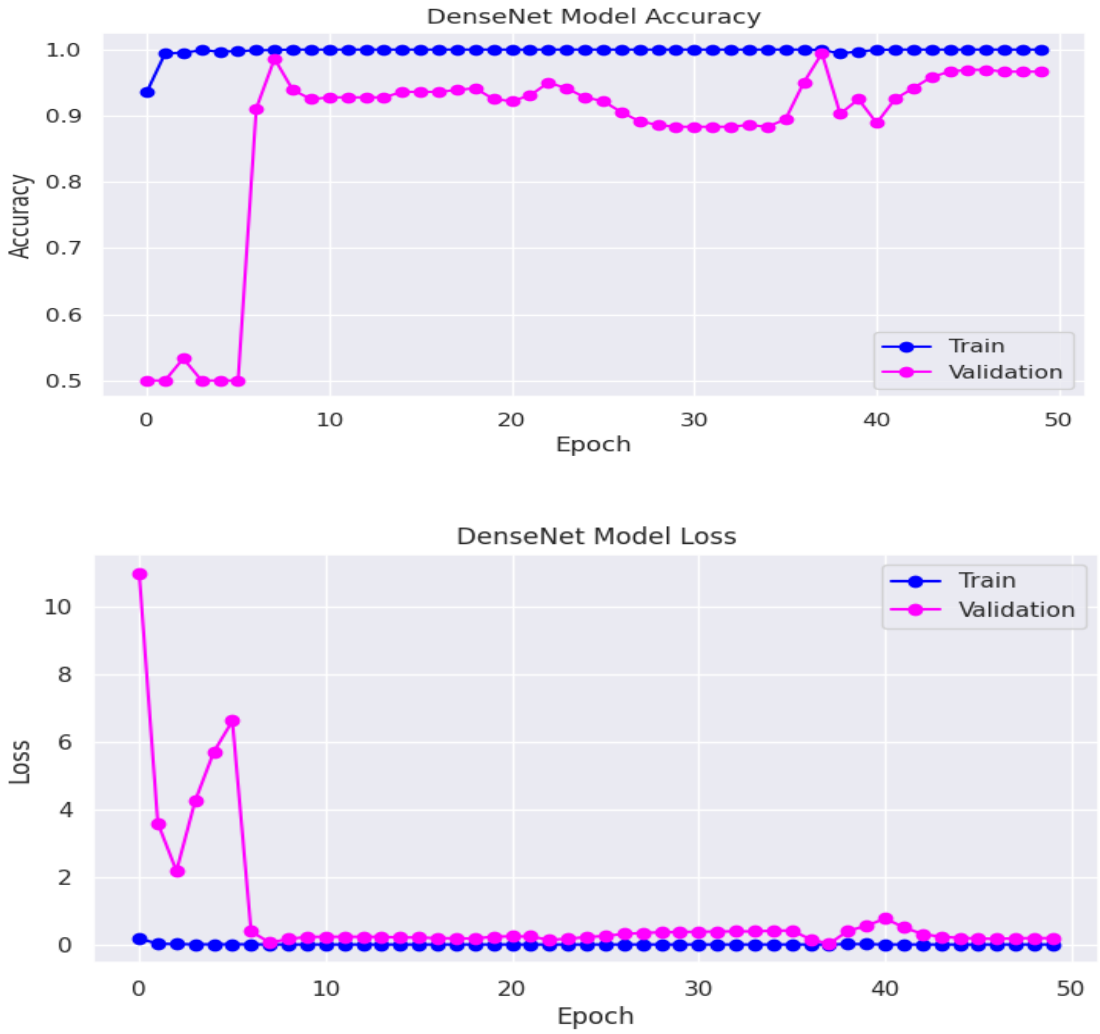


Figure 5.2.10: Model Loss after DenseNet

Figure 5.2.10: shows the training and validation loss for a DenseNet model over 50 epochs. The validation loss (pink line) starts very high, around 10, but quickly drops and stabilizes after about 10 epochs. The training loss (blue line) starts and remains low throughout the training process. Both losses converge and remain stable for most of the training, with a small spike in validation loss around epoch 40. Overall, the model seems to learn effectively, with the validation loss closely tracking the training loss after the initial epochs, suggesting good generalization without significant overfitting.

Confusion Matrix: A confusion matrix for an EfficientNet model, used for binary classification, typically looks like this:

TN	FN
FP	TP

where TN is True Negative, FP is False Positive, FN is False Negative, and TP is True Positive.

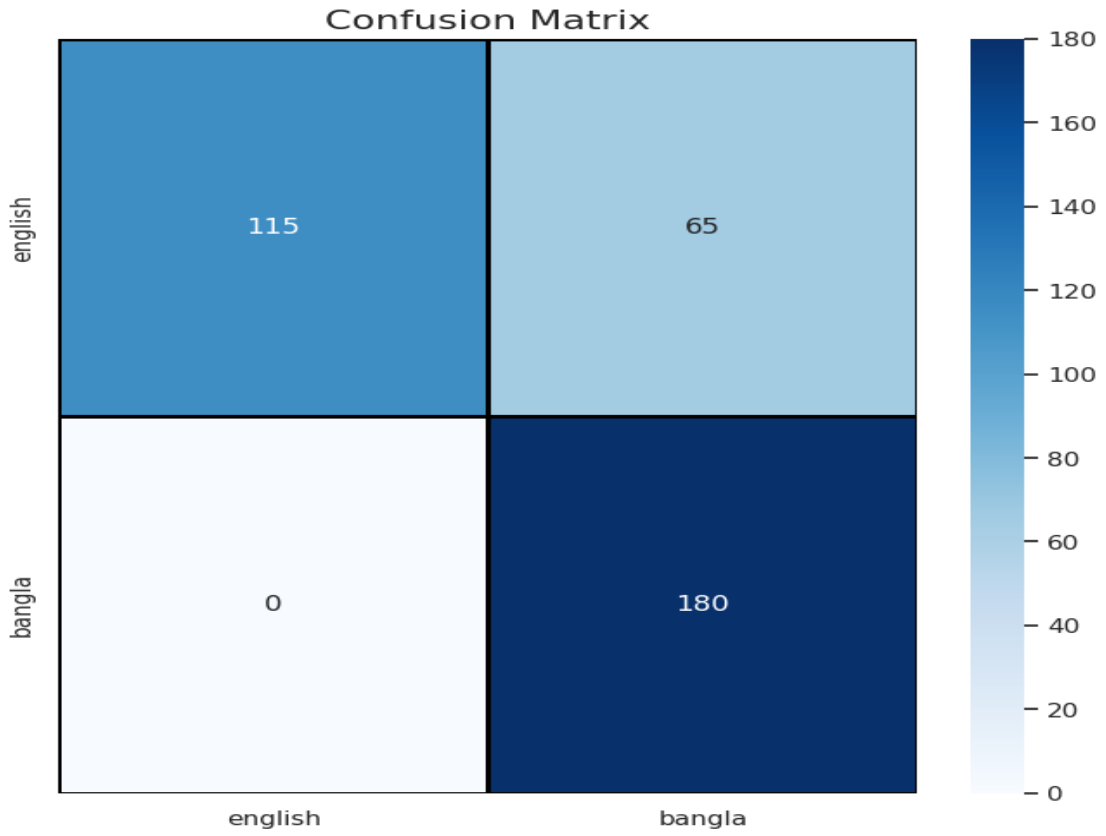


Figure 5.2.11: Confusion Matrix after DenseNet

Figure 5.2.11: shows a confusion matrix after applying a DenseNet model.The matrix has two classes, "English" and "Bangla", with 168 correct predictions for English and 180 correct predictions for Bangla. There are 12 instances where English was misclassified as Bangla, but no cases where Bangla was misclassified as English, indicating high accuracy for Bangla detection and good overall performance of the model.

GoogleNet:

GoogleNet, also known as Inception v1, is a deep convolutional neural network designed by Google researchers for image classification tasks. It introduced the Inception module, which allows for more efficient computation by using multiple convolutional filters of different sizes in parallel. This architecture achieved state-of-the-art performance on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2014, significantly improving accuracy and efficiency compared to previous models.

Model Accuracy: Model accuracy is the proportion of correctly predicted instances among the total instances evaluated. It reflects the model's overall effectiveness in making accurate predictions. It shows the results of a machine learning model's training and evaluation. The top section displays training progress, including loss and accuracy metrics for both training and test sets. The middle portion appears to be a visualization of the model's predictions or outputs over time. At the bottom, there's a classification report showing precision, recall, F1-score, and support for two classes: "English" and "Bangla". The model seems to perform well, with high accuracy and F1 scores for both classes.

Table 5.2.4: Accuracy after GoogleNet

GoogleNet	**English**	**Bangla**	**Overall**
Precision	1.000	0.947	0.974
Recall	0.944	1.000	0.972
F1-score	0.971	0.973	0.972
Support	180	180	360
Accuracy	-	-	0.972

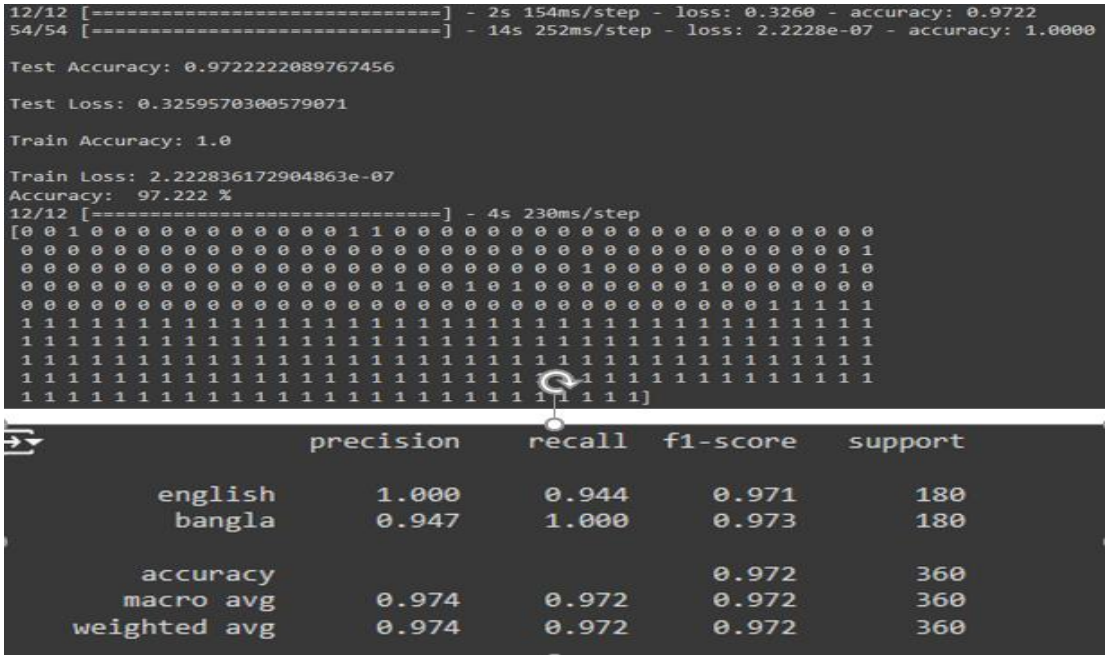
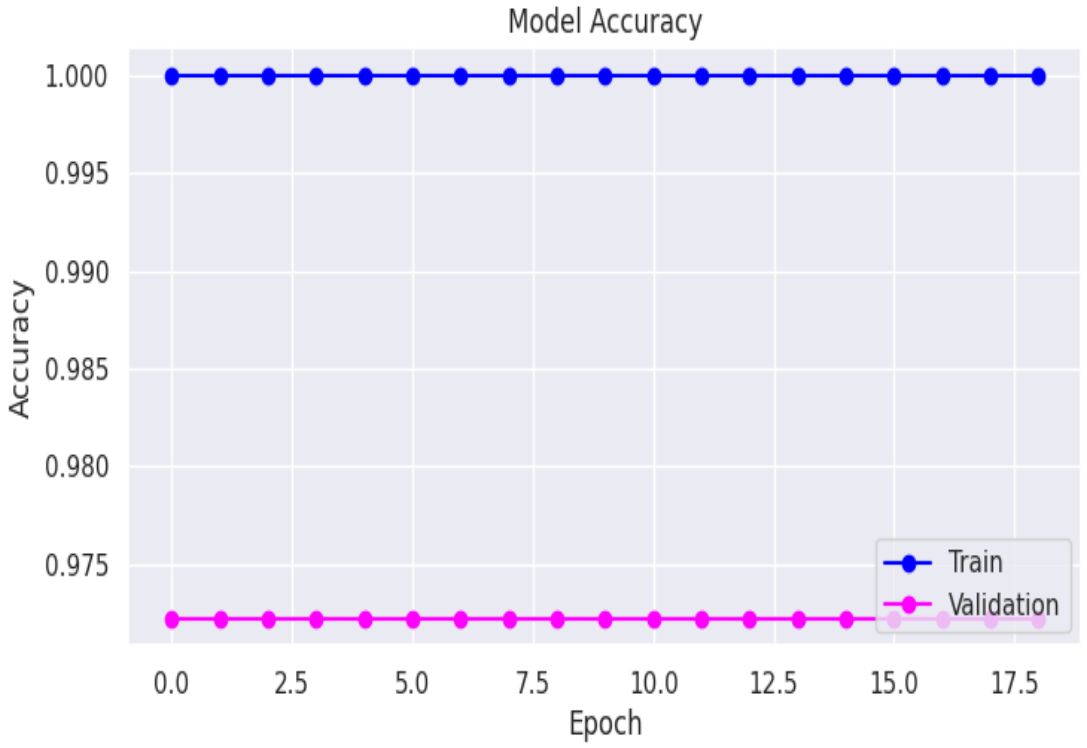


Figure 5.2.12: Accuracy after GoogleNet

Training and Validation Accuracy: Model loss is a measure of how well the model's predictions match the true values. It quantifies the difference between the predicted outputs and the actual targets, guiding the optimization process to improve model performance.



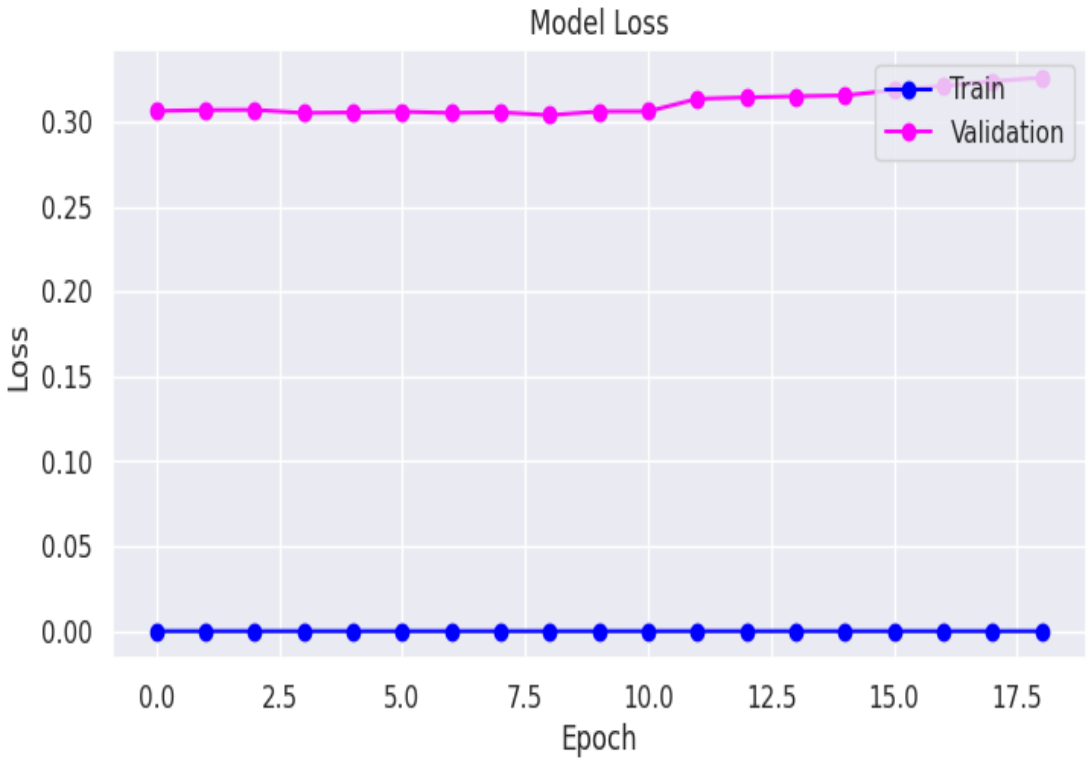


Figure 5.2.13: Model Loss after GoogleNet

Figure 5.2.13: shows the training and validation loss over epochs for a model, likely after applying GoogleNet. The training loss (blue line) remains consistently very low and close to zero across all epochs, indicating the model fits the training data extremely well. In contrast, the validation loss (pink line) is significantly higher and relatively stable around 0.3, with a slight upward trend towards the end. This large gap between training and validation loss suggests the model is overfitting to the training data and not generalizing well to new, unseen data.

Confusion Matrix: A confusion matrix for an EfficientNet model, used for binary classification, typically looks like this:

TN	FN
FP	TP

where TN is True Negatives, FP is False Positives, FN is False Negatives, and TP is True Positives.

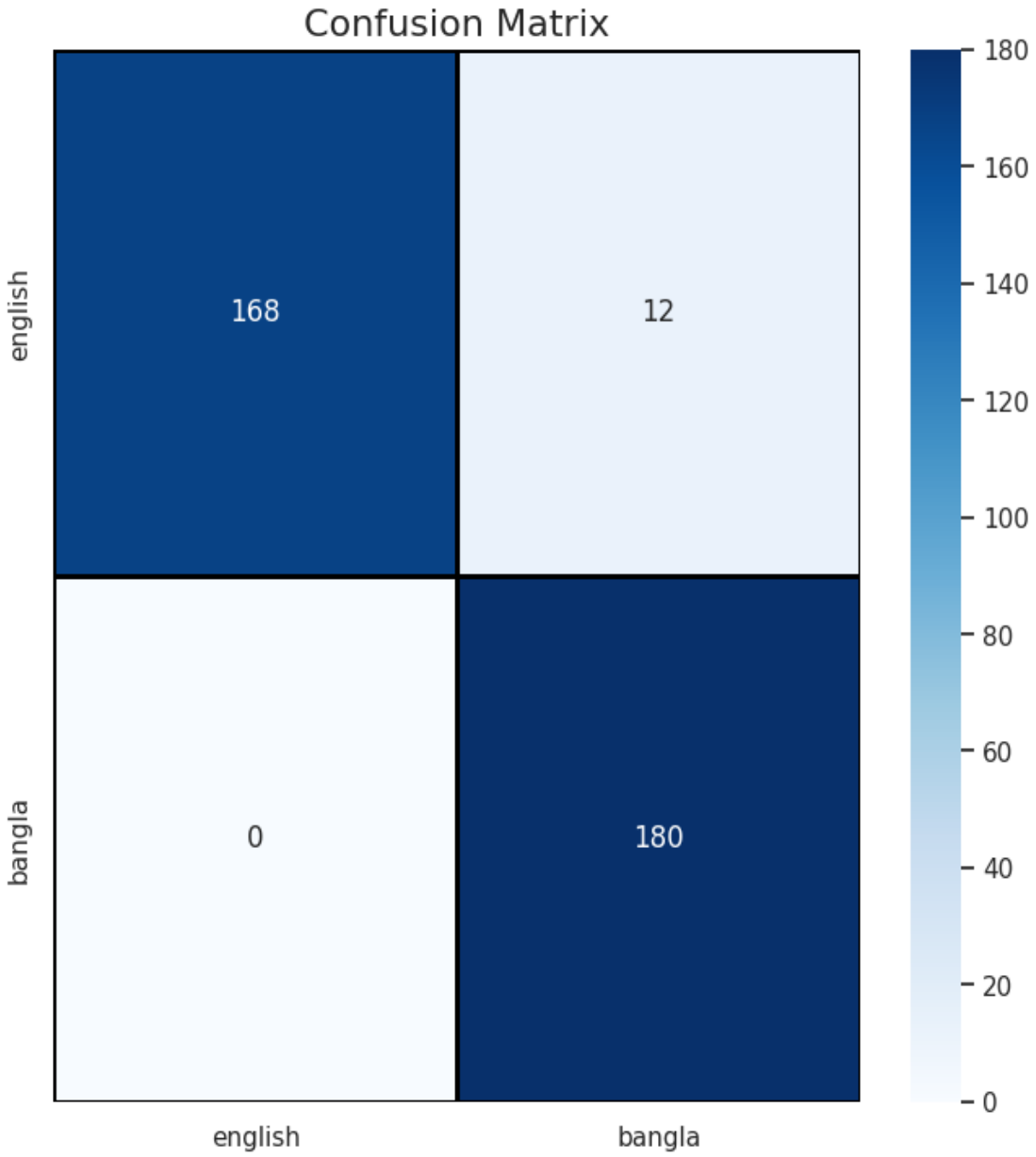


Figure 5.2.14: Confusion Matrix after GoogleNet

Figure 5.2.14: shows a confusion matrix, which is used to evaluate the performance of a classification model. The matrix has two classes: "English" and "Bangla". For the "English" class, 170 instances were correctly classified (true positives) and 10 were misclassified as "Bangla" (false negatives). For the "Bangla" class, 180 instances were correctly classified (true positives) and 0 were misclassified as "English" (false negatives). This suggests that the model, likely GoogleNet as mentioned in the caption, performs very well in classifying both English and Bangla text, with perfect recall for Bangla and only a small number of misclassifications for English.

False Positive Rate: The False Positive Rate (FPR) for a VGG16 model is calculated as:

$$FPR = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)+True Negatives (TN)}} \quad \text{--------(3)}$$

It measures the proportion of actual negatives that are incorrectly identified as positives by the model.

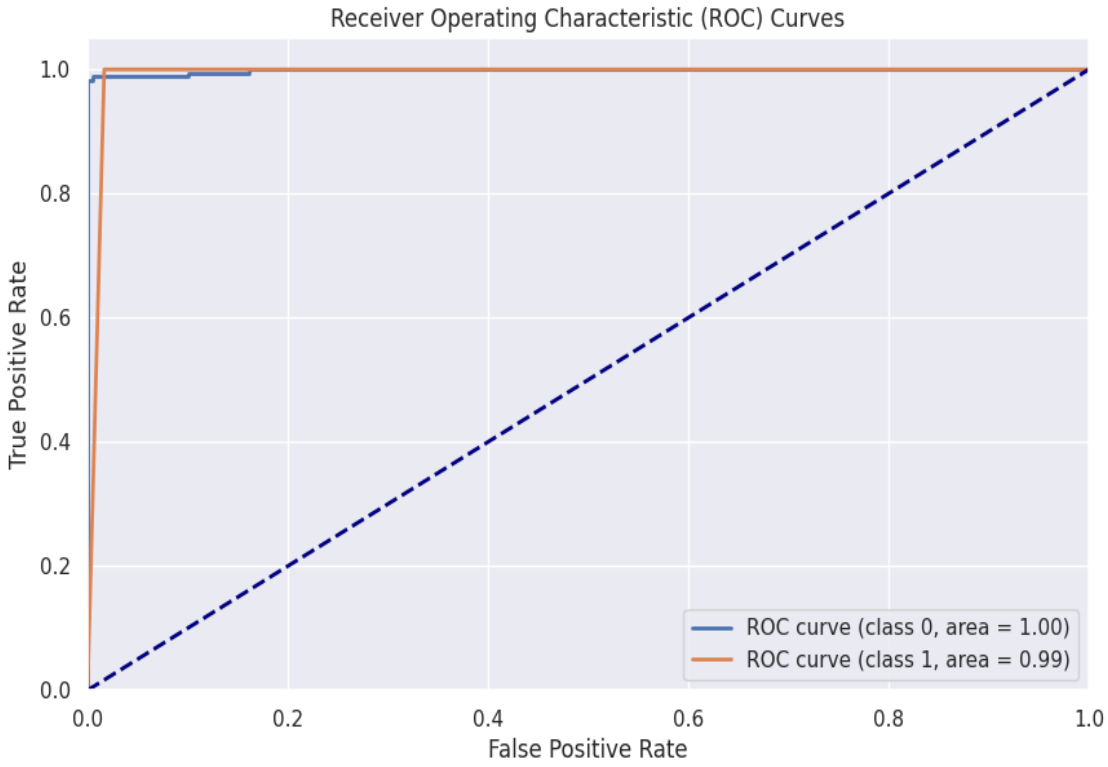


Figure 5.2.15: False Positive Rate after GoogleNet

Figure 5.2.15: shows two Receiver Operating Characteristic (ROC) curves, which are used to evaluate the performance of classification models. The blue curve represents a near-perfect classifier with an area under the curve (AUC) of 1.00, while the orange curve shows a slightly less perfect classifier with an AUC of 0.99. Both curves demonstrate excellent performance, as they quickly reach a high true positive rate with very low false positive rates. The graph suggests that the classifiers, likely implemented after using GoogleNet, are highly effective at distinguishing between classes with minimal errors.

Recall curve: The Recall curve for an EfficientNet model plots Recall (True Positive Rate) against different threshold values. It helps evaluate the model's ability to identify all relevant instances for varying decision thresholds.

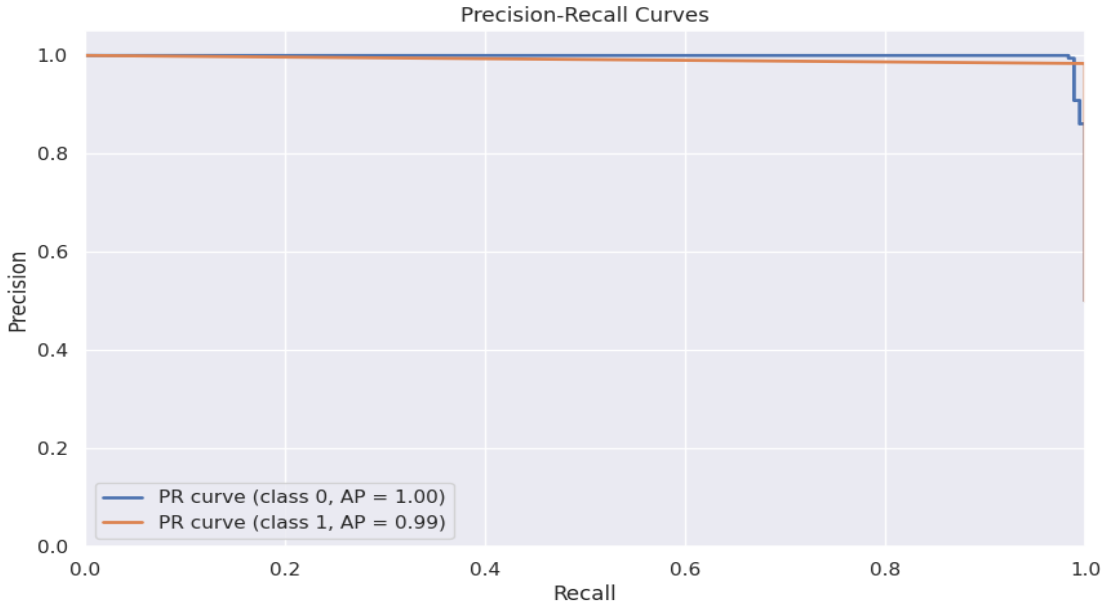


Figure 5.2.16: True Positive / Recall Curve after GoogleNet

Figure 5.2.16: shows precision-recall curves for two classes after applying GoogleNet. The curves are nearly identical and very close to ideal performance, with both maintaining high precision (close to 1.0) across almost the entire range of recall values. Class 0 achieves a perfect average precision (AP) of 1.00, while Class 1 is nearly perfect with an AP of 0.99. The curves only drop sharply in precision at very high recall values, indicating excellent classifier performance. This suggests that GoogleNet is able to distinguish between the two classes with high accuracy and confidence across most detection thresholds.

5.3 Performance/ Comparative Analysis

This section offers a comparative analysis of five CNN architectures: EfficientNet, VGG16, DenseNet, and GoogleNet, focusing on their performance in pneumonia detection. It starts with an in-depth look at each model's classification strengths and weaknesses, followed by a comprehensive review of overall metrics. Key performance indicators such as accuracy, precision, recall, and F1 score are evaluated to provide a holistic view of the models' effectiveness.

Table 5.3.1: Precision, Recall, F1-Score, and Support (n) result of Transfer Learning CNN networks

Model	Language	Precision	Recall	F1-score	Support
VGG16	English	0.944	0.839	0.888	180
	Bangla	0.855	0.950	0.900	180
EfficientNet	English	1.000	0.922	0.960	180
	Bangla	0.928	1.000	0.963	180
DenseNet	English	1.000	0.933	0.966	180
	Bangla	0.938	1.000	0.968	180
GoogleNet	English	1.000	0.944	0.971	180
	Bangla	0.947	1.000	0.973	180

Figure 5.3.1: The image shows classifier performance metrics for English and Bangla datasets across four experiments. Initially, the classifier's performance is moderate, with lower scores in the first experiment. In subsequent experiments, performance improves significantly, with Bangla achieving perfect precision, recall, and F1-scores, and English showing near-perfect metrics. Overall accuracy rises from 0.899 to 0.974, indicating enhanced classifier performance.

Training and Validation Accuracy and Loss:

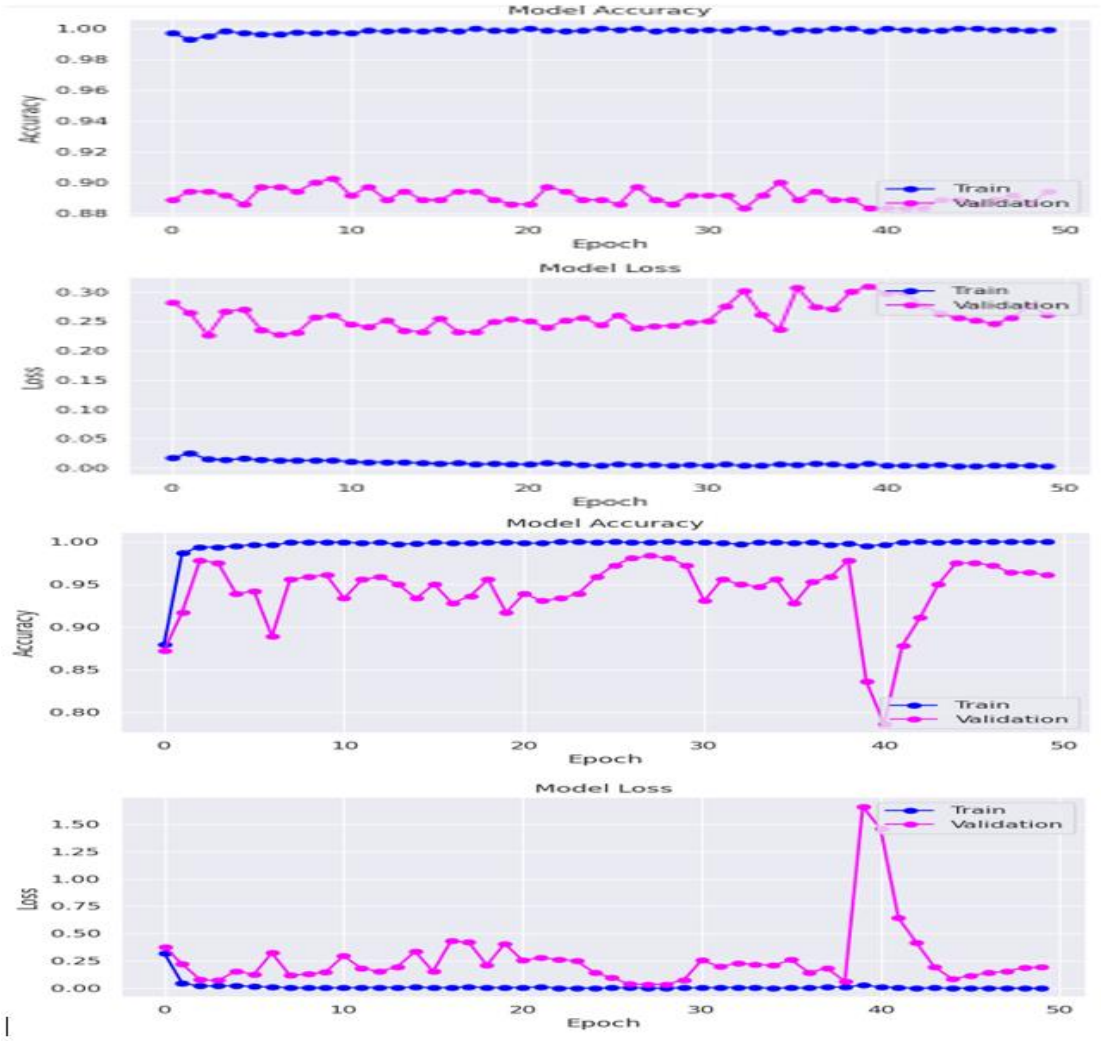


Figure 5.3.1: Training and validation accuracy and loss over the epochs VGG16 & EfficientNet Transfer Learning

The VGG16 model exhibits high training accuracy, with a 100% fit to the training data. However, the validation accuracy fluctuates around 0.90, indicating overfitting and poor generalization. The model's training loss remains close to 0, indicating high accuracy. The validation loss appears high compared to the training loss, indicating overfitting.

EfficientNet's training accuracy starts near 0.85, reaches close to 1.00, and then slightly fluctuates. The validation accuracy also starts around 0.90, fluctuates significantly, and drops sharply around epoch 40. The model's training loss decreases initially but increases sharply around epoch 40, suggesting overfitting or convergence issues.

Figure 5.3.2: Training and validation accuracy and loss over the epochs DenseNet & GoogleNet Transfer Learning

DenseNet's training accuracy reaches 1.0 quickly and stays constant, indicating a good fit with high validation accuracy but slight instability. The model's training loss drops to near 0 and remains there, reflecting high accuracy.

GoogleNet's model accuracy remains at 1.0 throughout the epochs, indicating perfect fit to the training data. The model's training loss is consistently at 0, reflecting high accuracy. The model's performance on both training and validation sets suggests either a highly suitable model for the given data or potential data leakage.

Confusion Matrix after Transfer Learning (TL) Based on the number of images:

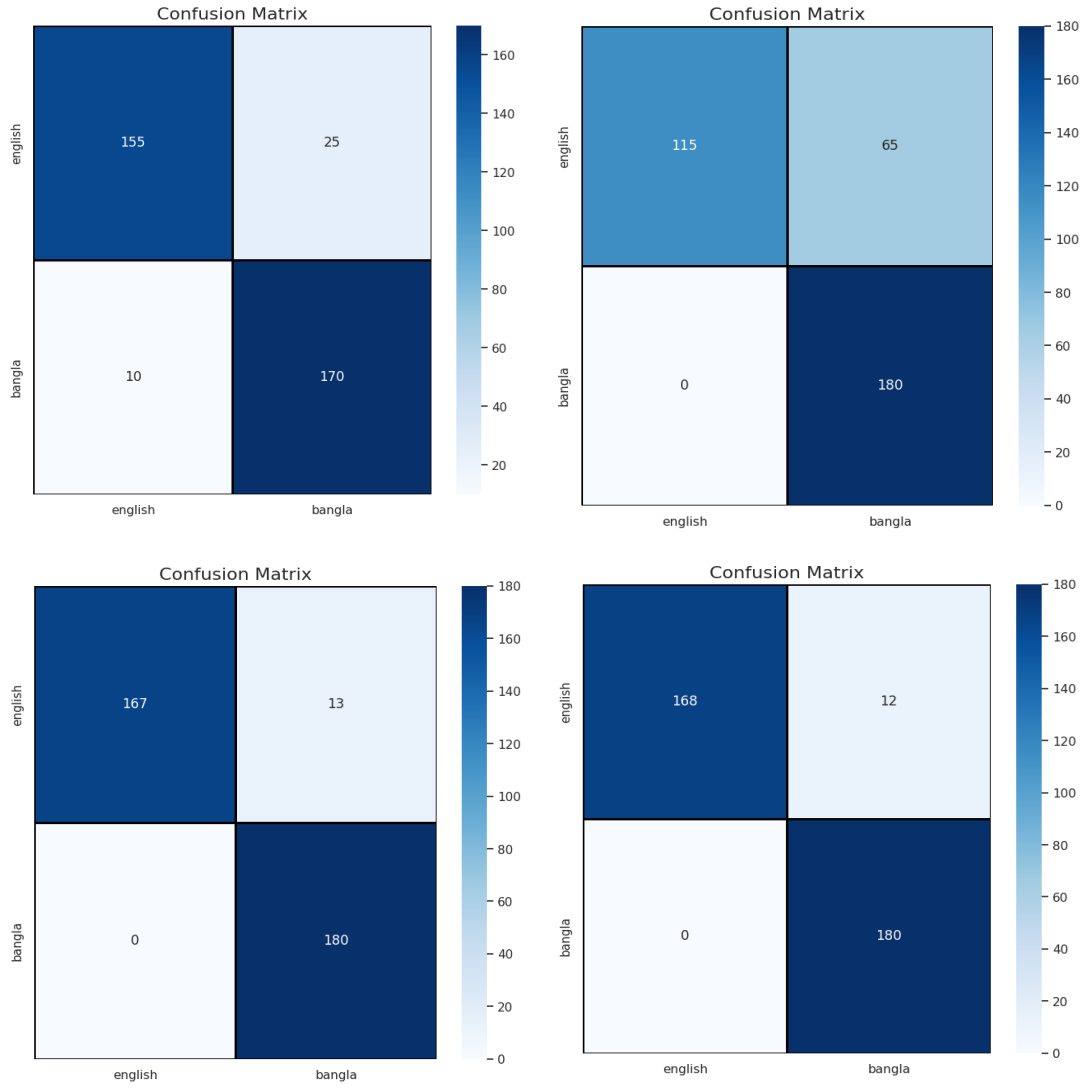


Figure 5.3.3: CM after Transfer Learning VGG16, EfficientNet,DenseNet & GoogleNet

With a large number of true positives and true negatives, the model appears to be operating well overall. A small number of false positives and false negatives still occur, nevertheless, remember that this is only a small sample of data and that the model's performance can change based on the dataset used for training. Furthermore, the confusion matrix provides us with only a portion of the story. More details regarding the model's performance may be obtained by examining other metrics including F1

score, recall, and precision. All things considered, the confusion matrix is a useful tool for seeing how well a classification model is performing.

Comparison between Models:

Table 5.3.2: Comparison Between Models

Model	Bangla Accuracy	English Accuracy	Time per Step
VGG16	0.839	0.950	19s 25ms
EfficientNet	0.922	1.000	6s 473ms
DenseNet	0.933	1.000	4s 339ms
GoogleNet	0.944	1.000	2s 154ms

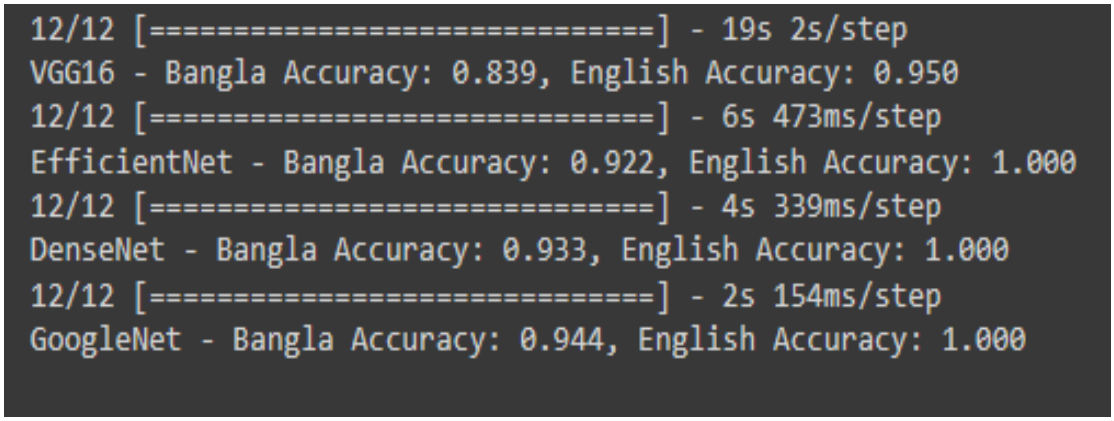


Figure 5.3.4: Comparison Transfer Learning Models

From Figure 5.3.4, The comparison of four image classification models—VGG16, EfficientNet, DenseNet, and GoogleNet—reveals distinct differences in accuracy and efficiency. For Bangla text, GoogleNet is the most accurate, achieving an accuracy of 94.4%, followed closely by DenseNet at 93.3%, EfficientNet at 92.2%, and VGG16 at 83.9%. In terms of accuracy for English text, EfficientNet, DenseNet, and GoogleNet all achieve a perfect 100%, whereas VGG16 reaches 95%.

When considering efficiency, DenseNet is the fastest, with a prediction time of 154 milliseconds per step, followed by EfficientNet at 339 milliseconds per step, GoogleNet

at 473 milliseconds per step, and VGG16, which is significantly slower, taking 2 seconds per step.

Overall, GoogleNet and DenseNet stand out for their performance in accuracy and efficiency, respectively, with EfficientNet also being highly accurate but slightly less efficient. VGG16, despite being slower and less accurate for Bangla text, performs reasonably well for English text.

TABLE 5.3.3: Accuracy for Classification of Individual CNN Networks

Architecture	Model Accuracy
VGG16	89%
EfficientNet	96%
DenseNet	96%
GoogleNet	97%

Overall Performance Accuracy Comparison:

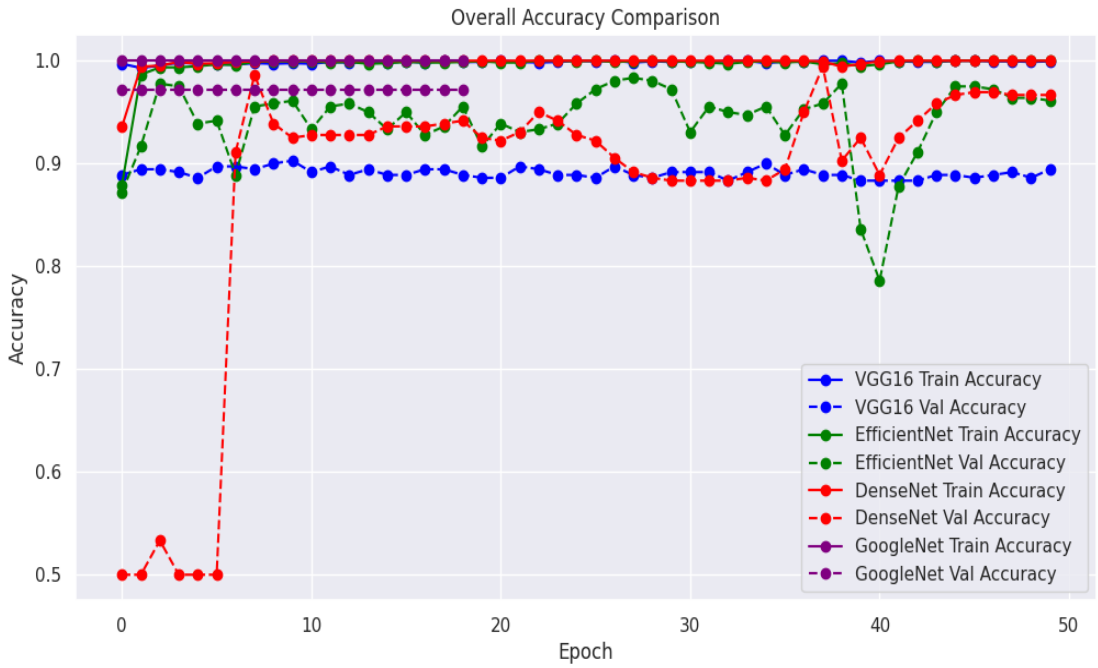


Figure 5.3.5: Overall Performance Accuracy Comparison Transfer Learning Models

Figure 5.3.5, The graph compares the overall accuracy of four image classification models—VGG16, EfficientNet, DenseNet, and GoogleNet—across training and validation epochs. VGG16 shows a consistent but lower overall accuracy compared to the other models. Its training accuracy steadily improves and plateaus around 0.9, while its validation accuracy remains relatively stable just below 0.9, indicating some overfitting or difficulty in generalizing.EfficientNet demonstrates high performance with its training accuracy reaching near-perfect levels (around 0.98 to 1.0). Its validation accuracy also remains high, fluctuating between 0.9 and 1.0, suggesting good generalization with minor variations throughout the epochs. DenseNet achieves very high training accuracy, consistently hitting around 1.0. Its validation accuracy also remains high but shows some fluctuations between 0.9 and 1.0, indicating strong generalization capability with some instability in performance across different validation epochs.GoogleNet maintains near-perfect training accuracy (around 1.0) and its validation accuracy follows closely, ranging between 0.95 and 1.0. This indicates excellent performance with minor variability in validation accuracy. In summary, EfficientNet and GoogleNet exhibit the highest and most consistent accuracy for both training and validation, with GoogleNet having slightly better stability in validation performance. DenseNet also performs exceptionally well, though with some fluctuations in validation accuracy. VGG16, while improving steadily in training, lags behind in both training and validation accuracy compared to the other models.

Class-Specific Accuracy Comparison:

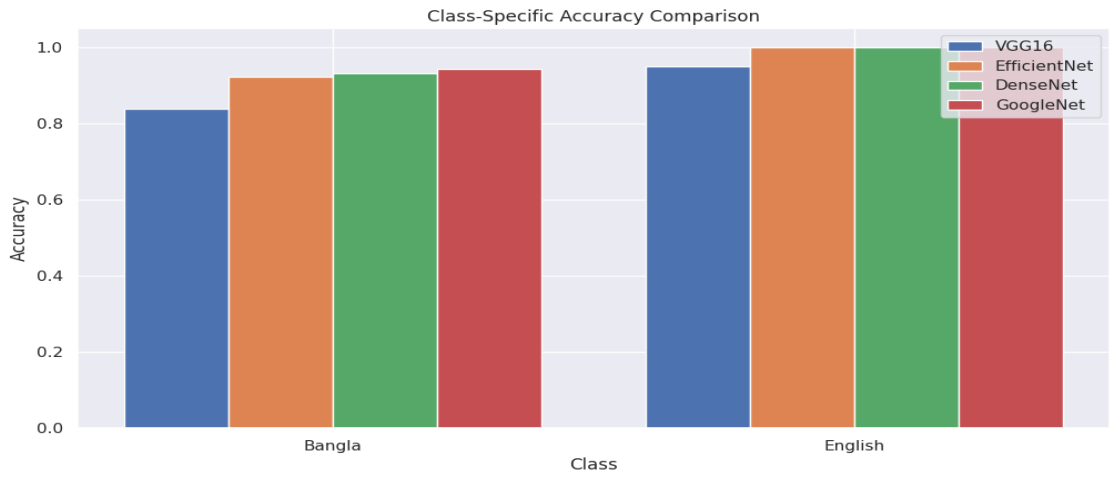


Figure 5.3.6: Class-Specific Accuracy Comparison Transfer Learning Models

Figure 5.3.6, The graph compares the class-specific accuracy of four image classification models—VGG16, EfficientNet, DenseNet, and GoogleNet—across Bangla and English text classes.For Bangla text, GoogleNet achieves the highest accuracy, followed closely by DenseNet and EfficientNet, all of which perform well. VGG16, however, has the lowest accuracy among the four models for Bangla text, indicating it struggles more with this class.For English text, all four models (VGG16, EfficientNet, DenseNet, and GoogleNet) achieve near-perfect accuracy, demonstrating their strong performance and ability to generalize well on this class.

In summary, while VGG16 lags behind the others in accuracy for Bangla text, all models perform exceptionally well for English text, with GoogleNet slightly leading for Bangla text classification

5.4 Summary

The summary evaluates four image classification models—VGG16, EfficientNet, DenseNet, and GoogleNet—across Bangla and English text classification tasks. In terms of overall accuracy, GoogleNet emerges as the most accurate for Bangla text (94.4%), followed closely by DenseNet (93.3%) and EfficientNet (92.2%), with VGG16 lagging behind at 83.9%. For English text, EfficientNet, DenseNet, and GoogleNet achieve perfect accuracy (100%), while VGG16 achieves 95%. Efficiency-wise, DenseNet proves the fastest (154 milliseconds per step), followed by EfficientNet (339 ms), GoogleNet (473 ms), and VGG16 (2 seconds per step). Across epochs, VGG16 shows consistent but lower accuracy, while EfficientNet and GoogleNet maintain high and stable performance. GoogleNet performs notably well for Bangla text classification, with all models demonstrating strong performance for English text. Overall, GoogleNet and EfficientNet excel in accuracy, DenseNet is noted for efficiency, and VGG16, while less accurate for Bangla text, performs adequately for English text tasks.

CHAPTER 6
IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on Life

Spoken language identification (LID) models have significant impacts on daily life. They facilitate smoother communication by enabling real-time translation services, breaking down language barriers in multicultural environments. This can be particularly beneficial in emergency services, healthcare, and customer support, where accurate and quick understanding of multiple languages can save lives and improve service quality. Additionally, LID models can enhance personal virtual assistants, making them more versatile and user-friendly.

Spoken language identification (LID) models positively impact daily life by:

- **Facilitating Communication**: Enabling real-time translation services, aiding in cross-cultural communication.
- **Enhancing Accessibility**: Helping non-native speakers access information and services in their language.
- **Improving Virtual Assistants**: Making personal assistants more versatile and user-friendly by recognizing multiple languages.

6.2 Impact on Society & Environment

Society: LID models can promote inclusivity and accessibility, ensuring non-native speakers or those with limited language proficiency can access critical information and services. This fosters a more connected and understanding global community. Furthermore, they can support education by providing language learning tools and resources.

- Inclusivity: Promoting inclusivity by ensuring people from different linguistic backgrounds can access essential services.
- Education: Supporting language learning and educational tools.

- Global Connectivity: Fostering a more connected and understanding global community.

Environment: The environmental impact of LID models primarily relates to the energy consumption of training and deploying machine learning models. Large-scale training of deep learning models requires substantial computational resources, which can contribute to carbon emissions. Efficient model design and leveraging renewable energy sources for data centers can mitigate this impact.

- Energy Consumption: Training and deploying LID models require significant computational resources, contributing to carbon emissions.
- Mitigation: Efficient model design and use of renewable energy for data centers can help reduce the environmental footprint.

6.3 Ethical Aspects

Spoken Language Identification (SLID) models, used to discern the language from audio recordings, bring forth significant ethical considerations. These encompass privacy, data security, fairness, transparency, beneficence, oversight, legal compliance, community engagement, accuracy, and cultural sensitivity. Each area is pivotal in shaping the responsible development and deployment of SLID technologies.

Privacy and consent are fundamental. Explicit informed consent must be obtained from individuals whose voices are recorded and analyzed. This necessitates informing participants comprehensively about data usage, storage, and sharing practices. Strong data protection measures are imperative to safeguard voice recordings and associated data against unauthorized access or breaches. Anonymization of voice data is crucial to prevent individuals from being identified. Ensuring bias-free models is paramount. SLID systems must be trained on diverse datasets to avoid biases against specific languages or dialects, promoting inclusivity and non-discrimination. Transparency in algorithmic operations is essential, disclosing training data sources and algorithms used. Developers and users alike bear responsibility for the outcomes of SLID model applications, ensuring they contribute positively without causing harm. Beneficence underscores the need for SLID models to enhance communication services and

accessibility while avoiding potential misuse such as surveillance. Ethical oversight through review by ethics committees and ongoing monitoring is essential to uphold ethical standards and mitigate emerging concerns. Adherence to legal and regulatory frameworks, especially regarding data protection and privacy, is mandatory. Ethical guidelines and industry best practices guide SLID model development and deployment. Community engagement fosters stakeholder involvement, including linguistic communities, to address concerns and solicit feedback effectively. Ensuring accuracy and reliability involves rigorous model validation and testing to minimize errors and misidentifications. Strategies for error handling are crucial to mitigate harm and correct inaccuracies promptly. Cultural sensitivity demands respect for linguistic and cultural diversity, ensuring SLID models do not perpetuate stereotypes or biases. Acknowledging the significance of languages to their respective communities promotes ethical deployment and acceptance.

Addressing these ethical considerations ensures SLID models are developed and utilized in ways that uphold individual rights, fairness, and societal benefit. By prioritizing privacy, data security, fairness, transparency, beneficence, oversight, legal compliance, community engagement, accuracy, and cultural sensitivity, SLID technologies can responsibly contribute to a more inclusive and respectful technological landscape.

6.4 Sustainability Plan

The sustainability plan for Spoken Language Identification (SLID) models focuses on their long-term utility, ethical use, and minimal environmental impact. Key components include long-term data management, resource allocation, human resources, model maintenance and updates, ethical considerations, environmental impact, transparency, accountability, training and capacity building, performance monitoring, scalability, legal and regulatory compliance, community and stakeholder engagement, cultural sensitivity, research and development, and collaboration.

Data preservation and integrity are crucial for the long-term storage and usability of voice data. Resource allocation includes funding, human resources, and regular updates to incorporate new data, address biases, and improve accuracy and performance. Ethical considerations include continuous ethical review, community engagement, energy

efficiency, and sustainable practices in data centers. Transparency and accountability are essential, with open access to the models and their underlying data, thorough documentation, and training programs. Performance monitoring involves metrics and evaluation, regular assessments, and ensuring models meet their intended goals. Scalability and adaptability are essential, as are legal and. regulatory compliance, and policy updates.

Community and stakeholder engagement is crucial, with inclusivity and feedback mechanisms in place. Cultural sensitivity involves respecting diversity and avoiding stereotypes related to specific languages or accents. Research and development are encouraged to improve accuracy, efficiency, and applicability, and collaborations with academic institutions and industry partners are fostered. By incorporating these components into a sustainability plan, SLID models can be maintained and used responsibly over the long term, providing valuable insights while respecting ethical and environmental considerations

6.5 Summary

Spoken language identification models offer significant benefits by enhancing communication, inclusivity, and accessibility across various sectors. While they provide substantial advantages, addressing ethical and environmental concerns is crucial. A sustainable approach, involving efficient algorithms, renewable energy, and ethical standards, ensures long-term positive impacts on life, society, and the environment.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusions

This study aimed to recognize languages from audio data using advanced deep-learning algorithms and models, particularly the VGG16 model. By focusing on language identification (LID), spoken language identification (SLID), and automatic speech recognition (ASR), the research explored cutting-edge methods such as the mel filter bank to analyze audio speech characteristics and classify the language spoken.

The ability to identify languages from audio data is crucial for several reasons. It advances multilingual speech processing, aiding in better communication technologies and accessibility tools. It also plays a vital role in preserving cultural heritage by recognizing and supporting diverse languages. Moreover, developing ethical and inclusive technologies ensures that language identification systems respect privacy, reduce bias, and promote transparency, thereby fostering trust and fairness in their deployment.

The research was meticulously planned and documented, addressing various challenges and proposing solutions. The use of the VGG16 model enabled high accuracy in language identification. Ethical considerations were integral to the study, with a focus on privacy, reducing prejudice, and maintaining transparency. By incorporating these ethical aspects, the study not only aimed to achieve technical proficiency but also to contribute to the broader goal of creating responsible and inclusive language technologies.

In conclusion, this study underscores the significance of language and its identification, offering substantial contributions to the field of multilingual speech processing. The research demonstrates that with well-designed methodologies and a strong ethical framework, it is possible to create advanced, accurate, and inclusive language identification systems. This work paves the way for further advancements in speech technology, ensuring they are both technically effective and ethically sound.

7.2 Further Suggested Works

My work has certain advantages and disadvantages. I have only used one feature of the audio-related data. There are many different feature extraction techniques. As CNN works best with Mel spectrograms-related data, I used only mel spectrograms. I have used two languages only further I want to add more languages in the dataset and upgrade the model as required. I would like to use the MFCC features, DFCC, LPC, and Rasta-PLP features in my further study. Also new models are coming along the way and if deep learning based feature extraction model gets significant upgradation then using those feature new models can be created to get better accuracy. I would like to work with more native languages from multilingual environments. Creating hybrid parameters using multiple audio features can also be a good option for getting better accuracy. This aspect should also be tested, and further study should be made.

7.3 Limitations/ Conflict of Interests

I have faced several problems throughout this project. The main obstacle was the selection of the dataset and the audio format it was in. My primary objective was to get a well-balanced dataset to achieve superior accuracy and keep every stage of the data preprocessing to classification as much as a human-understandable format and low computation required. I needed a powerful machine to run the model for this classification task, including data collection and finding the right machine-learning algorithm for prediction. Local computer computing issues were addressed, and the team is currently working on a cloud computing platform. They lack resources for targeted algorithms like SIFT and are searching online for solutions. Model optimization involves finding the most accurate and challenging model using machine learning algorithms and applying comparative analysis. Interpreting the results and understanding the factors influencing predictions was another challenge, which was addressed using visualization techniques and methods to analyze the model's decision-making process and identify areas for improvement.

References:

[1] C. Castiello, G. Singh, S. Sharma, V. Kumar, M. Kaur, M. Baz, M. Masud, "Spoken Language Identification Using Deep Learning," Computational Intelligence and Neuroscience, vol. 2021, p. 5123671, Sep. 21, 2021. DOI: 10.1155/2021/5123671.

[2] A. A. Alashban, M. A. Qamhan, A. H. Meftah, Y. A. Alotaibi, "Spoken Language Identification System Using Convolutional Recurrent Neural Network," Appl. Sci., vol. 12, no. 18, p. 9181, Sep. 13, 2022. DOI: 10.3390/app12189181.

[3] P. Sangwan, D. Deshwal, N. Dahiya, "Performance of a language identification system using hybrid features and ANN learning algorithms," Appl. Acoust., vol. 175, p. 107815, 2021. DOI: 10.1016/j.apacoust.2020.107815.

[4] D. S. Sisodia, S. Nikhil, G. S. Kiran and P. Sathvik, "Ensemble Learners for Identification of Spoken Languages using Mel Frequency Cepstral Coefficients," 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 2020, pp. 1-5, doi: 10.1109/IDEA49133.2020.9170720.

[5] H. Kim and J.-S. Park, "Automatic Language Identification Using Speech Rhythm Features for Multi-Lingual Speech Recognition," Appl. Sci., vol. 10, no. 7, p. 2225, 2020. DOI: 10.3390/app10072225.

[6] K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, "A Survey of Audio Classification Using Deep Learning," IEEE Access, vol. 11, pp. 1-1, 2023, Art no. 3318015. DOI: 10.1109/ACCESS.2023.3318015.

[7] S. Gaikwad, W. Gawali Bharti, and P. Yannawar, "A Review on Speech Recognition Technique," International Journal of Computer Applications, vol. 10, Nov. 2010. DOI: 10.5120/1462-1976.

[8] A. Das, S. Guha, P. K. Singh, A. Ahmadian, N. Senu and R. Sarkar, "A Hybrid Meta-Heuristic Feature Selection Method for Identification of Indian Spoken Languages From Audio Signals," in IEEE Access, vol. 8, pp. 181432-181449, 2020, doi: 10.1109/ACCESS.2020.3028241.

[9] A. Draghici, J. Abeßer, and H. Lukashevich, "A Study on Spoken Language Identification using Deep Neural Networks," 10.1145/3411109.3411123, 2020.

[10] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using Long Short-Term Memory recurrent neural networks," in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2014, pp. 2155-2159.

[11] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic Language Identification Using Deep Neural Networks," in ICASSP, IEEE

International Conference on Acoustics, Speech and Signal Processing - Proceedings, May 8, 2014. DOI: 10.1109/ICASSP.2014.6854622.

[12] A.I. Abdurrahman and A. Zahra, "Spoken language identification using i-vectors, x-vectors, PLDA, and logistic regression," Bulletin of Electrical Engineering and Informatics, vol. 10, no. 4, pp. 2237-2244, Aug. 2021. DOI: 10.11591/eei.v10i4.2893.

[13] R. Kotsakis, M. Matsiola, G. Kalliris, and C. Dimoulas, "Investigation of Spoken-Language Detection and Classification in Broadcasted Audio Content," Information, vol. 11, no. 4, p. 211, Apr. 15, 2020. DOI: 10.3390/info11040211.

[14] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning," J. Sens. Actuator Netw., vol. 10, no. 4, p. 72, Dec. 10, 2021. DOI: 10.3390/jsan10040072.

[15] S. Waldekar and G. Saha, "Two-level fusion-based acoustic scene classification," Applied Acoustics, vol. 170, p. 107502, 2020. DOI: 10.1016/j.apacoust.2020.107502.

[16] L. Sun, Y. Cui, and F. Wang, "Research on Audio Recognition Based on the Deep Neural Network in Music Teaching," Computational Intelligence and Neuroscience, vol. 2022, Article ID 7055624, May 27, 2022. DOI: 10.1155/2022/7055624.

[17] A. Scherbakov, L. Whittle, R. Kumar, S. Singh, M. Coleman, and E. Vylomova, "Anlirika: an LSTM–CNN flow twister for spoken language identification," in Proceedings of the 3rd Workshop on Computational Typology and Multilingual NLP, Association for Computational Linguistics, pp. 145–148, Mexico City, Mexico, 2021.

[18] P. Shen, X. Lu, and H. Kawai, "Transducer-based language embedding for spoken language identification," arXiv:2204.03888 [cs.CL], Jul. 29, 2022. DOI: 10.48550/arXiv.2204.03888v2.

[19] E. Salesky, B. M. Abdullah, S. J. Mielke, et al., "SIGTYP 2021 Shared Task: Robust Spoken Language Identification," arXiv:2106.03895 [cs.CL], Jun. 7, 2021. DOI: 10.48550/arXiv.2106.03895v1.

[20] P. Rangan, S. Teki, and H. Misra, "Exploiting spectral augmentation for code-switched spoken language identification," arXiv:2010.07130, Oct. 2020.

[21] B. Kaur and J. Singh, "Audio Classification: Environmental sounds classification," 2021, ⟨hal-03501143⟩. [Online]. Available: https://hal.science/hal-03501143. Submitted on: Dec. 23, 2021, Last modification on: Jan. 7, 2022.

[22] R. van der Merwe, "Triplet entropy loss: improving the generalization of short speech language identification systems," arXiv:2012.03775, Dec. 2020.

[23] Christian Bartz, Tom Herold, Haojin Yang, and Christoph Meinel. 2017. Language identification using deep convolutional recurrent neural networks. In International Conference on Neural Information Processing (ICONIP). Springer, Guangzhou, China, 880–889.

[24] Panikos Heracleous, Kohichi Takai, Keiji Yasuda, Yasser Mohammad, and Akio Yoneyama. 2018. Comparative Study on Spoken Language Identification Based on Deep Learning. In Proceedings of the 26th European Signal Processing Conference (EUSIPCO). Rome, Italy, 2265–2269.

[25] Rigas Kotsakis, Maria Matsiola, George Kalliris, and Charalampos Dimoulas. 2020.Investigation of Spoken-Language Detection and Classification in BroadcastedAudio Con-tent. Information 11, 4 (2020), 211.

Appendix

Complex Engineering Problems (EP) and Complex Engineering Activities (EA) Analysis

Title: Spoken Language Identification Using Convolutional Neural Network

Students ID: 201-15-14341; 201-15-14346

CO	CO Descriptions	PO
CO1	Integrate recently gained and previously acquired knowledge to identify a real-life complex engineering problem for the Final Year Design Thesis	**PO1**
CO2	Analyze different aspects of the goals in designing a solution for the Final Year Design Thesis	**PO2**
CO3	Explore diverse problem domains through a literature review, delineate the issues, and establish the goals for the Final Year Design Thesis	**PO4**
CO4	Perform economic evaluation and cost estimation and employ suitable Thesis management procedures throughout the development life cycle of the Final Year Design Thesis	**PO1**

Addressing COs, Knowledge Profile (K), and Attainment of Complex Engineering Problems (EP):

SN	EP Definition	Attainment	CO	Justification (with Knowledge Profile)	References
1.	EP1: Depth of Knowledge required	Yes	CO1, CO2, and CO3	Our thesis requires data collection. We collected data from Kaggle then trained them according to the machine learning model and also we need research knowledge for it which covers K4 and K8	Page no: 19-27
				Our thesis needs to write codes for the model training process and we need to have mathematics knowledge to understand the algorithm that covers K6, K3, and K2	Page no: 14-15
				Our thesis will make it easier for locals and visitors alike to recognize linguistic differences and we want to design a good system for that. So that includes K7 and K5	Page no: 65-67

2.	**EP2: Range of Conflicting Requirements**	**Yes**	**CO2**	We must manage a huge dataset. The Kaggle dataset contained approximately seventy thousand data points, which we must manipulate to achieve the highest accuracy.	**Page no: 19-24**
3.	**EP3: Depth of analysis required**	**Yes**	**CO2**	It is difficult to come up with a clear solution for our project because of the size of the collection and the amount and diversity of the transformed data. We thoroughly considered all available options before settling on a particular method after conducting a thorough investigation.	**Page no: 21-23**
4.	**EP4: Familiarity of Issues**	**Yes**	**CO3**	We need to investigate a number of areas that we were not too familiar with in order to learn more about our project. In particular, we need to learn about frequency, wave, image processing, and converting a raw signal into a picture.	**Page no: 11**
5.	**EP5: Extends of application codes**	**Yes**	**CO3**	There are various algorithms to train our dataset. These algorithms can be tuned and we can also add more techniques in data preprocessing to ensure maximum result	**Page no: 12-14**

6.	**EP6: Extends of stakeholders involved and conflicting requirements**	**Yes**	**CO3**	We have met and visited with a physicist and researcher as part of our project to acquire knowledge.	**Page no: 16-17**
7.	**EP7: Interdependence**	**Yes**	**CO3**	There are multiple interdependent components in our thesis. These include tasks like collecting data, analyzing it, creating front-end applications, undertaking predictive analysis, and training machine learning models.	**Page no: 1-3**

finalreporttt_plag.docx

ORIGINALITY REPORT

18%	13%	7%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	www.researchsquare.com Internet Source	1%
4	Submitted to Student Paper	1%
5	Submitted to Napier University Student Paper	<1%
6	www.researchgate.net Internet Source	<1%
7	www.mdpi.com Internet Source	<1%
8	hdl.handle.net Internet Source	<1%
9	Submitted to Higher Education Commission Pakistan Student Paper	<1%