Independent University **Bangladesh (IUB)**

IUB Academic Repository

Computer Science and Engineering | Undergraduate Thesis

2025-05

Advancing Bengali Dialect Identification (DiD) Through The BengDiDa Dataset & Dialect Classification

Kabir, Md Zakaria

IUB

http://ar.iub.edu.bd/handle/11348/1004

Downloaded from IUB Academic Repository



Advancing Bengali Dialect Identification (DiD) Through The BengDiDa Dataset & Dialect Classification

By

Md Zakaria Kabir, Md. Zannat Chowdhury Sagar

Student ID: 1921371, 1920392

Spring, 2025

Supervisor:

Md Asif Bin Khaled

Senior Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

May 3, 2025

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

We are conscious of the nature of plagiarism and the university's stringent anti-plagiarism policy. We confirm that this is our original work. Moreover, any software or textual work by others that is utilized in this project is acknowledged appropriately and in accordance with generally accepted academic standards.

Signature	Date
Md Zakaria Kabir	
Name	
Signature	Date
Md. Zannat Chowdhury Sagar	
Name	

Acknowledgement

We would like to start by thanking Almighty Allah for giving us the willpower, persistence, and ability to work hard, as well as for allowing us to complete our senior project. Our sincere thanks are due to Mr. Md Asif Bin Khaled, our Supervisor and Senior Lecturer at Independent University, Bangladesh. We are deeply grateful for his expert guidance, unwavering patience, generous allocation of time, and insightful advice on various facets of our project and the formulation of this comprehensive report. With his direction and assistance, our project developed without any problems. Furthermore, we express our gratitude to the committee members for their valuable contributions during the defence, making the experience constructive and enjoyable. Their perceptive critiques and invaluable recommendations significantly enriched our project. In addition to our academic mentors, our gratitude extends to our family and friends for their unwavering support and encouragement throughout our B.Sc. program, particularly during the challenging phases of this senior project.

We extend our deep appreciation to Independent University Bangladesh for providing a robust senior project program, which has played a crucial role in preparing and moulding us for the research industry.

Letter of Transmittal

May 3, 2025
Md Asif Bin Khaled
Senior Project Supervisor & Senior Lecturer
Department of Computer Science and Engineering,
Independent University, Bangladesh.

Subject: Senior project report on 'Advancing Bengali Dialect Identification (DiD) Through The BengDiDa Dataset & Dialect Classification'.

Dear Sir,

We are truly grateful for the chance to submit a senior project report to you on "Advancing Bengali Dialect Identification (DiD) Through The BengDiDa Dataset & Dialect Classification". This report is a comprehensive reflection of the knowledge and effort we put into the project under your supervision. During the project, we applied a diverse set of skills and emerging technologies, which have enriched our understanding and proficiency in this field. We believe that this report effectively fulfills its intended purpose. We are extremely grateful for your invaluable guidance, mentorship, knowledge, and support. We hope that this report meets your expectations and is deemed satisfactory.

Thank you for your continuous cooperation and invaluable assistance throughout the semester.

Yours sincerely,

Md Zakaria Kabir (ID: 1921371)
Md. Zannat Chowdhury (ID: 1920392)

Evaluation Committee

Supervision Panel

.....
Supervisor Md Asif Bin Khaled

Internal Examiners

.....
-

.....
-

External Examiners

.....
-

Contents

Attestation **i**

Acknowledgement **ii**

Letter of Transmittal **iii**

Evaluation Committee **iv**

1 Introduction **2**

2 Background and Motivation **4**

2.1 Importance of Dialect Identification 4

2.2 Challenges in Bengali Dialect Recognition 5

2.3 Role of Deep Learning in Speech Processing 6

2.4 Bengali Language and Dialects 7

3 Applications of Dialect Identification **8**

3.1 Automated Speech Recognition Systems 8

3.2 Sociolinguistic Analysis 9

3.3 Fake Speaker Detection 9

3.4 Personalized Voice Assistants 10

3.5 Speech Therapy and Rehabilitation 10

4 Dataset Development for Dialect Identification 11

4.1 Requirements for Dialect Corpora 11

4.2 Why Dataset Development Was Necessary? 12

4.2.1 Challenges in Creating the Dataset 12

4.3 BengDiDa Dataset 13

4.3.1 Overview 13

4.3.2 Dataset Building 16

4.3.3 Dataset Accessibility 28

4.3.4 Ethical Considerations in Data Collection 28

5 Systematic Literature Review on Dialect Identification 29

5.1 Methodology for Literature Review 29

5.1.1 PRISMA-2020 Framework Overview 29

5.1.2 Selection Criteria 30

5.1.3 Data Extraction Process 30

5.2 State of Research in Dialect Identification 31

5.2.1 Existing Methods for DID 31

5.2.2 Challenges in DID Research 31

5.2.3 Gaps in Bengali DID Studies 32

5.3 Comparative Analysis of DID Techniques 32

5.3.1 Machine Learning vs Deep Learning Approaches 32

5.3.2 Hybrid Models in DID 33

5.3.3 Role of Pretrained Models 33

6 Convolutional Recurrent Neural Networks (CRNN) for DID **34**

6.1 Introduction to CRNN Architecture 34

6.2 Convolutional Layers for Spatial Features 35

6.3 Recurrent Layers for Temporal Modeling 35

6.4 Advantages of CRNN 36

7 Experimental Configuration **38**

7.1 Hardware and Software Environment 38

7.2 Data Preparation 38

7.3 Model Architecture 39

7.3.1 Custom CRNN 39

7.3.2 ResNet50-Based CRNN 40

7.4 Model Optimization and Regularization 41

7.5 Hyperparameter Optimization 42

7.6 Training Configuration 43

8 Result Analysis **44**

8.1 Evaluation Metrics 44

8.2 Comparative Evaluation 44

8.3 Loss and Accuracy Curves 45

8.4 Confusion Matrix of Best Performing Model 47

9 Future Directions in Bengali DID **49**

9.1 Expansion of BengDiDa Dataset 49

9.1.1 Fieldwork for Data Collection 49

9.1.2 Inclusion of More Dialects 50
9.2 Advances in Self-Supervised Learning 50
9.2.1 End-to-End Learning Approaches 50
9.2.2 Applications in Dialectal Speech Processing 50
9.3 Parameter-Efficient Transfer Learning 50
9.3.1 Techniques for Model Adaptation 50
9.3.2 Efficiency in Resource-Constrained Environments 51
9.4 Integration with Real-World Applications 51
9.4.1 Voice Assistants 51
9.4.2 Speech-to-Text Systems 51
9.5 Broader implications of DID 51

10 Conclusion **53**

Bibliography **59**

List of Figures

4.1 Proposed dataset structure illustration 14

4.2 Proposed dataset structure illustration 16

4.3 Six-stage pipeline for building the BengDiDa corpus. 17

4.4 Geographic regions of the 20 selected Bengali dialects. 18

4.5 Left: raw (top) vs. noise-reduced (middle) waveforms, superimposed raw and noise-reduced (bottom). Right: corresponding spectrograms highlighting noise removal. 21

4.6 MOS distributions for raw vs. noise-reduced audio: (top) DNSMOS, (bottom) NISQA. 21

4.7 Visualization of the imputation process for low-resource dialects, showing segment selection, augmentation, and reinsertion. 23

4.8 Original vs. augmented waveforms. 24

4.9 Impact of augmentation on generating new variations of the original sample. 24

4.10 Feature extraction pipeline. 26

5.1 PRISMA protocol study selection process. 30

7.1 Architecture overview: Custom stacked CNN as feature encoder, followed by stacked RNN classifier. 41

7.2 Architecture overview: ResNet50 as feature encoder followed by RNN classifier. (5-class experiment). 43

8.1 Training and validation loss/accuracy curves for all major model configurations (Part 1). 46

8.1 Training and validation loss/accuracy curves for all major model configurations (Part 2). 47

8.2 Confusion Matrix for Best Performing Model (CNN-BiGRU on Raw - MFCC, GFCC, RPLP) 48

List of Tables

4.1 List of dialects included in the dataset 14

4.2 Audio specifications 15

4.3 Feature file specifications 16

4.4 Complete list of features stored in each Hickle file 27

7.1 Hardware and software configuration 38

7.2 Key layers and roles in ResNet50 + BiGRU architecture 42

8.1 Comparative performance of different models and configurations. 45

Abstract

Dialect Identification (DID) is necessary for recognizing linguistic diversity alongside improving speech technologies, especially for Bengali dialects which are considerably diverse in nature due to influence inflicted by geographical, cultural, and social factors. Phonetic similarities to adjacent dialects, the scarcity of comprehensive datasets, and the dynamic nature of speech sounds raise the difficulty of identifying these dialects. In this paper, we introduce the BengDiDa dataset, an extensive Bengali dialect speech corpus. BengDiDa includes 48,000 audio samples from 20 distinct dialects, designed to support the development of advanced modeling techniques. This research also looks at the efficacy of Convolutional Recurrent Neural Networks for the DiD task. The results highlight the importance of effective feature extraction and the management of spatial and channel-wise dependencies, thereby advancing automatic speech recognition and contributing to the preservation of linguistic heritage. Among evaluated architectures, a fine-tuned ResNet-50 backbone raises accuracy from 80.8 % (F1 = 0.83) to 85.3 % (F1 = 0.85), yet a purpose-built four-layer CNN + BiGRU achieves the best score of 90.3 % (F1 = 0.90) on raw MFCC/GFCC/RPLP features, underscoring the advantage of lightweight, task-specific designs.

Keywords— Bengali Dialect; Bangla Accent Speech Dataset; Regional Language; Dialect Identification; Speech Recognition; Multipurpose Dataset; Deep Learning; CRNN

Chapter 1

Introduction

Dialect Identification (DID) is a crucial area of research within the domain of speech and audio processing. It integrates computational methods to analyze and classify linguistic variations like dialects. Dialects represent the subtle deviation in patterns of pronunciation, vocabulary, grammar, and idiomatic expressions in speech within a particular language. These are influenced by geographical location, social class, ethnicity, and individual experiences [1]. These variations pose substantial challenges for automatic speech recognition (ASR) systems due to the diversity in spoken language across regions and communities. Bengali dialects, like many other languages, exhibit noteworthy heterogeneity in their linguistic features, making them an important subject for DID studies.

The biggest challenge for DID systems is extensive training and experimentation on ideal data sets. Much research shows significant gaps in newly found data sets for a multitude of languages and dialects. For instance, relative to the Arabic language, there is much research dedicated to the identification of Arabic dialects using Modern Standard Arabic as a cross-cutting orthography for unintelligible vernaculars [2]. This is not the case for the Bengali dialects. When comprehensive data sets do not exist across languages, training and testing comparable DID systems become less and less attainable [3]. Therefore, the construction of BengDiDa is imperative, as it compiles a large-scale audio data set composed of 20 Bengali dialects with 48,000 samples.

A lot of research work uses neural networks to provide results based on complicated classification tasks for the implementation of DID technology; for example, the Convolutional Recurrent Neural Network (CRNN) effectively implements convolution and recurrent neural layers to allow convolution for feature extraction and temporal modeling. Recently, noise reduction and specificity have helped in such modeling, which seeks to remedy gradient vanishing or explosion during training. In addition, CRNNs are supplemented by context-dependent word embeddings for successful learning [4]; BERT is the most common form.

The creation of BengDiDa represents a significant step forward in addressing the scarcity of resources for Bengali dialect identification. By providing a diverse selection of dialectal audio, this dataset facilitates rigorous experimentation with CRNN-based models. Furthermore, leveraging memory cells within Long Short-Term Memory (LSTM) networks can help overcome challenges related to temporal dependencies in sequential data [5]. Future work may focus on refining these architectures and incorporating multi-scale neural network products to capture finer linguistic details across dialects [6].

Speech language technology continues to evolve rapidly as researchers integrate elements from computer science, linguistics, and artificial intelligence to enable machines to interpret human language effectively [7]. The study of Bengali dialect identification contributes not only to advancements in automatic speech recognition but also to broader efforts aimed at preserving linguistic diversity and understanding regional communication patterns.

Chapter 2

Background and Motivation

2.1 Importance of Dialect Identification

DID is very important for studying language diversity and improving speech technology. Knowing how to tell apart different dialects helps with speech recognition, identifying people, solving criminal cases and saving languages. People's cultural and regional identity is often seen in their dialects' expressions of sound, grammar and meaning. These differences allow researchers to better design systems that are context & situation-aware.

Forensic experts are now using DID to deal with crimes where voices are used over the phone, like voice phishing. Predicting a person's region and dialects with AI methods helps solve criminal cases. The reason for this demand is the need for speaker profiling tools that uncover language patterns related to different regions or communities [8]. Probability density functions of feature vectors used in classification have been modelled by Gaussian Mixture Models, making use of changes in sound frequencies among dialects [9].

Intonation plays a significant role in recognizing spoken accents and native dialects. Studies on Indian languages have demonstrated how dynamic and static vowel behaviors contribute to effective identification processes without increasing computational costs due to dimensionality reduction techniques [10, 11]. Besides, CNNs have displayed powerful abilities to label English dialects by applying particular encoder-decoder frameworks [12].

Progress in text-based models with deep learning has not yet addressed these challenges in speech-based systems because critical data is lost during the feature extraction. As a result, there is a need for methods that can both keep important data from a language and achieve better classification [4]. End-to-end neural networks provide a promising solution because they map speech sounds directly onto character sequences [5].

All in all, the work done in DID encourages technological advancements and also helps keep

track of various regional language variations. Through reviews and by growing the data used, researchers can develop new ways for advanced modeling to suit a variety of languages and language groups.

2.2 Challenges in Bengali Dialect Recognition

The Bengali Dialect Recognition problem is complicated by many factors, like the richness of Bengali language, inadequate datasets and arduous computations. As there are so many different Bengali dialects with their individual pronunciations, vocabulary and tonal patterns, distinguishing between Bengali dialects is difficult. For this reason, classifiers sometimes have difficulty separating Bengali from nearby languages such as Assamese and Odia, because they have similar sounds. One can see this overlap in the pronunciation of common words like "rice" and "fish" with subtle existing differences [13]. Because the two groups sound very similar, models are needed that can pick up on tricky acoustic details.

As there are no standard sets of data for Bengali dialect identification, the situation is made even more difficult. Even though projects such as BengDiDa have brought together examples from many dialects, difficulties remain in getting samples from all regional variations. Choosing the right balance between the sexes and their recording guarantees fairness in the final model result. For example, The makeup of speakers and the quality of their recordings differ greatly from area to area in Barisal and Noakhali [14]. Since biases may be introduced in the data, machine learning models that rely on this information are likely to produce biased results.

Bengali dialects are known for these sounds always changing dynamically which presents an extra challenge. The burden of stratifying plosives and affricates is due to their brief and changeable nature, plus their interaction with surrounding vowels [15]. Distinguishing these features calls for advanced methods of preprocessing and strong feature extraction. Other approaches that match MFCCs and GMMs have performed well in the past, suggesting they should be localized or adapted for the Bengali dialect [16].

How well we can reduce noise is also a crucial factor affecting model accuracy. Important elements of the sound that are needed for proper classification can be hidden under environmental noise or other sound anomalies. Because it helps distinguish different dialects well, the i-vector model is common for dialect recognition [17]. Nevertheless, recognizing dialects in Bengali needs special tuning, due to the language's wide range of phonetics.

Not having access to transcriptions is a big roadblock for Bengali dialect processing. Experts often solve this issue by using phone recognition language modeling (PRLM) or parallel PRLM [18]. Still, the complexities in Bengali speech may not be fully caught without extra linguistic annotations.

Scaling models for corpuses as large as BengDiDa is still troubled by high computation demands. While recent achievements show that accuracy can be increased without adding more features, further changes are necessary to use larger amounts of data successfully. [11], further changes are necessary to use larger amounts of data successfully.

Solving these issues requires new modeling approaches, bigger balanced datasets and improved ways of preparing data for Bengali dialects.

2.3 Role of Deep Learning in Speech Processing

Advancements in deep learning mean speech processing can now solve tough challenges, including identifying various dialects, checking identities of speakers and sensing their emotions. The field has grown thanks to Deep Learning's ability to model complex patterns found in audio using networks like CNNs, RNNs and CRNNs. They make use of both feature hierarchy learning and temporal sequence modeling to identify connections in speech signals at both local and global levels.

The authors of [19] note that recurrent neural networks are valuable for learning time-domain patterns from speech. RNN-based systems skip long-aligned decoding processes, concentrating on strong features. Introducing random Gaussian noise during training improves this approach and reduces overfitting. Such approaches are necessary for Bengali DID because they permit the recognition of subtleties in phonetics and prosody among different dialects.

Extracting features is a very important step in speech processing systems that use deep learning. Researchers note in [3] that MFCCs are preferred in modeling acoustic scenarios due to their similarity to how we perceive speech. By decreasing noise, we help the process of feature extraction so that useful details are kept and worthless details are removed. Preprocessing the input data in this way supports the goals of CRNN-based DID systems for reliable classification.

Improvements in system performance have been seen when both spectral and prosodic features are used. According to the studies in [20], combining these elements during training increases recognition rates over training them individually. Spectral characteristics include information in the frequency domain, while prosodic characteristics include the rhythm and intonation patterns in speech. When used together, they extensively represent speech signals, helping to distinguish between similar dialects.

Doing preprocessing is essential for deep learning in speech recognition tasks. Following the work in [21], determining speech boundaries helps to properly cut out segments of the data for better training of analysis models. It matters especially for CRNN designs which learn sequenced information effectively.

Normalisation methods help by making the features input to the model more in a more standard

manner for all present dialect or emotion someone speaks with. To fix the incongruences of phoneme-level and word-level features in [8], the authors normalized them using a reference dialect showing neutral patterns. This allows features to be represented in a consistent manner, lowering risks from extra variables and helping improve accuracy.

Researchers now compare traditional machine learning techniques with deep learning strategies. Authors in [22] examine how Random Forests and CNN models perform on tasks related to accents. Even though engineered features are used in traditional systems, deep learning can discover multiple levels of meaning directly from audio which makes it more flexible and easy to use on larger datasets.

All in all, deep learning is growing more important for speech processing with innovative neural network architectures, methods of finding features, data preprocessing techniques and multimodal integration. As a result of filling in the gaps found in systematic reviews, future studies could pursue new model approaches that would make existing technologies more successful [16, 4, 23, 24].

2.4 Bengali Language and Dialects

Bengali evolved from Magadhi Prakrit and Sanskrit, which has also given birth to many languages spoken in the sub-continental region now. Hence, Bengali as a language has deep old roots, but comparatively new branches or dialects. Today, it ranks fifth by native speakers and seventh overall. The tongue stretches from the Sundarbans to Assam and into migrant pockets overseas [13, 25]. Standard Colloquial Bangla (SCB) sits in Kolkata–Dhaka media, yet rural hills and river deltas whisper their own phonemes, retroflex taps, nasal vowels. Trade, colonial empires, partition have all nudged the soundscape, but geographical isolation and strong cultural traditions kept many dialectal identities untouched [26, 13].

Dialects may be close-sounding due to high phonetic overlap can be problematic to distinguish, as clustering studies prove it [13]. These subtelties are valuable for diversity in a culture, but can pose significant challenges for machines. These challenges need to be identified and addressed because dialect-aware ASR and speech technology for serving a language community should be the utmost inclusive. If these technologies do not perform well, mass adoptation by end-users won't be possible, and this would in turn introduce digital divide. This project aims to set the track within Bengali speech technology community in such a way that even if it is one language, everyone with their own dialectal subtleties must be accomodated by our technology.

Chapter 3

Applications of Dialect Identification

3.1 Automated Speech Recognition Systems

There is an increase in the adoption of Automated Speech Recognition (ASR) systems which can distinguish between dialects accurately. Spoken language recognition in these systems now relies on advanced computational models that can more clearly distinguish dialects. They rely on different ways to make them adapt easily and function even better in multilingual situations.

Deep neural networks (DNNs) are an innate and essential block of today's ASR systems. DNNs outperform peer technologies at understanding subtle patterns found in spoken human language. Improvements in performance when handling multiple dialects have been made using techniques including Maximum Likelihood Linear Regression (MLLR) and Maximum A Posteriori (MAP) adaptation. It is now possible for dialect classification systems to take advantage of existing ASR models with the help of transfer learning. You can use it when you have limited labeled data.

An essential aspect of identifying dialects is extracting speech characteristics that truly make a difference. Characteristics such as Mel Frequency Cepstral Coefficients (MFCCs), spectral flux, and entropy are used to detect fine-grained differences in pronunciation [18]. Add prosodic information such as rhythm and intonation to these characteristics and we have a more detailed image. Removing background noise from content before the data goes into the model helps ensure cleaner results [27]. Over the past few years, models based on CRNNs have become more popular. Speaker identification and the discovery of unusual speech pattern are tasks regularly solved by dialect-aware systems in real-world problems like security and forensics. [25].

3.2 Sociolinguistic Analysis

Understanding the links between language, society and culture depends greatly on sociolinguistic analysis. Researchers depend on DID to study the differences in language used in various regions and communities. Because of their diversity and history, Bengali dialects make excellent examples for studying sociolinguistic issues.

Such finding are regularly used in sociolinguistics to explore how language traits are affected by factors such as place, economic activities and political borders. Notably, identifying dialects is important for saving rare and endangered dialects as it helps document their unique aspects before those dialects are wiped out by changes caused by language homogenization or urbanization pressures.

Application of noise reduction in classification models plays an important role in sociolinguistics as speech data picked up in practical uses is commonly affected by ambient noise. By saving the original form of the words, language experts can analyze them without errors. The Mel Frequency Cepstral Coefficient (MFCC) parameters have demonstrated success in identifying certain phonetic features needed to separate different dialects [6, 28]. Applying such methods to linguistic studies makes it easier to find links between languages and the societies they exist in.

Furthermore, DID systems also support forensic linguistics by helping distinguish a speaker's identity using regional accents or markers of dialect [14, 2]. Using these understandings, sociolinguists can examine how speaker's identity is attached to the speech of certain languages or look at instances where talking in a certain way proves someone belongs to a particular community. Sociolinguistics and advanced modeling techniques allow researchers to understand better the relationships between language and society.

3.3 Fake Speaker Detection

Fake speaker detection is greatly helped by dialect identification, especially when artificial speech is used to confuse systems or people. Without being able to detect fake speakers, the security of voice systems, forensic work, customer support and secure authentication would be put at risk.

In many cases, detecting fake speakers requires looking at a particular sound and speech characteristics. As Hossain et al. show, careful control of regional dialects in Bangla helps to ensure clear recognition of spoken signals made by speech synthesis systems.

Novel methods for extracting features help enhance the detection of fake speakers. Combining RASTA-processed MFCCs, as explained by Nagaratna et al. and ML models helps to spot

fake voices accurately [29]. The integration of machine learning models further strengthens the ability to detect fake speakers. Sunija et al. showed in their work on Malayalam dialect identification that SVM gives outstanding precision and recall metrics [3]. Pitch alterations which form part of prosodic feature, have proven useful in dividing genuine speech from fake speech broadly recognized as synthetic speech [18].

3.4 Personalized Voice Assistants

Personalized voice assistants are an important example of using dialect identification tools to give users better interaction and a set of experiences tailored for them exactly. Having an automatic way to identify dialects helps these systems better tailor their results to a person's language background [1, 30].

3.5 Speech Therapy and Rehabilitation

DID is very valuable in the domains of speech therapy and rehabilitation. Since multilingual communities have many different dialects. The ability to accurately identify dialects is essential for tailoring speech therapy interventions to the linguistic and phonetic characteristics of individual patients, particularly in multilingual regions or communities with diverse dialectal variations. Bengali Dialect Identification (DID), as explored in this research project, offers promising applications in these fields by leveraging advanced computational models such as Convolutional Recurrent Neural Networks (CRNN).

Speech therapists are able to address dialects and ways of speaking that can play an important role in speech issues, thanks to DID in speech therapy. Even in challenging settings, CRNN models are able to identify dialectal patterns correctly. The reason this quality matters is that it allows doctors to tell apart disorders like dysarthria or apraxia of speech from ordinary speech problems.

Chapter 4

Dataset Development for Dialect Identification

4.1 Requirements for Dialect Corpora

The development of dialect speech corpora requires utmost care, particularly to those specific criteria that make the corpora useful for applications involving dialect identification. For making robust AI models, it should include elements that introduce diversity, such as prosody, phonetics, and acoustics. Exclusion of spontaneous speech reduces the efficiency of any model. When we speak without a script, our pronunciation varies more, as opposed to a speech that is read straight from a page. According to Korean dialectology, natural speech recordings help show actual patterns in speech. [8].

Varying speech duration and segmentation variability are essential in building dialect corpora. Handling different lengths of audio data is vital so that each set is uniform throughout the dataset. Very short recordings may be discarded since they contain very little information. On the other hand, for a lengthy item, segmenting it helps analyse at the sentence level [1, 30]. It ensures the data is distributed evenly and fairly.

In addition, a linguistic corpus should include several dialects to increase its accuracy. The Kannada Dialect Speech Corpus is made up of five major Kannada dialects. This is indicative of the need for expansive data relating to regional languages to support reliability [29, 31]. The corpora for identifying Arabic dialects have included data from several Arabic dialects [32].

Furthermore, the corpus flexibility is a vitally important feature. Adding more samples to the model or more variety to the dialects may improve its performance.

4.2 Why Dataset Development Was Necessary?

Dialects are the fingerprints of a language. The subtle variation in vocabulary, sentence structure, pronunciation, rhythm, and prosodic features is indigenous to specific geographical locations and the lifestyle of the speaker. Communities that move apart due to historical factors or through cultural differences, their modes of expression also start to follow different paths. Thus, one finds hints of the surrounding cultures and regional traditions.

The occupations of people and their belief systems also directly affect their everyday speech. Moreover, every individual's personality, such as the tone of their voice, gender or age, education, etc., affects their speaking style. Bengali provides a handy case study, as it is the first language of over 250 million individuals and another 41 million who speak it as a second language. The language is spoken in different forms across Bangladesh and the eastern areas of India. Each locality contributes its own distinctive features to the overall linguistic system. Each region adds unique features to the greater linguistic structure. Linguists can distinguish these differences through close observation of phonetic variations, pitch differences, and temporal configurations. Unlike more developed languages like English, Arabic, and Chinese, however, the dialectal variations of Bangla have been given too little attention in fields of natural language processing and speech signal processing.

The necessity for the development of a Bengali Dialect Corpus was thus firmly grounded. Our goal was to create tools capable of recognizing a wide variety of regional dialects to form the backbone of an accurate, precise Bengali dialect recognition system. We started the process by requesting five people to choose what they believe are the most unique district-level dialects. Through the use of perceptual analysis, we ascertained that there are twenty distinctive Bengali dialects. To have speaker independence in our data, we constructed samples from only publicly accessible online sources and purposely avoided personal traits such as gender and age, thereby enabling the system to concentrate entirely on dialectal variation. Although obtaining sufficient recordings for each dialect was rather difficult, it also spurred innovative approaches to fixing data deficiencies. The result is an evenly balanced, diverse and well-built resource that we believe will be valuable for anyone working on Bengali audio and speech technology.

4.2.1 Challenges in Creating the Dataset

Making datasets for Bengali dialect identification is a difficult process. Such issues are the result of difficulties from technical and linguistic, to logistical matters. The first challenge is to gather a lot of good audio files that can be used effectively which represent the twenty different Bengali spoken dialects.

The challenges faced in creating datasets for Bengali dialect identification are complicated and multidimensional. They emanate from a range of technical, linguistic, and logistical issues. The

first challenge is gathering a complete and extensive set of audio recordings that can effectively represent the 20 Bengali dialects. It is still challenging to gather a good variety of recordings for all dialects. It is necessary to carefully plan and carry out each step to capture the linguistic variation and subtleties unique to each dialect. The task requires significant effort in identifying native speakers who can represent their dialects authentically.

Another major challenge is related to data accessibility. Offline data collection processes, either field collection methods or crowdsourcing, were both time-consuming and costly. Therefore, we conducted data collection from multiple online sources such as Facebook, YouTube, and TikTok. They are most commonly used in Bangladesh. There was also a problem with data availability. Even though we found data easily in some regions, we observed that not enough data was available for specific dialects.

A big challenge is to make sure all the recordings have the same level of sound quality. Problems with recording conditions are often caused by background noise or due to the microphone used. With different settings, you might find that the results are less accurate. The use of methods like noise reduction is instrumental in combating such challenges as highlighted by [31]. Even so, using these techniques means more resources are needed and the process becomes more complicated.

The balance of read speech and conversational speech comes with several challenges. Although the spontaneous speech is generally expected to demonstrate a higher level of natural variability that can boost the model's generalizability, it is, nonetheless, prone to reduced intelligibility compared to the read speech examples. Combining both speech types in the data will make the dataset more useful, but it will require careful planning to ensure its compatibility with modern classification systems.

4.3 BengDiDa Dataset

4.3.1 Overview

Dialects

We gathered speech samples from 20 distinct Bengali dialects spanning all of Bangladesh and parts of India. Below Table 4.1 is the full list:

Table 4.1: List of dialects included in the dataset

Dialect	**Dialect**	**Dialect**	**Dialect**
Bagerhat	Barisal	Bogura	Chittagong
Comilla	Old Dhaka	Dinajpur	Jessore
Khulna	Kolkata	Kushtia	Manikganj
Mymensingh	Noakhali	Rajshahi	Rangpur
Sathkhira	Sirajgonj	Standard Bengali	Sylhet

Directory Layout

The dataset is split into three top-level directories, each containing *20 subdirectories* (one per dialect), as shown in Figure 4.1:

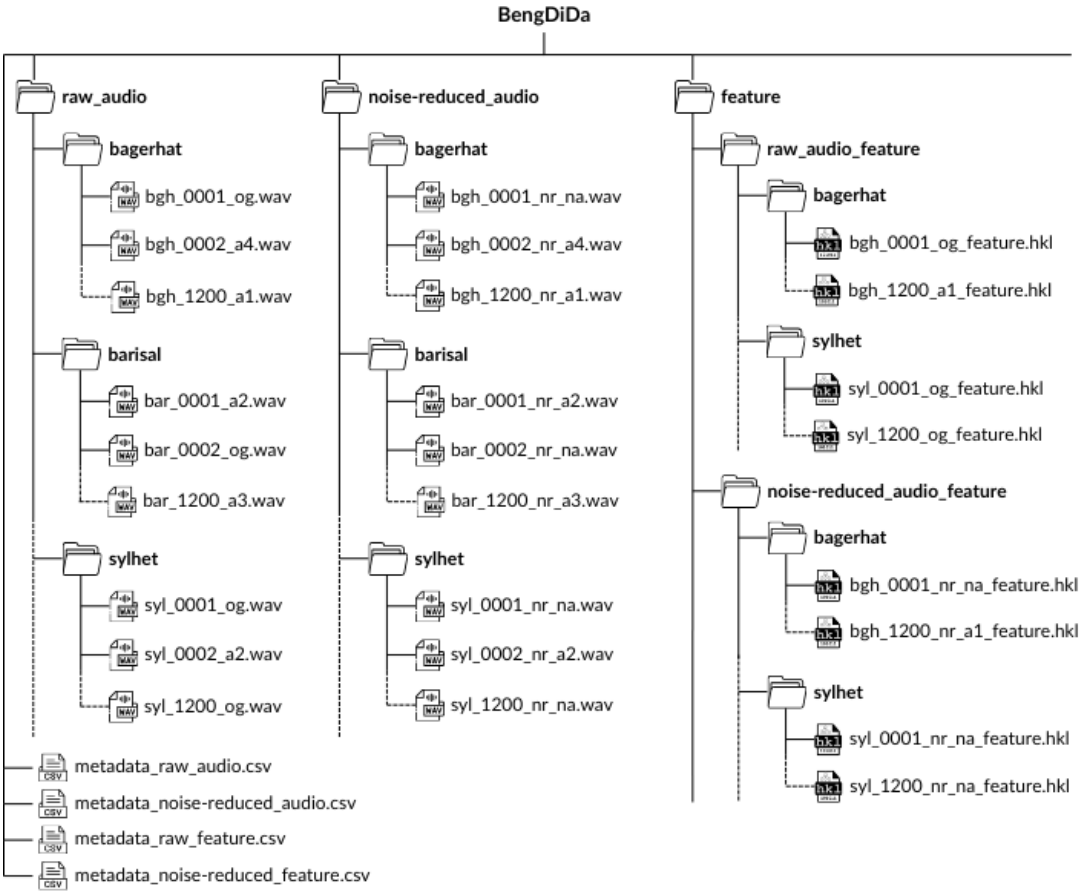


Figure 4.1: Proposed dataset structure illustration

1. `raw_audio`
    - Original 10-second WAV clips, straight from YouTube, Facebook, or TikTok.
2. `noise-reduced_audio`
    - The same clips after applying our noise-filtering pipeline.

3. `features`

    - Pre-computed audio descriptors (Hickle files) for every clip in both raw and noise-reduced form.

Within `raw_audio` and `noise-reduced_audio`, as Figure 4.2 shows, you'll find exactly 1,200 WAV files per dialect (a mix of original recordings and augmented copies). In total, each of these two folders holds **24,000** audio segments:

- **Duration per file:** 10 seconds
- **Total duration per folder:** 66.67 hours
    - Original recordings: 16,069 files ($\tilde{4}4.64$ h, 66.95 %)
    - Augmented variants: 7,931 files ($\tilde{2}2.03$ h, 33.05 %)
- Each folder includes a CSV with file names, dialect labels, processing flags, and metadata.

Mirroring the audio folders, `features` contains:

- `raw_audio_feature`
- `noise-reduced_audio_feature`

Each dialect subfolder holds 1,200 Hickle (`.hkl`) files with cepstral, statistical, frequency, and temporal descriptors extracted from its corresponding WAV clip.

Audio Specifications

Table 4.2: Audio specifications

Attribute	Value
Sample rate	22,050 Hz
Bit depth	16-bit PCM
Bit rate	352.8 kbps
Channels	Mono
Container/Extension	WAV

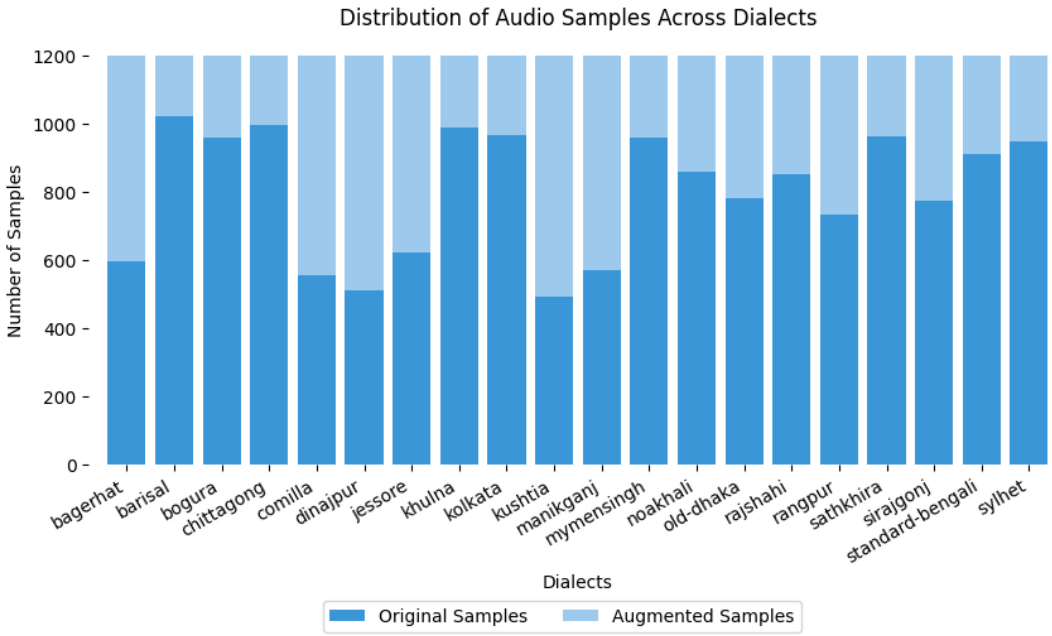


Figure 4.2: Proposed dataset structure illustration

Feature Specifications

Table 4.3: Feature file specifications

Property	Details
Number of feature files	24,000 (1,200 per dialect × 20 dialects)
Feature types	Cepstral, Statistical, Frequency, Temporal
File format	HDF5 via Hickle (`.hkl`)

4.3.2 Dataset Building

Creating a speech corpus from the ground up—particularly for a resource-scarce language such as Bengali—requires a carefully planned, modular workflow. Drawing on established practices in the speech-technology community [33], we divided our process into six broad stages (see Figure 4.3).

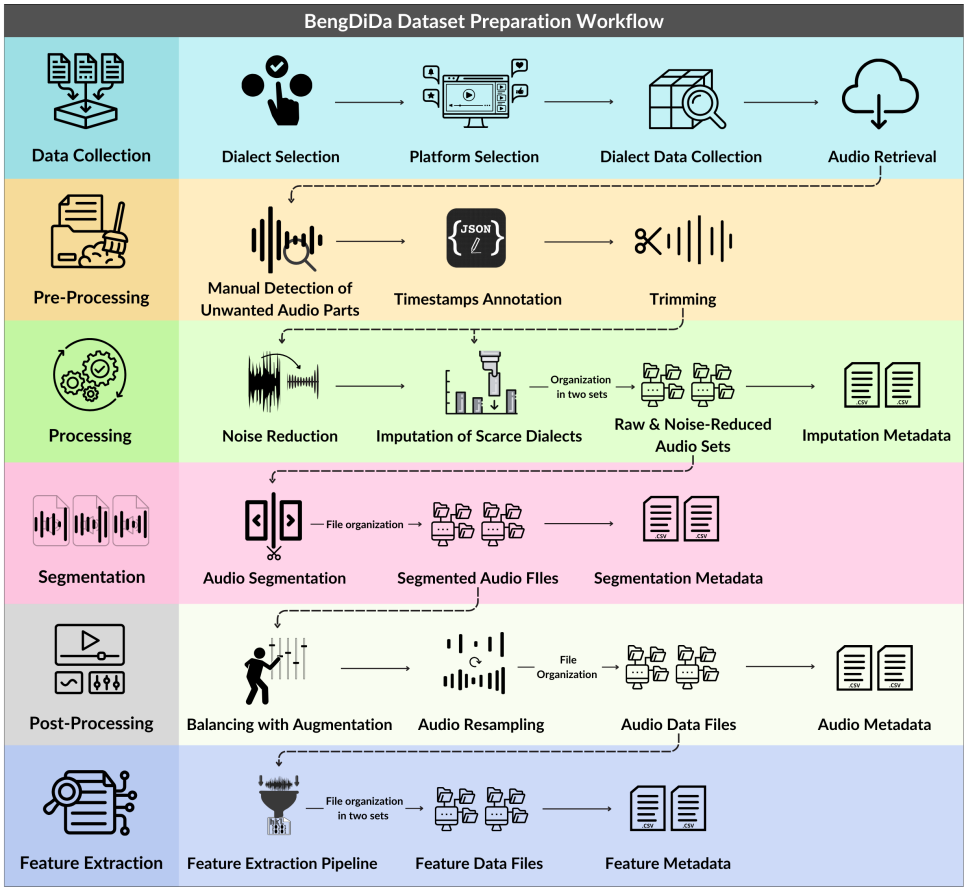


Figure 4.3: Six-stage pipeline for building the BengDiDa corpus.

Data Collection

The first phase, Data Collection, began with selecting which Bengali varieties to include. Given that Bangladesh alone comprises 64 administrative districts—each with its own subtle speech patterns—we chose a manageable subset of twenty dialects based on perceptual distinctiveness among untrained listeners.

Dialect Selection Five volunteers unfamiliar with the project were asked to listen to a broad pool of samples and mark any two clips they felt sounded "the same dialect." We then retained those varieties that yielded consistently high agreement across listeners. This yielded the following twenty dialects: Bagerhat, Barisal, Bogura, Chittagong, Comilla, Old Dhaka, Dinajpur, Jessore, Khulna, Kolkata, Kushtia, Manikganj, Mymensingh, Noakhali, Rajshahi, Rangpur, Sathkhira, Sirajgonj, Standard Bengali, Sylhet. Figure 4.4 provides a visual representation of the geographic regions that we covered.

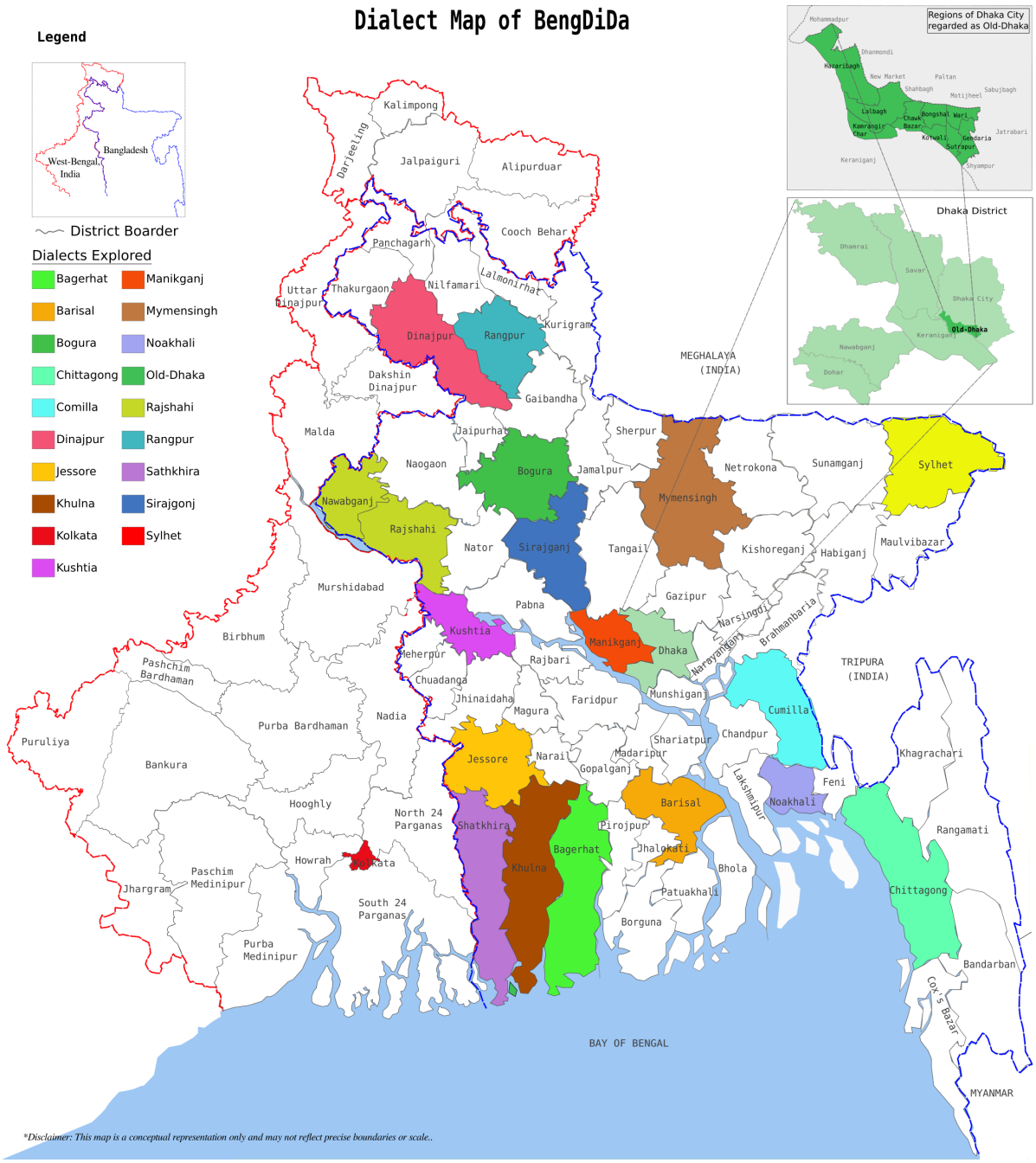


Figure 4.4: Geographic regions of the 20 selected Bengali dialects.

Platform Selection and Data Collection Delimiting the list of our twenty target dialects led us to begin collecting samples from the most famous social media apps in Bangladesh: Facebook, YouTube and TikTok. We developed a large group of search terms in Bengali and English, then checked the findings by trial searches, adjusting them until both languages produced a good range of results. In each platform's search outputs and suggested channels, we manually vetted creators and clips, applying strict criteria—reviewing titles, descriptions, comments, and, when necessary, repeated listening—to confirm that every snippet genuinely reflected its claimed regional variety. Approved items were organized into per-dialect playlists. In total, we

hand-selected approximately 650 distinct items, ranging from 13 s to 1 h 26 min of spontaneous or conversational speech.

Audio Retrieval and Organization With playlists in hand, we automated the download process using the `yt-dlp`. We extracted only 48 kHz, 16-bit, mono audio streams to guarantee consistency across recordings. Each file was then saved into its dialect's folder, and core metadata (source URL, duration, dialect label, etc.) was logged in a master CSV file. This script not only accelerates future updates but also ensures that every clip is traceable back to its origin and clearly annotated for downstream processing.

Preprocessing

Before any analysis, all recordings were passed through a cleanup pipeline to isolate only the dialectal speech.

Trim Any segment containing silence, music, ambient noise, or off-dialect speech was marked as irrelevant. Each file was listened to by annotators who logged start and end times of unwanted material in a JSON manifest.

```
{
  "filename.wav": [
    { "start": 0.00, "end": 2.13 },
    { "start": 15.40, "end": 18.75 },
    ...
  ],
  ...
}
```

Using the timestamp JSON, a script excised the marked intervals and stitched together the remaining speech. Post-trimming, clip durations ranged from 10 s up to 36.3 min, depending on the speaker and source.

Data Augmentation

To boost model robustness and balance dialect sample counts, we applied slight audio perturbations using the `Audiomentations` library. Four augmentation "recipes" were defined; each recipe combines several transforms in sequence or at random. A summary follows:

Series A1 (a1)

- Random time–shift (±40% of clip length)
- Speed adjustment (factor 0.8–1.2)
- Bit depth reduction (12–14 bits)
- 7-band parametric EQ (±12dB per band)

Series A2 (a2)

- Random time–shift (±30%)
- Simulated room reverb (absorption 0.08–0.30, RT60 0.3–1.3s)
- Additive Gaussian noise (amp 0.002–0.008)

Series A3 (a3)

- Speed adjustment (0.8–1.2)
- Simulated reverb (70% chance; same params as A2)
- Random time–shift (±15%)
- One random filter:
  - High-pass (cutoff 80–180Hz, roll-off 12–24dB/oct)
  - Low-pass (cutoff 3–3.8kHz, roll-off 12–24dB/oct)

Series A4 (a4)

- Time masking (bandwidth 5–8% of clip)
- Random time–shift (-50% to +30%)
- Pitch shift (±2 semitones)
- One random distortion:
  - Tanh distortion (factor 0.15–0.25)
  - Clipping distortion (threshold 5–15%)
- 7-band parametric EQ (±14dB per band)

Processing

In this stage, we applied advanced signal-processing and data-augmentation techniques to produce two parallel corpora (raw and noise-reduced) and to address the limited sample counts in certain dialects.

Noise Suppression To create a noise-reduced counterpart for each recording, we employed the `DeepFilterNet3` model [34]. This neural network applies deep filtering to attenuate ambient noise while preserving speech integrity. After processing, all cleaned files were saved under `noise-reduced_audio` with the suffix `_nr` in their filenames. Figure 4.5 compares a representative utterance before and after filtering, showing both waveform overlays and spectrograms.

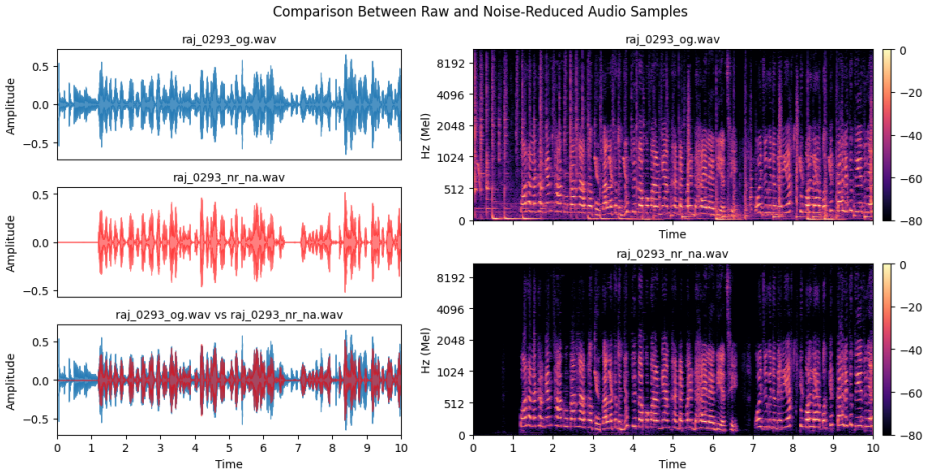


Figure 4.5: Left: raw (top) vs. noise-reduced (middle) waveforms, superimposed raw and noise-reduced (bottom). Right: corresponding spectrograms highlighting noise removal.

We quantitatively assessed noise suppression using non-intrusive metrics `DNSMOS` [35] and `NISQA` [36]. Figure 4.6 plots the distributions of mean opinion scores (MOS) for raw and denoised samples across background noise and speech quality dimensions. The right-skewed plot of noise-reduced speech shows higher background-noise scores, confirming effective noise suppression. A similar score for signal and discontinuity confirms effective noise removal without degrading intelligibility.

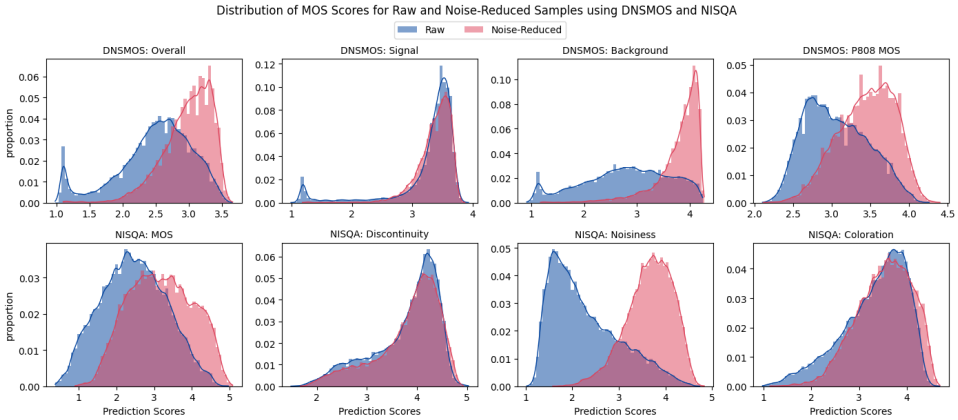


Figure 4.6: MOS distributions for raw vs. noise-reduced audio: (top) DNSMOS, (bottom) NISQA.

Imputation for Low-Resource Dialects Certain dialects lacked sufficient audio due to limited online availability. To standardize class durations, we applied an imputation scheme that generates entirely new utterances, each differing in content from all originals, by leveraging existing recordings.

Let T_d be the total duration (in seconds) of dialect d. We computed the median duration across all classes:

$$T_{\text{target}} = \text{median}\{T_d : d = 1, \ldots, 20\} \approx 8600 \text{ s}.$$

Any dialect with $T_d < T_{\text{target}}$ was marked for imputation. The required percentage of additional material is given by:

$$\text{Impute}\%_d \;=\; \frac{T_{\text{target}} - T_d}{T_d} \;\times 100.$$

For each low-resource dialect:

1. We selected a random contiguous segment from an original clip.
2. Applied the `augment series (a1)` recipe to this segment.
3. If the chosen start time lies too close to the end, we wrapped the selection around to the beginning—treating the audio as a continuous loop—to ensure seamless coverage.
4. Concatenated the augmented fragment back into the original at a new random insertion point.

By repeating this until the imputation percentage $\text{Impute}\%_d$ was met, we expanded each dialect's corpus without ever duplicating speech content.

All clips were saved alongside originals in the dialect's folder, using the same naming conventions (with an augmentation flag). A CSV file records:

- Original filename and duration,
- Imputation flag and percentage applied,
- Time indices of each augmented segment.

Figure 4.7 illustrates the imputation workflow for a sample dialect.

Segmentation

Longer recordings were split into fixed-length clips to standardize inputs for feature extraction and modeling.

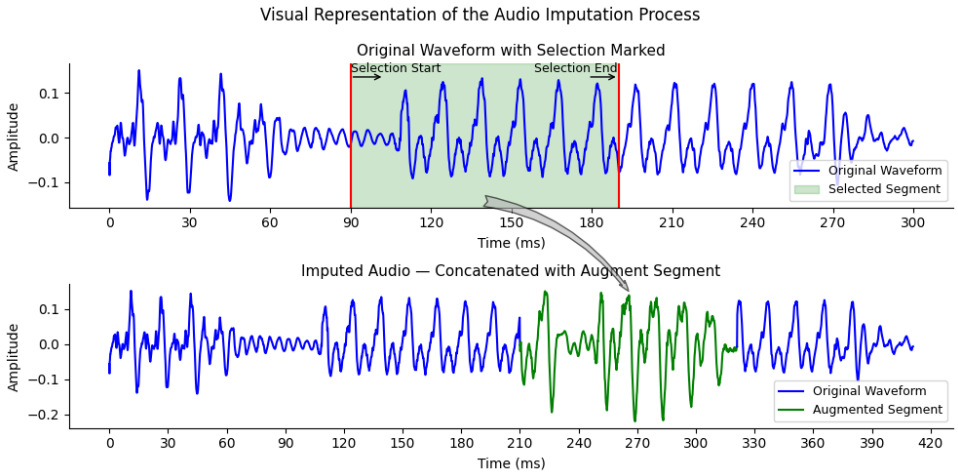


Figure 4.7: Visualization of the imputation process for low-resource dialects, showing segment selection, augmentation, and reinsertion.

- Each processed file—whether raw, imputed, or denoised—was chopped into non-overlapping 10s segments.
- Segments were saved as WAV: 48kHz sample rate, 16-bit PCM, mono channel.
- A segment is labeled "augmented" if it contains any portion of augmented audio; otherwise, it remains "original."

Post-processing

After segmentation, we applied two final steps—class balancing via augmentation and resampling to ensure uniformity across all 20 dialects.

Class Balancing Although low-resource dialects received imputation, high-resource dialects remained largely unaltered. To introduce consistent variability, we applied extra rounds of `augmentation series (a2, a3 a4)`, randomly to at least 10% of each dialect's "original" segments. Augmentation continued until every dialect reached the target of 1,200 total segments, blending new variants evenly into each class, also increasing the overall quantity of the dataset. Figure 4.8 illustrates waveform overlays of three original clips against their augmented counterparts.

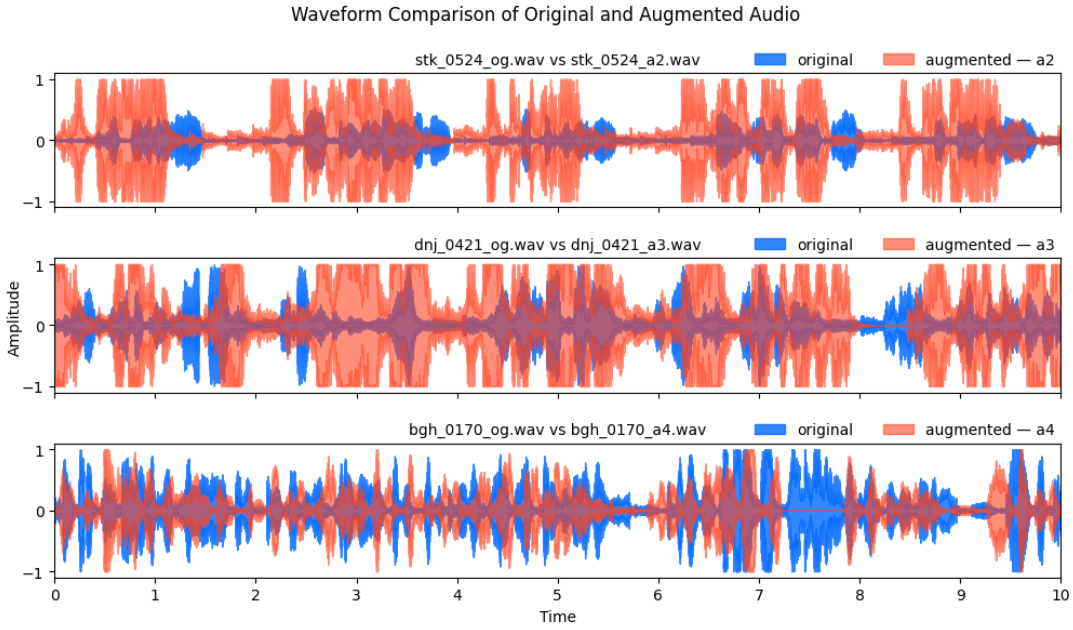


Figure 4.8: Original vs. augmented waveforms.

Resampling and Renaming All 10s segments were downsampled to 22.05kHz from 48kHz (16-bit, mono) to maintain a compatible storage management.

Figure 4.9 illustrates the impact of audio augmentation using self- and cross-similarity matrices. The self-similarity matrix (left) shows strong diagonal patterns, indicating structural consistency within the original audio file. In contrast, the cross-similarity matrix (right) compares the original with an augmented version and reveals reduced structural alignment, suggesting significant acoustic transformation due to augmentation.

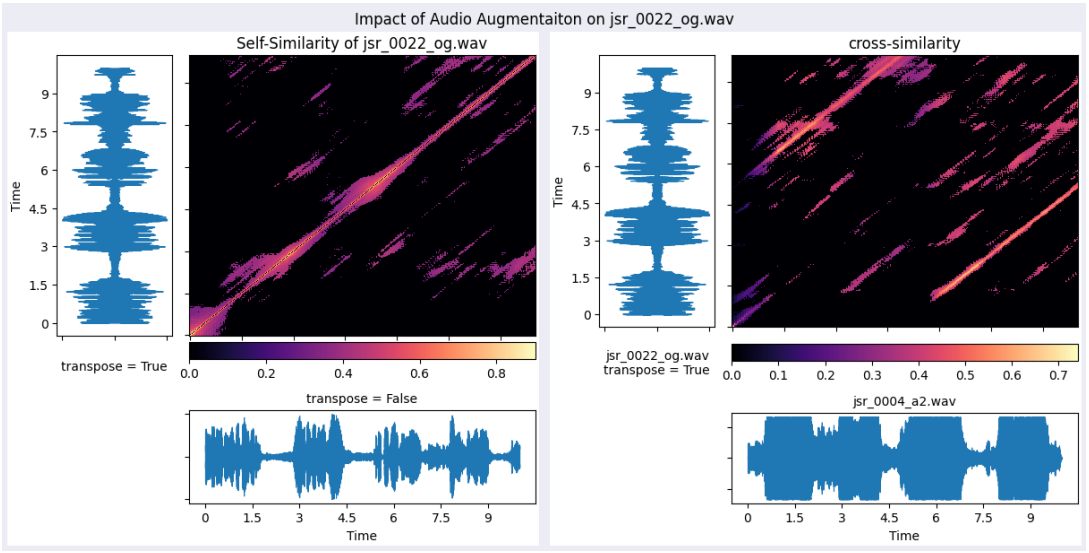


Figure 4.9: Impact of augmentation on generating new variations of the original sample.

Feature Extraction

Transforming raw audio into a set of informative descriptors is the foundation of any speech classification pipeline. In our corpus, each 10s segment is converted into cepstral, spectral, temporal, and functional features that together capture the fine-grained acoustic cues needed to tell dialects apart. These feature vectors serve classical machine-learning algorithms (e.g. random forests, SVMs, or ensemble learners), and can also be arranged into multi-dimensional arrays for convolutional or recurrent neural networks.

Time–frequency representations such as mel frequency cepstral coefficient (MFCC) naturally form (coeffs $\times$ frames) matrices. Instead of flattening them into long vectors, we keep their native 2D layout to maintain frame-wise essential features for CNNs and sequence models that exploit local patterns in time and frequency simultaneously. This design choice follows best practices in speaker and dialect recognition, where preserving temporal locality has proven beneficial.

Choosing the right audio descriptors is pivotal for capturing the nuanced acoustic patterns that distinguish Bengali dialects. To inform our selection, we performed a thorough review of the speech-processing literature, particularly studies on accent detection and dialect identification, which consistently highlight the benefits of integrating cepstral, spectral, temporal, and phonatory/articulatory features. By synthesising these best practices, our feature suite balances perceptual relevance with computational efficiency, making it suitable for both traditional machine-learning pipelines and modern deep-learning architectures. Figure 4.10 provides a visual of the transformation of audio into the feature-based representation.

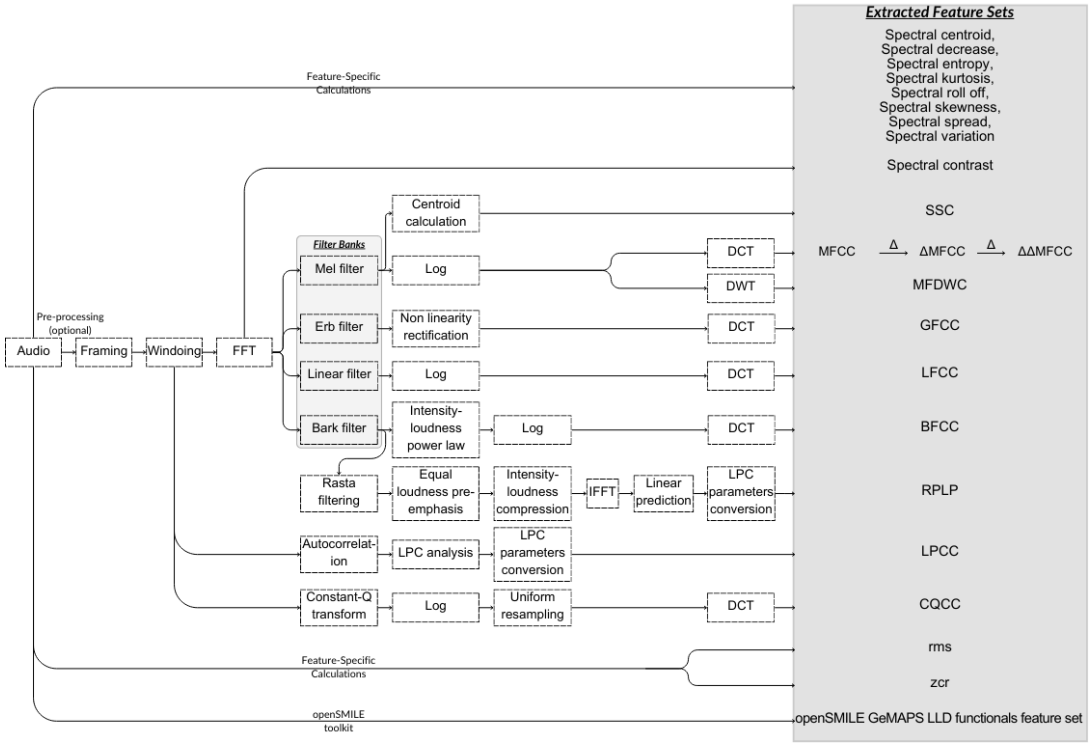


Figure 4.10: Feature extraction pipeline.

Feature categories:

- **Cepstral:** Mel-Frequency Cepstral Coefficients (MFCCs), Mel-Frequency Discrete Wavelet Coefficients (MFDWC), Bark-scale (BFCC), Gammatone-scale (GFCC), Linear-freq (LFCC), LPCC, CQCC and RPLP—all extracted via `librosa` [37] and `spafe` [38].
- **Spectral:** Centroid, bandwidth, skewness, contrast and Spectral Subband Centroid (SSC), computed with `TSFEL` [39], characterizing timbre and formant distributions.
- **Temporal:** Zero-Crossing Rate (ZCR) and Root-Mean-Square energy (RMS), capturing rhythm and loudness.
- **Functional:** 66 low-level descriptors from the GeMAPS [40] minimal set (jitter, shimmer, formants, etc.) via `OpenSMILE` [41], reflecting phonatory and articulatory traits.

Storage and organization All features are stored in Hickle (`.hkl`) files—an HDF5-based container offering fast I/O, compression, and support for mixed data types. The directory layout mirrors the audio folders, ensuring each segment's features are immediately locatable. Within each Hickle file, features live in a dictionary of key–value pairs; Table 4.4 illustrates the structure.

Table 4.4: Complete list of features stored in each Hickle file

Key	Shape	Description
zcr	scalar	Zero-Crossing Rate: rate of sign changes in the waveform.
rms	scalar	Root-Mean-Square energy: frame-level loudness.
spectral_contrast	(7, 512)	Difference between spectral peaks and valleys across 7 bands.
spectral_statistics	(8,)	Eight statistical descriptors (centroid, bandwidth, skewness, flatness, flux, roll-off, etc.).
mfcc	(24, 512)	Mel-Frequency Cepstral Coefficients: spectral envelope representation.
mfcc_delta	(24, 512)	First derivative of MFCCs: captures temporal dynamics.
mfcc_delta2	(24, 512)	Second derivative of MFCCs: captures acceleration of spectral changes.
bfcc	(24, 512)	Bark-Frequency Cepstral Coefficients: perceptual Bark-scale features.
gfcc	(24, 512)	Gammatone-Frequency Cepstral Coefficients: auditory-inspired filterbank features.
lfcc	(24, 512)	Linear-Frequency Cepstral Coefficients: uniformly spaced filterbank features.
lpcc	(24, 512)	Linear Predictive Cepstral Coefficients: LPC-based spectral envelope.
cqcc	(24, 512)	Constant-Q Cepstral Coefficients: pitch-sensitive spectral features.
mfdwc	(53,512)	Mel-Frequency Discrete Wavelet Coefficients: wavelet-based time–frequency features.
ssc	(24, 512)	Spectral Subband Centroids: center of energy in each of 24 subbands.
opensmile_lld_functionals	(66,)	Functional summaries (mean, std, percentiles, slopes) of 66 low-level descriptors (GeMAPS).

The inclusion of raw audio and its extracted descriptors allows user to choose either end-to-end neural approaches or conventional feature-based modeling approaches, or even a blend of both approaches depending on their particular study goals. Such flexibility ensures that the BengDiDa set is capable of supporting a wide variety of dialect identification methods.

4.3.3 Dataset Accessibility

`BengDiDa` soon will be publicly released on Zenodo. The dataset will be the first large-scale and openly available Bengali dialect speech repository. The well-structured framework of the dataset, combined with standardized preprocessing approaches and support for broadly accepted machine learning toolboxes, enables easy integration and reproducibility in dialect-related research.

4.3.4 Ethical Considerations in Data Collection

Consent and Privacy The BengDiDa speech corpus was exclusively collected from publicly accessible sources on social media platforms such as YouTube, Facebook, and TikTok. Utilizing data from these platforms for non-commercial academic purposes adheres to fair use regulations and does not infringe upon copyright laws. All audio samples underwent a comprehensive processing protocol that included segmentation, augmentation, randomization, and various other alterations. This procedure guarantees speaker anonymity and maintains research ethics along with privacy standards.

Chapter 5

Systematic Literature Review on Dialect Identification

5.1 Methodology for Literature Review

5.1.1 PRISMA-2020 Framework Overview

PRISMA framework issues standardized guidelines for how Systematic Literature Review (SLR) on a given topic is carried out in a reliable and transparent way. It lets other researchers see the methods used to select articles and appraise the findings. It helps pinpoint the areas of research, in our case DID in Bengali, that are not being thoroughly explored. Especially gaps like implementing CRNN models for Bengali DID research have been observed.

Researchers start the process by defining their questions and searching using preset keywords such as "Bengali dialects" and "CRNN". Commonly, DID studies that use LID point out how acoustic and spectral characteristics play a role [18]. Investigations of Arabic dialects draw attention to problems faced during research with limited resources, much like how Bengali DID has been examined[26, 42]. MFCC, RASTA and PLP feature sets for speech tasks have demonstrated promising results and might help Bengali DID [43].

The classification scores increase with models such as X-vectors and systems using phonotactic fusion [44]. Strong results have been produced by CRNNs, but their use for Bengali DID or classification tasks have been rare [21]. This symmetric KL divergence method uses clever approaches for comparing features that are significant for dialect modeling [45]. Adhering to PRISMA standards ensures comprehensive searching and finding of related literature that the DID researchers can replicat with high standards. [26, 46, 18, 43].

5.1.2 Selection Criteria

Only the studies covering DID that were of high quality and related to Bengali were given priority, especially when the researchers used CRNNs. Researchers focused mostly on studies using unique feature extraction approaches, including dynamic formant frequencies and using spectrograms for recognizing tones in Kannada and Chinese dialects. [10, 47, 48].

Reliable results came from studies involving real speakers and data that included noise reduction, as they proved to be more accurate [1, 30, 29, 31]. Since Bengali DID is a diverse field. Systems involving various types of SVMs, XGB classifiers and LSTM-HMM hybrids were given considerable attention [15, 5].

Both studies focusing on multilingualism and software localization and studies assessing spoken LID systems, advised that systems should account for differences in dialect [19]. Reviews of spoken LID systems reinforced the central role of spectral and acoustic features in CRNN architectures [21]. This type of selection aids in improving Bengali DID practices.

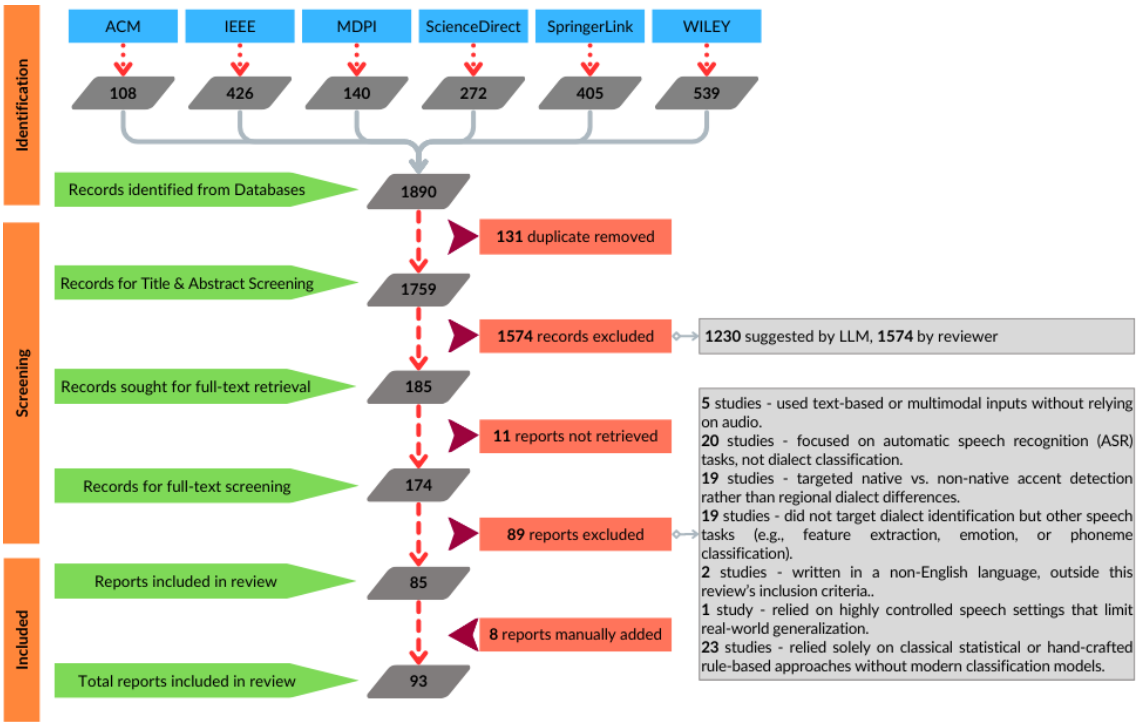


Figure 5.1: PRISMA protocol study selection process.

5.1.3 Data Extraction Process

It is crucial to extract data when making robust DID systems for Bengali. Researchers are to gather and arrange information about datasets, features and models. Feature extraction, preprocessing and preparing the dataset are common steps.

Normalized Kaldi and Fbank features are stressed as ways to increase reliability [49]. Carefully removing noise and pauses from speech segments makes the data easier to understand [10]. Large collections of information from public archives and YouTube have solved problems with limited resources in LID research [50], and studio recordings cut down on ambient or background noise [27]. Using features from several acoustic sources improves classification and I-Vector technology with GMMs helps make the data easier to handle by reducing dimensions drastically [11, 51]. Doing data collection based on regions, as in the case of Kannada studies, helps represent the many varieties of the language [52]. All of the strategies together educate the process of creating Bengali DID datasets.

5.2 State of Research in Dialect Identification

5.2.1 Existing Methods for DID

Thanks to machine learning and deep neural networks, DID methods have become more advanced. Combining convolutional and recurrent layers in CRNNs allows for strong results in noisy conditions by utilizing processed MFCCs. [13]. LSTM and GRU models manage the processing of sequential information well, but deep speech representation is yet to be widely studied [53].

The technique of sharing only some parameters greatly reduces the number of network weights and gives the network the ability to be flexible in various tasks. [54]. The use of MAP and KLD in UBMs makes it easier to tell apart dialects that have similar features [45]. Most of the work still depends on feature engineering and many use acoustic-phonetic and phonotactic strategies, yet classifying same-sounding words is tough [18, 2].

Using standardization, for example adding silent audio, helps when training the model [55]. Chinese Deep Invertible Dense Networks have shown that iterative learning can contribute to improved results on dynamic datasets [23]. More work in Bengali DID requires that the datasets are supported by models that are acclimated to noise and change.

5.2.2 Challenges in DID Research

Research on DID has major obstacles caused by constrained datasets. Diversity in dialects and technological limitations are also to be accounted for. Difficulty in advanced modelling is faced since there are no comprehensive speech corpora for the likes of under-represented languages like Malayalam and Bengali [27]. Regional mixture imposes challenges in modelling under-analyzed dialects like SCB [56]. Often, spectral and phonotactic methods have difficulties with being flexible and consistent in extracting the prosodic features of different dialects [26].

Experimenting on phone mixture analysis indicate that it does not help classification much, which shows the necessity for distinguishable feature separation [45]. According to research, MLPs perform reliably, but they require a uniform dataset [17]. Indian languages from the Indo-Aryan and Dravidian families differ so much that it becomes important to use language-specific methods [10]. Although Res-BLSTM with i-vector models help improve results, they require both large data and a large amount of computing power [57, 58]. Minimizing noise and adapting features is still necessary but often a difficult task. It is important to design solutions in the future that are both effective and accurate for many languages at the same time.

5.2.3 Gaps in Bengali DID Studies

When it comes to DID, Bengali is much less studied than Arabic, Chinese or Korean. An important obstacle is that there are no comprehensive dialect audio datasets available. Alternatively, other languages make good use of hand-picked data that allows for sturdy feature extraction and modeling [53, 2].

Although CNNs, LSTMs and GMM-MFCC models have worked well for other languages, usage in Bengali DID tasks is still limited. [6, 21]. Diversity in Bengali dialects imposes a level of modeling problems [59]. While we have seen that vowel formants and prosodic aspects help in other languages, these are mostly unexplored in Bengali DID [60, 56, 26, 61].

Furthermore, glottal signal analysis, designed for handling these troublesome data, has not seen much use yet [46]. Closing these gaps requires using BengDiDa and adding CRNNs, prosodic features and innovative ways to handle noise in future studies.

5.3 Comparative Analysis of DID Techniques

5.3.1 Machine Learning vs Deep Learning Approaches

Both ML and DL demonstrate strong abilities when identifying dialects. Works using Support Vector Machines (SVMs) together with i-vector features like MFCC and LPCC have accomplished reliable results in several experiments where understanding and detailed feature engineering play key roles [62, 63].

Alternatively, LSTMs, DCNNs and BiLSTMs with attention are able to automatically discover important hierarchies in data to capture temporal features within acoustic events [64, 65]. High accuracy in VarDial has been demonstrated by models that use a combination of ML and DL methods like Kernel Discriminant Analysis (KDA) [57]. While noise reduction, Mel spectrograms and feature scaling are beneficial to both approaches, [22], there are still problems in

under-resourced languages such as Bengali [66, 67]. Using transformers and enriched BengDiDa data could be useful for further studies.

5.3.2 Hybrid Models in DID

The combination CNNs and RNNs in CRNN models leads to their usefulness in large and diverse corpus like BengDiDa as they address both spatial and temporal aspects. [68, 2]. Using noise reduction and feature adaptation as preprocessing can make these models generalize and adapt well on unseen data. [7].

According to researchers, CRNNs work better and are more efficient than some ensemble methods [10, 68]. However, these models should be optimized and checked on speaker-independent datasets to prevent overfitting [69, 70]. Future studies can experiment with merging attention mechanisms and transformers to enhance understanding of local accents and SIDEKIT automates many of the stages necessary before training [70]. Bengali DID is heavily dependent on hybrid approaches that ensure large-scale and accurate systems.

5.3.3 Role of Pretrained Models

Bengali DID is a challenging task due to it's under-resourcedness and dialectal diversity. Pretrained models are a way to address and mitigate this issue. Transfer learning from pretrained CNNs and RNNs can relate commonalities in features from audio and they can be tuned on BengDiDa. This helps keep the cost of computation low all while accuracy is raised [71, 72].

On the first layer, CRNNs use pretrained filters to gather spectral features. On subsequent layers, they use recurrent structures to uncover relations in the temporal domain. Adjusting and fine-tuning these layers helps models pick up Bengali phonetic and prosodic features [17, 72]. Getting rid of noise and silence makes the pretraining sample alignment better and increases the network's robustness [45].

By applying attention or different loss functions, attention to dialect can be increased, especially for intonation [2]. Even though pretrained models may have biases from the original data, adding them to GRU models for training may lead to better results in future tasks [71]. Although many studies on Arabic and Korean DID exist [26, 61, 2], the use of these methods in Bengali is rare, indicating a promising subject for further study.

Chapter 6

Convolutional Recurrent Neural Networks (CRNN) for DID

6.1 Introduction to CRNN Architecture

A Convolutional Recurrent Neural Network (CRNN) is a hybrid deep learning model that combines convolutional layers (CNN) and recurrent layers (RNN) in sequence. In a CRNN, the CNN layers act as a front-end to extract high-level spatial features from the input (such as a spectrogram), and the RNN layers act as a back-end to model temporal or sequential patterns over those features. This hybrid architecture thus combines a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN) to process sequential data. In the context of speech processing and dialect identification (DID), a CRNN can first use convolutional filters to learn local spectro-temporal patterns (for example formant or energy structures in short time windows) and then use recurrent units to learn how those patterns evolve [73]. This makes CRNNs especially suitable for DID, since spoken dialects differ in both local acoustic cues (e.g., phoneme pronunciations) and global sequence-level patterns (e.g., prosody, intonation over the utterance). In fact, recent research on spoken language and dialect identification has shown that a CNN+LSTM (a type of RNN) architecture can effectively capture both short-term acoustic features and long-term dependencies [73, 74]. For example, [74] used a CNN front-end with an LSTM back-end to perform Arabic dialect identification and found that this CLSTM architecture combined the advantages of convolutional feature extraction and recurrent temporal modelling. All in all, CRNNs apply CNNs to discover spatial patterns in speech and then use RNNs to combine those patterns one after another which makes them suitable for dialect identification since localized speech analysis and sequential modelling are both needed for this task.

6.2 Convolutional Layers for Spatial Features

The convolutional layers in a CRNN handle the input spectrogram or other time-frequency image to find local features. A spectrogram displays speech in two dimensions, with time on one side and frequency on the other and CNNs view it as an image. Convolutional filters are moved over it to identify patterns in the spectrogram. The main benefit is that CNNs use local spectro-temporal locality: they learn small filters that react to certain time-frequency shapes and these filters stay the same no matter where they appear on the spectrogram. As a result, a CNN can spot phonetic details and short-term sounds at any point in time or frequency. [75] describes that CNNs allow for finding local features in a spectrogram by sliding filters over small image sections. To explain, a CNN takes features from the spectrogram and learns to show things like formant trajectories or consonant bursts that are informative for distinguishing dialects. [76].

It is very important to use local feature extraction in dialect identification because there are only small acoustic variations in specific sounds when different dialects of Bengali are spoken. As an example, small pockets of Bengali communities may change their consonant aspiration, vowel quality or intonation within a short time span. Because of their localised receptive fields, CNNs can spot these specific acoustic details. The use of weight-sharing convolutional kernels in CNNs helps them find distinctive sounds even when these occur slightly ahead of or behind their expected moments. Because the model is invariant in time and frequency, it focuses on which features appear rather than when or where they appear. Experts report that dialectal speech includes different sounds and sound combinations, along with specific ways of pronouncing words in each dialect [77]. Using local filters, CNNs can identify both the resonance of vowels and the sounds made by consonants, which may be pronounced differently in different dialects [77]. Overall, convolutional layers in a CRNN learn different filters across the speech spectrogram which allows them to automatically discover features such as formants, spectral shapes and time-local dynamics (onset offsets, localized energy bursts) that help to distinguish Bengali dialects.

6.3 Recurrent Layers for Temporal Modeling

After the CNN has distilled each short-time frame of audio into a feature vector (often one vector per time frame or per block of frames), the recurrent layers (typically LSTM or GRU units) process the resulting sequence of vectors. The RNN is meant to recognize features that change with the course of the utterance. RNNs have internal state (memory) that evolves as they read each frame in order, so they can remember information from previous time steps and integrate context across time. This is important for dialect identification as a lot of dialectal features are found in the way speech is delivered over several frames. The way someone uses pitch or how they move from one sound to another often reflects their dialect. Unlike convolutional models,

recurrent layers can connect distant frames. In particular, LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) cells are designed to alleviate the vanishing gradient problem and retain information over longer sequences.

To represent Bengali DID, we need sequential modelling to show how sound features change as a word is spoken and how sounds are sequenced within an utterance. The RNN (such as a bidirectional LSTM) takes the output of the CNN at each time step and updates its hidden state. The process allows the network to recognise that hearing an aspirated consonant before a vowel can show which dialect the speaker uses. In reality, CRNN models add BiLSTM layers to examine both previous and upcoming frames which helps them recognize movement over time. As an illustration, [74] point out that a CNN front-end works well at extracting features, while the back-end LSTM network can observe longer-term patterns in the data. This case demonstrates that the CRNN architecture relies on RNNs to deal with the sequence details needed to tell apart dialectal speech recordings.

6.4 Advantages of CRNN

The main benefit of CRNNs is their joint modelling of local spectral features and temporal dynamics. Convolutional layers capture spectro-temporal patterns and recurrent layers capture dependencies over time, so enabling CRNNs to efficiently use both short-term and long-term speech information. Especially on sequential speech tasks, this complete modelling typically beats models using just CNN or just RNN. It lets the model depict elements like formant changes or consonant bursts that help to distinguish dialects.

Translation invariance and weight sharing introduced by the CNN component help the model to be robust to fluctuations in the exact timing or frequency of features. Practically, a CRNN can learn dialect-specific acoustic signals such as formant changes or noise bursts anywhere they occur, so improving robustness to minor misalignments or changes in speech speed.

CNN layers significantly lower the dimensionality of the input sequence by pooling before feeding it to the RNN, thus CRNNs can be more parameter-efficient than completely connected deep recurrent systems. Often resulting in faster convergence and lower overfitting, the convolutional feature compression helps isolate salient features, which the RNN then processes.

CRNNs naturally accept varying-length inputs in sequence. In speech tasks where utterance duration may vary, the ability of the recurrent layers to process sequences of arbitrary length is quite important. This adaptability lets the same model manage both long and short audio clips without retraining.

Previous research repeatedly shows that CRNNs perform rather well on speech and language tasks. In speech separation and enhancement, for instance, a CRNN architecture can efficiently

combine CNN and RNN benefits to raise output quality. Likewise, spoken language identification systems leveraging CRNNs have produced state-of-the-art performance [73]. These achievements imply that the sophisticated patterns in speech, including dialectal variation, are well-suited for the combined spatial-temporal modelling of CRNNs.

Chapter 7

Experimental Configuration

7.1 Hardware and Software Environment

The model training and evaluation were conducted using Google Colaboratory with access to GPU acceleration. The following table outlines the computational setup:

Component	Specification
Hardware	NVIDIA Tesla T4 GPU with 16 GB VRAM
Runtime Environment	Google Colab Notebook
Software Framework	TensorFlow 2.18 (Keras API)
Operating System	Ubuntu-based Colab runtime
Python Version	Python 3.10

Table 7.1: Hardware and software configuration

7.2 Data Preparation

The dataset was loaded and preprocessed using standard techniques. Audio features were already extracted beforehand. We experimented with multi-channel input for our model. First, we experimented with 3-channel (MFCC, GFCC, RPLP), resulting input shape (24x512x3). Then we tried with stacking more feature channels channel-wise. Taking upto 8 features, making the input shape 24x512x8. We also tried to compare noise-reduced and raw audio, and also tried to compare with the pre-trained model, ResNet50. The dataset was split into training (70%), validation (20%), and testing (10%) sets.

Feature Stacking Strategy In this study, several feature types (such as Mel-spectrograms, excitation residuals and prosodic contours) are combined by stacking and form a 3D tensor of

size (time × frequency × channels). If eight feature maps are made for each frame, the input of the next layer becomes (24, 512, 8). Treating each feature as a different colour channel lets the first convolutional layer train filters that can work with all types of features together. Each feature set keeps its particular characteristics, allowing the network to still discover connections between features in deeper parts of the network. In comparison to connecting features one after another along their spatial axis, channel stacking helps the network by not growing its spatial dimensions or time and frequency inputs. Using this approach allowed the CRNN to work with spectral, excitation and prosodic features of the BengDiDa dataset all at once.

7.3 Model Architecture

7.3.1 Custom CRNN

The proposed architecture Figure 7.1 is a Convolutional Recurrent Neural Network (CRNN) designed specifically for dialect classification based on audio spectrograms. All parts of the model serve specific functions that make it possible to extract and interpret both spatial and temporal features present in dialectal speech.

- **Input Layer:** The Input Layer takes in a 3-channel spectrogram (24, 512, x) that includes static and dynamic spectral features. This data maintains both the frequency and time-dependent variations necessary for recognising dialects.
- **Convolutional Layers:** A sequence of four 2D convolutional layers, with 32, 64, 128 and 512 filters. It is used to find local features in the spectrogram using learnable filters. These filters look for different compounds in speech, such as the patterns of formants or transitions in how a word is said, because they tend to be distinct among different dialects.
- **Pooling Layers:** After every convolutional block, max pooling is used to lower the size of the feature maps and keep the most important information. The approach also makes the model invariant across translations and reduces its computational requirements.
- **Normalization Layers:** Normalizing activations during each batch is the role of Batch-Normalization layers which speeds up training and preserves learning consistency. Also, they act like a regularizer to help prevent the model from overfitting.
- **Global Pooling Layer:** GlobalAveragePooling2D takes the whole time-frequency plane and averages it, so the result is a compact 1D feature vector. It doesn't overfit like flattening and still keeps the big picture in mind.
- **Reshape Layer:** Changes the pooled vector into a 3D sequence that is suitable for recurrent layers. Because of this conversion, each frame in the spectrogram is treated as

a sequence of frame-level embeddings.

- **Bidirectional RNN Layers:** Two stacked Bidirectional RNN layers, with 64 and 128 units, are used to handle temporal dependencies in both forward and backward directions. These layers are lightweight yet effective for modeling sequence data, making them well-suited for dialect recognition in variable-length audio segments.
- **Dropout Layers:** Following both the CNN and RNN layers, the dropout layer randomly disconnects neurons during the training phase. This reduces overfitting and improves the generalizability of the model.
- **Dense Softmax Layer:** The output layer here is fully connected with 20 units, each representing a dialect. It uses softmax activation to turn the logits into a probability distribution of dialect classes.

This architecture consists of approximately 1.12 million trainable parameters and supports both BiGRU (Bidirectional Gated Recurrent Unit) and BiLSTM (Bidirectional Long Short-Term Memory) variants.

7.3.2 ResNet50-Based CRNN

Apart from the custom CRNN baseline, we also used a transfer learning method that combines a ResNet50 model as a feature extractor Figure 7.2. ResNet50 is used for spatial feature encoding and afterward, a Bidirectional GRU block is used to model the sequence of the data.

The zero-padding layer is to ensure the image's dimensions are not reduced at the edges, also to match the required height for resnet. The first batch normalization layer ensures that the input is well-regulated. A following channel adapter convolution2D layer rearranges the input into three channels to suit the shape needed by ResNet50. The main part of the model is ResNet50, trained upfront on the vast ImageNet [78] dataset, which delivers a dense map of 2048-dimensional features at each time step. A batch normalization layer is put after ResNet50, with two dense layers ad_down and ad_up used for bottleneck compression and expansion between feature spaces of 32 and 2048 units. As a result, it functions as a light-weight attention-like mechanism. To further improve temporal abstraction, a pooling layer and a Bidirectional GRU layer with 128 units are used to capture connections over a long period. A dropout layer separates every convolutional, recurrent and dense block to reduce the possibility of overfitting. Lastly, a Dense classifier layer adds a softmax activation to predict over the dialect options.

We performed experiments with the freezing of the ResNet50 backbone during training to maintain the pretrained spatial features, as well as the fine-tuning of the model by making all the layers trainable. This approach allowed us to study the effect of feature reuse compared to complete adaptation on future performance results.

Model: "CRNN2D"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 24, 512, 32)	2,048
max_pooling2d (MaxPooling2D)	(None, 12, 256, 32)	0
batch_normalization (BatchNormalization)	(None, 12, 256, 32)	128
conv2d_1 (Conv2D)	(None, 12, 256, 64)	30,784
max_pooling2d_1 (MaxPooling2D)	(None, 6, 128, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 6, 128, 64)	256
conv2d_2 (Conv2D)	(None, 6, 128, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 3, 64, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 3, 64, 128)	512
conv2d_3 (Conv2D)	(None, 3, 64, 512)	590,336
max_pooling2d_3 (MaxPooling2D)	(None, 2, 32, 512)	0
batch_normalization_3 (BatchNormalization)	(None, 2, 32, 512)	2,048
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
reshape (Reshape)	(None, 1, 512)	0
bidirectional (Bidirectional)	(None, 1, 128)	221,952
bidirectional_1 (Bidirectional)	(None, 256)	198,144
dropout_1 (Dropout)	(None, 256)	0
dense (Dense)	(None, 20)	5,140

Total params: 1,125,204 (4.29 MB)
Trainable params: 1,123,732 (4.29 MB)
Non-trainable params: 1,472 (5.75 KB)

Figure 7.1: Architecture overview: Custom stacked CNN as feature encoder, followed by stacked RNN classifier.

The architecture has a total of 24.55 million parameters, with 24.5 million trainable parameters which is shown in Table 7.2. The architecture was tried out using both 3-channel and 7-channel inputs. With 7-channel inputs, the adapter layer adjusts its size to match the ResNet requirement of 3 channels.

7.4 Model Optimization and Regularization

We apply different popular optimization techniques to achieve stable and efficient training of the CRNN.

- Early Stopping: The process stops after a set number of epochs without a drop in validation loss, so there is no overfitting to the data used for training.

Layer Component	Purpose
`ZeroPadding2D`	Maintain edge dimensions
`Conv2D Adapter`	Match ResNet input channel size
`ResNet50 (pretrained)`	Deep spatial feature extractor
`BatchNormalization`	Normalize feature maps
`Dense Bottleneck (Down/Up)`	Feature compression and expansion
`MaxPooling2D`	Temporal dimensionality reduction
`Bidirectional GRU`	Sequential context modeling
`Dropout Layers`	Regularization
`Dense Softmax Layer`	Final classification over dialects

Table 7.2: Key layers and roles in ResNet50 + BiGRU architecture

- Batch Normalization: Once fully connected to the convolutional layer, Batch Normalization processes the neural inputs to avoid internal covariate shift and allows the use of a higher learning rate.
- Learning Rate Scheduling: To balance quick learning in the beginning and fine-tuning as we approach the solution, we use a reduce-on-plateau scheduler that lowers the learning rate when validation performance stabilizes.
- Regularization Layers: These techniques combined accelerate model convergence, improve generalization, and prevent overfitting during training.
  - Dropout: Disconnecting some of the units at each update increases the chance that the model does not become overly reliant on particular units.
  - Weight Penalty: By multiplying losses with a penalty term based on the magnitude of the weights, we discourage very large values and maintain well-behaved functions.

7.5 Hyperparameter Optimization

We apply Bayesian optimization to look through the hyperparameter space and find sets of values that achieve the highest validation accuracy. The model uses a Gaussian process to connect hyperparameter values with performance and the acquisition function recommends new settings that mix seeking out unknown regions with focusing on strong ones. Bayesian optimization achieves high-performing hyperparameters faster than grid or random searches which saves time and money on training several CRNN variants. It allows us to set up our model parameters in a nearly optimal way, without us having to conduct countless manual experiments.

Model: "crnn2d_classifier"

Layer (type)	Output Shape	Param #
pad (ZeroPadding2D)	(1, 32, 512, 8)	0
norm_in (BatchNormalization)	(1, 32, 512, 8)	32
chan_adapter (Conv2D)	(1, 32, 512, 3)	27
resnet50 (Functional)	(None, 1, 16, 2048)	23,587,712
norm_back (BatchNormalization)	(1, 1, 16, 2048)	8,192
ad_down (Dense)	(1, 32)	65,568
ad_up (Dense)	(1, 2048)	67,584
time_pool (MaxPooling2D)	?	0
conv_drop (Dropout)	?	0
bidir_rnn (Bidirectional)	(1, 128)	811,776
rnn_drop (Dropout)	?	0
dense_fc (Dense)	(1, 128)	16,512
dense_drop (Dropout)	?	0
classifier (Dense)	(1, 5)	645

Total params: 24,558,048 (93.68 MB)
Trainable params: 24,500,816 (93.46 MB)
Non-trainable params: 57,232 (223.56 KB)

Figure 7.2: Architecture overview: ResNet50 as feature encoder followed by RNN classifier. (5-class experiment).

7.6 Training Configuration

The model was trained using the Adam optimizer with sparse categorical cross-entropy loss. The training ran for up to 100 epochs with early stopping enabled. The configuration included:

- Loss Function: Sparse Categorical Crossentropy
- Evaluation Metric: Sparse Categorical Accuracy
- Callbacks: EarlyStopping (patience = 5), ReduceLROnPlateau (factor = 0.5, min_lr = 1×10^{-5}), ModelCheckpoint, TensorBoard, CSV Logger
- Save Format: keras

Chapter 8

Result Analysis

8.1 Evaluation Metrics

To assess the performance of our dialect classification models, we used standard classification evaluation metrics:

- **Validation and Test Accuracy:** Measures the percentage of correctly classified dialects in validation and test sets.
- **Precision:** Indicates the proportion of correct positive predictions relative to the total predicted positives.
- **Recall:** Represents the proportion of actual positives that were correctly predicted.
- **F1-Score:** Harmonic mean of precision and recall, providing a balance between the two.
- **Confusion Matrix:** A confusion matrix shows how predictions compare with actual labels, highlighting misclassifications. It helps identify which dialects are often confused and assesses the model's sensitivity to similar dialects.

8.2 Comparative Evaluation

We evaluated multiple architectures using varying CNN backbones, RNN types, feature sets, and preprocessed audio inputs. Table 8.1 presents the comparative analysis.

With ResNet50 as the backbone, freezing the ImageNet weights leads to a test accuracy of 80.82% (F1 0.83), but fine-tuning all layers raises accuracy to 85.33% (F1 0.85) which is a 4.5 percentage-point gain from fine-tuning. Nevertheless, the results show that the four-layer

Table 8.1: Comparative performance of different models and configurations.

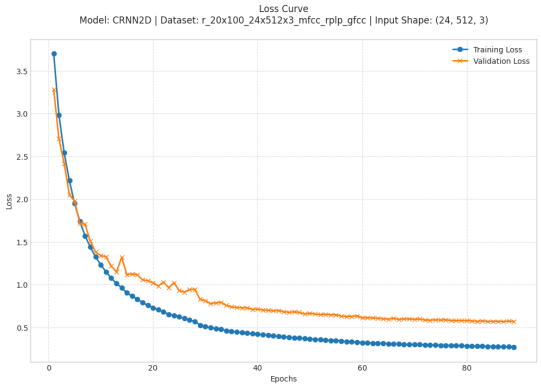
CNN Layer	RNN	Data	Features	VAL ACC	TEST ACC	Precision	Recall	F1-Score
Custom	BiGRU	raw	MFCC, GFCC, RPLP	**91.08**	**90.31**	**90.41**	**90.31**	**90.31**
Custom	BiLSTM	raw	MFCC, GFCC,RPLP	89.66	88.33	88.49	88.34	88.35
Custom	BiGRU	raw	MFCC, RPLP, LPCC, GFCC, CQCC, BFCC, LFCC	87.66	87.29	87.45	87.29	87.31
Custom	BiGRU	noise-reduced	MFCC, GFCC, RPLP	86.67	86.10	86.38	86.10	86.15
ResNet50[b]	BiGRU	raw	MFCC, Delta_MFCC, GFCC, RPLP	84.88	85.33	85.41	85.33	85.25
ResNet50[a]	BiGRU	raw	MFCC, Delta_MFCC, GFCC, RPLP	80.03	80.82	81.18	80.82	80.83

[a] Backbone frozen (ImageNet weights fixed).
[b] Backbone fully trainable (all layers fine-tuned).

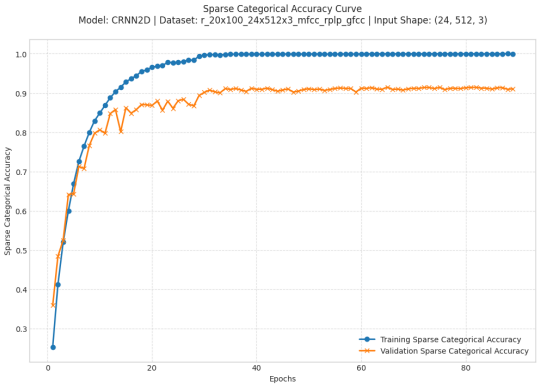
CNN + BiGRU on raw MFCC/GFCC/RPLP features performs even better than the fine-tuned ResNet50. Performance on the test set shows that the BiLSTM variant (88.33%) and the noise-reduced setup (86.10%) outperform the frozen backbone. So, even though deep pretrained features are useful, a carefully designed lightweight CNN is still better for this task.

8.3 Loss and Accuracy Curves

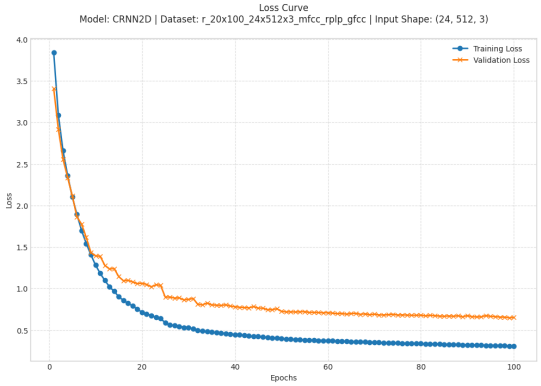
The training and validation accuracy and loss curves for each model are shown in Figure 8.1. These plots help visualize convergence behavior, generalization performance, and potential overfitting.



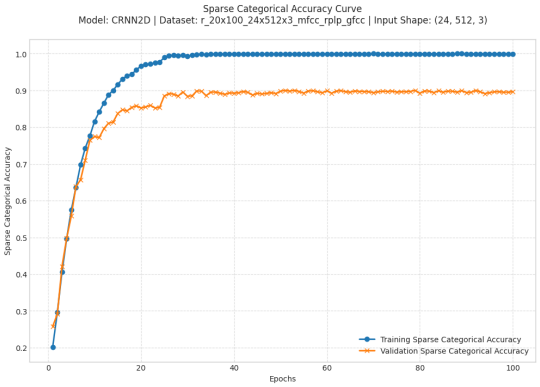
(a) Loss – Custom CNN+BiGRU (raw, 3 features)



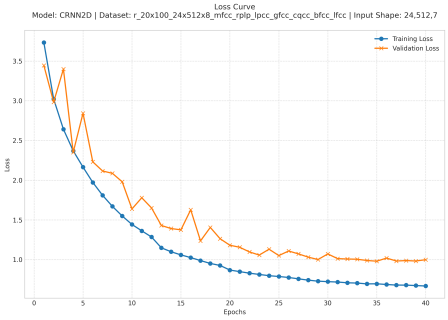
(b) Accuracy – Custom CNN+BiGRU (raw, 3 features)



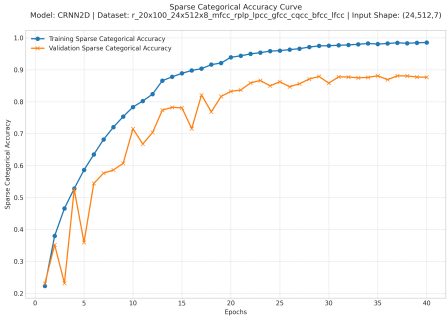
(c) Loss – Custom CNN+BiLSTM (raw, 3 features)



(d) Accuracy – Custom CNN+BiLSTM (raw, 3 features)

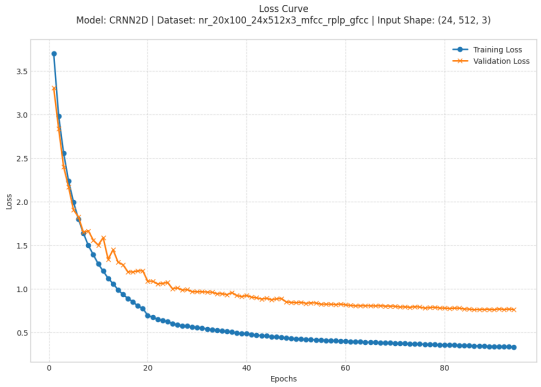


(e) Loss – Custom CNN+BiGRU (raw, 7 features)

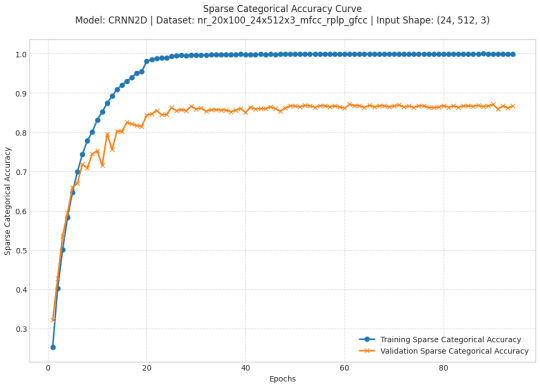


(f) Accuracy – Custom CNN+BiGRU (raw, 7 features)

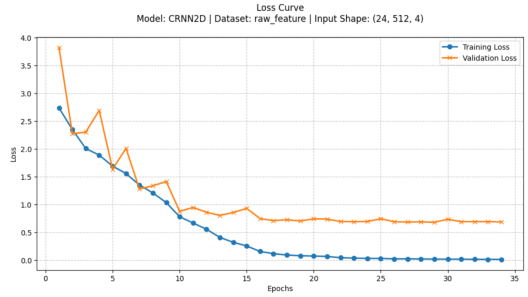
Figure 8.1: Training and validation loss/accuracy curves for all major model configurations (Part 1).



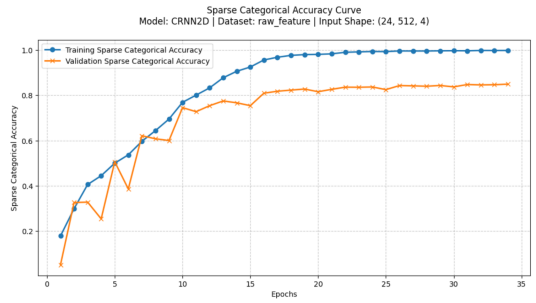
(g) Loss – Custom CNN+BiGRU (noise-reduced, 3 features)



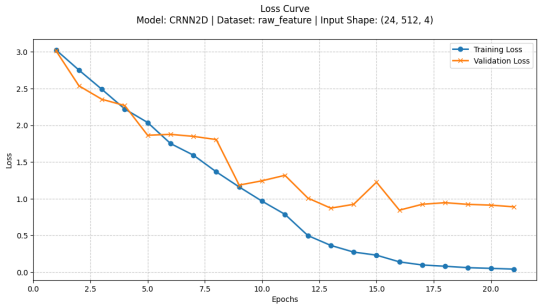
(h) Accuracy – Custom CNN+BiGRU (noise-reduced, 3 features)



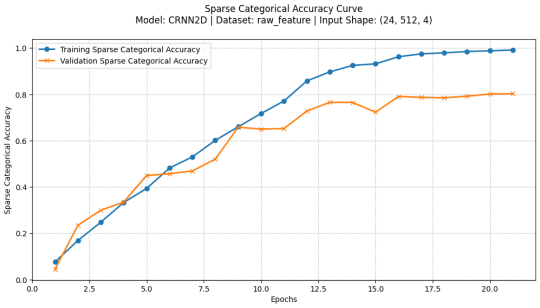
(i) Loss – ResNet50+BiGRU (raw, 4 features, fine-tuned)



(j) Accuracy – ResNet50+BiGRU (raw, 4 features, fine-tuned)



(k) Loss – ResNet50+BiGRU (raw, 4 features, fixed ImageNet weights)



(l) Accuracy – ResNet50+BiGRU (raw, 4 features, fixed ImageNet weights)

Figure 8.1: Training and validation loss/accuracy curves for all major model configurations (Part 2).

8.4 Confusion Matrix of Best Performing Model

To further analyze performance across the 20 dialect classes, we visualized the confusion matrix of our best model—*CNN-BiGRU trained on raw audio using MFCC, GFCC, and RPLP features.*

The matrix shows that most dialects are correctly classified with minimal confusion between phonetically similar ones. High classification rates for dialects like *bagerhat, jessore, manikganj* and *standard Bengali* indicate strong model generalization.

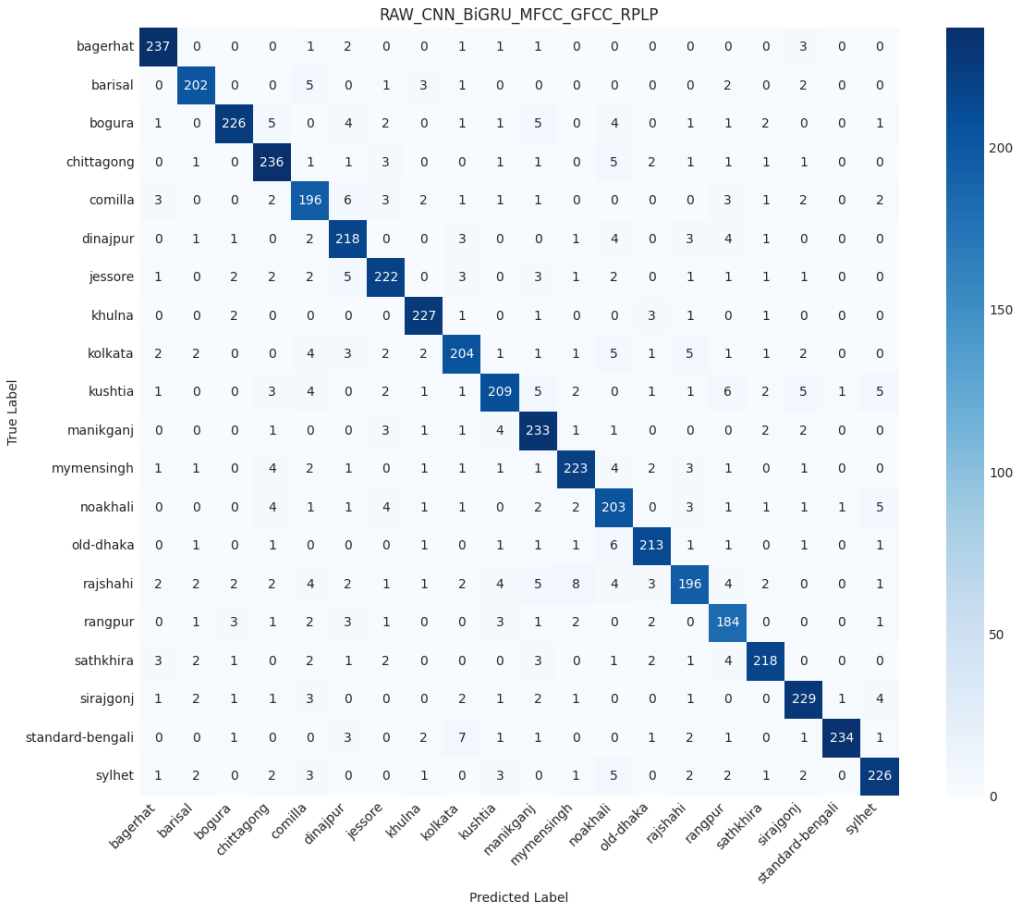


Figure 8.2: Confusion Matrix for Best Performing Model (CNN-BiGRU on Raw - MFCC, GFCC, RPLP)

Chapter 9

Future Directions in Bengali DID

9.1 Expansion of BengDiDa Dataset

9.1.1 Fieldwork for Data Collection

Expansion of the BengDiDa dataset is the foremost development we aim for in the future scope of this project. Currently, the corpus encompasses 20 Bengali dialects with 48,000 samples. The goal is to incorporate every single dialect of Bengali so that comprehensive DID tasks can be performed. To achieve this goal, there is no other viable alternative to extensive fieldwork for dialect speech data collection. Bengali dialect speakers are abundant for each chosen dialect. However, the data collection process would involve systematic planning and execution to ensure the diversity and quality of the collected audio data are maintained and well represented.

For speech data collection, the recruitment of volunteers who can gather and provide authentic audio samples of their native dialects. In field data collection, the researcher have full control over a lot of the parameters, so the task is more flexible. This control allows researchers to fulfil specific dataset requirements on the fly. For instance, if a class requires more speech data from women, it is simple to obtain that data by concentrating on that group. As a result, the data is completely controlled by human input, allowing both easier preprocessing, enhanced inclusivity and richness.

In addition, researchers should always consider ethics before and during their fieldwork. Having informed consent from the participants is very important in collecting speech samples [46]. People contributing should be informed about what their data will be used for, while being treated with respect and having their data kept confidential.

9.1.2 Inclusion of More Dialects

Recognizing various colloquial Bengali dialects in the BengDiDa dataset is an important improvement for Bengali DID systems. There are many different dialects spoken in the Bengali speaking regions and because it is hard to define dialects, it makes distinguishing between them unclear. Hence, it is theoretically possible that there are more than 20 dialects. In other words, expanding the dataset to more than 20 dialects will improve the data's variety and help the models improve their generalization capacity.

9.2 Advances in Self-Supervised Learning

9.2.1 End-to-End Learning Approaches

A further approach is end-to-end learning methods. Unifying all steps, end-to-end models use CRNNs and Bi-LSTMs to pick up both spatial and temporal information in the data. Performance yielded increases in low-resource scenarios through self-supervised and multi-task learning, while reinforcement learning and domain attentive fusion improve how well tasks are adapted and generalized. Feature adaptation and reducing noise increases a network's robustness. Having fully connected layers on top of Bi-LSTM increases the accuracy [72].

9.2.2 Applications in Dialectal Speech Processing

Thanks to self-supervised learning, features can be extracted from unlabeled data that aids in handling dialectal speech processing. Methods such as FHV-AE and CNNs can perceive sounds, while ILPR-LMS and downsampling make the approach more efficient. BengDiDa aims to develop Bengali DID mechanisms in the speech processing community. By adding phonotactic, lexical and acoustic elements to CNNs, you can improve classification. The main priority is to use tools like rich datasets and advanced models such as CRNNs.

9.3 Parameter-Efficient Transfer Learning

9.3.1 Techniques for Model Adaptation

Using a model adaptation approach benefits Bengali DID by helping it to function well in situations new to the model. Feature aggregation prevents overfitting in datasets which are small and finds ways for the network to learn a person's attributes. Combining CNNs, LSTMs,

attention and bottleneck features leads to better classification, mainly because of adapting to Bengali phonetic-prosodic characteristics [44].

9.3.2 Efficiency in Resource-Constrained Environments

Spatial and temporal elements are both handled effectively by CRNNs for efficient Bengali DID [19, 4]. Parameter reduction by reducing noise and changing the word features, relevance can be increased [20, 24]. Both transfer learning and semi-supervised learning help achieve accurate results at a low cost by using unlabeled data [49, 7, 19]. By performing both weighted-sum feature aggregation and phonotactic modeling, low-resource efficiency is achieved [6, 49]. Further efficiencies are achieved by using lightweight models such as TDNNs, wav2vec 2.0 and multi-resolution filters [4, 24]. Both the Hybrid CRNN and TEL-enhanced DenseNet techniques are scalable for solving DID problems [79].

9.4 Integration with Real-World Applications

9.4.1 Voice Assistants

Voice assistants have difficulty with dialects; with LID, they recognize the Bengali language much better [13]. Owing to CRNNs, transformers, prosodic features and VAD, the speech recognition system performs better in cases where there is noise or change in duration [19, 49, 2]. GMMs allow for statistical analysis [18], while BengDiDa and accent-specific tuning make GMMs easier to adjust to different acoustic scenarios [80, 72].

9.4.2 Speech-to-Text Systems

Bengali DID uses speech recognition to map the sound and rhythm of the language into text [8]. Improved prediction and accuracy are possible using methods that reduce noise and use CRNN/CNN models [26, 8]. These underlying mechanisms support robust ASR, HCI, and biometrics [81, 7], but there is a need for more data and models that focus on dialects [52, 28].

9.5 Broader implications of DID

DID is no longer niche, it is the hinge that lets speech technology swing open for everyone. Bengali is a very essential but under-researched testbed where we experiment with twenty dialects using CRNN. We show that padding tiny 24-bin spectrograms into an ImageNet backbone, then

shaking them with SpecAugment, keeps accuracy high even when the words come wrapped in noise [80]. Prosody and bottleneck features have been particularly useful in works [18, 82]. All of this results in a smarter ASR. This maybe a small step, but it is the step that plants the seed for growing branches of research in low-resource regional speech technology.

There is a cultural impact as well because each dialect is a branch of a language that carries a story, a trace of the lineage of ancestry. These dialects may go extinct due to neglected adaptation by technology. BengDiDa's 48 000 clips freeze these stories in WAV form and provides a hedge against erosion [19]. CRNNs then act as microscopes, they reveal the subtle features that makeup the dialect distinguishable from the others, spilling out many insights. Linguists can chase those trails and technologists can build inclusive tools.

Chapter 10

Conclusion

Looking into dialect identification in Bengali shows how language, culture and the advancements of technology are closely interlinked. Being able to recognize speech through an ASR system becomes difficult due to the many Bengali dialects, so it is clear that we must develop specific datasets and new ways to model speech. BengDiDa aims to tackle these limitations as it offers a reliable and rich audio archive that includes a large span of dialects.

More research is showing that applying Convolutional Recurrent Neural Networks can help a lot with detecting and classifying the dialects of speech. When these models are paired with curated feature extraction and preprocessing steps, they are able to acknowledge the small speech differences that characterize distinct dialects. In addition, using special context-sensitive embeddings and advanced techniques to reduce noise strengthens the performance of the system.

Identifying spoken languages is important not only because of new technologies, it also helps preserve language diversity and study the dynamics of regional communication. Through extensive work, researchers promote initiatives that save cultural heritage and improve communication among different communities.

Moving forward, overcoming data scarcity issues is necessary, and it's important to create modeling approaches that can handle spoken language's dynamic behavior. It is necessary to account for similar sounds among neighboring languages and to make balanced, inclusive and representative datasets to avoid bias in models.

Further study of Bengali dialect identification greatly impacts speech technology and gives us a better understanding of how language diversity thrives. When linguists, computer scientists and data engineers join forces, it's more likely that such sophisticated systems will develop further. Interdisciplinary collaboration will make automatic speech recognition systems better, so that communication tools meet the challenges and needs of speakers worldwide as language use evolves.

Bibliography

[1] N. B. Chittaragi, A. Limaye, N. T. Chandana, B. Annappa, and S. G. Koolagudi, "Automatic text-independent kannada dialect identification system," vol. 863, 1 2019.

[2] J. young Lee and K. wha Kim, "Korean dialect identification based on intonation modeling," 9 2021.

[3] S. A. Pa, R. T. Mb, and R. K. Sc, "Comparative study of different classifiers for malayalam dialect recognition system," *Procedia Technology*, vol. 24, pp. 1080–1088, 5 2016.

[4] D. Escobar Grisales, C. D. Rios-Urrégo, E. Noe th, and J. R. Orozco, "Colombian dialect recognition from call-center conversations using fusion strategies," 11 2022.

[5] W. YING, L. ZHANG, and H. DENG, "Sichuan dialect speech recognition with deep lstm network," *Front. Comput. Sci.*, 7 2019.

[6] N. Tibi and M. A. B. Messaoud, "Automatic arabic dialect identification using deep learning," 7 2022.

[7] K. Lounnas, M. Lichouri, and M. Abbas, "Enhancing spoken dialect identification with stacked generalization of deep learning models," *Multimedia Tools and Applications*, 8 2024.

[8] J. young Lee, K. wha Kim, and U. Author, "Korean dialect identification based on an ensemble of prosodic and segmental feature learning for forensic speaker profiling," 12 2022.

[9] P. Prosun Das, S. M. Allayear, R. Amin, and Z. Rahman, "Bangladeshi dialect recognition using mel frequency cepstral coefficient, delta, delta-delta and gaussian mixture model," 2 2016.

[10] N. B. Chittaragi and S. G. Koolagudi, "Acoustic-phonetic feature based kannada dialect identification from vowel sounds," *International Journal of Speech Technology*, vol. 1, 10 2019.

[11] S. Sinha, A. Jain, and S. S. Agrawal, "Fusion of multi-stream speech features for dialect classification," *CSIT*, vol. 2, 6 2015.

[12] T. Nejib and M. A. Ben Messaoud, "Implementation of dialect identification integrated with internet of things," 10 2023.

[13] A. Mandal, S. Pal, I. Dutta, M. Bhattacharya, and S. K. Naskar, "Is attention always needed? a case study on language identification from speech," *J. ACM*, vol. 37, 8 2018.

[14] K. Alam, M. H. Bhuiyan, and M. F. Monir, "Bangla speaker accent variation classification from audio using deep neural networks: A distinct approach," 10 2023.

[15] N. B. Chittaragi, P. Hegde, S. K. P. Mothukuri, and S. G. Koolagudi, "Spectral feature based kannada dialect classification from stop consonants," in *PReMI 2019*, vol. 11941, 10 2019.

[16] H. C. Das and U. Bhattacharjee, "K-means clustering and support vector machine for assamese dialect identification," 1 2024.

[17] R. Mahdavi, A. Bastanfard, and D. Amirkhani, "Persian accents identification using modeling of speech articulatory features," 5 2020.

[18] N. B. Chittaragi, A. Prakash, and S. G. Koolagudi, "Dialect identification using spectral and prosodic features on single and ensemble classifiers," *Arab J Sci Eng*, 10 2017.

[19] T. Ngo Trong, K. Jokinen, and V. Hautamäki, "Enabling spoken dialogue systems for low-resourced languages—end-to-end dialect recognition for north sami," vol. 579, 11 2019.

[20] V. Kadyan, A. Singh, and P. Wadhwa, "Hindi dialect (bangro) spoken language recognition (hd-slr) system using sphinx3," vol. 479, 1 2017.

[21] A. Alashban, M. Qamhan, A. Meftah, and Y. Alotaibi, "Spoken language identification system using convolutional recurrent neural network," *Appl. Sci.*, vol. 12, 9 2022.

[22] Q. T. Duong and V. H. Do, "Development of accent recognition systems for vietnamese speech," 10 2021.

[23] L. SUN, "Using prosodic and acoustic features for chinese dialects identification," 8 2020.

[24] A. Angra, H. Muralikrishna, D. A. Dinesh, and V. Thenkanidiyoor, "Exploring aggregated wav2vec 2.0 features and dual-stream tdnn for efficient spoken dialect identification," *IEEE Access*, vol. 13, 1 2025.

[25] S. M. S. I. Badhon, M. H. R. Nobel, F. Rupon, and S. Abujar, "Bengali accent classification from speech using different machine learning and deep learning techniques," 1 2021.

[26] S. Bougrine, H. Cherroun, and D. Ziadi, "Prosody-based spoken algerian arabic dialect identification," vol. 128, 3 2018.

[27] R. Kallooravi Thandil and K. P. M. Basheer, "Speaker independent accent based speech recognition for malayalam isolated words: An lstm-rnn approach," vol. 1546, 1 2022.

[28] P. Hegde, N. B. Chittaragi, S. K. P. Mothukuri, and S. G. Koolagudi, "Kannada dialect classification using cnn," vol. 11987, 10 2020.

[29] N. B. Chittaragi, S. K. P. Mothukuri, P. Hegde, and S. G. Koolagudi, "Robust dialect identification system using spectro-temporal gabor features," 10 2018.

[30] N. B. Chittaragi, A. Limaye, N. T. Chandana, B. Annappa, and S. G. Koolagudi, "Automatic text-independent kannada dialect identification system," vol. 863, 1 2019.

[31] N. B. Chittaragi, S. K. P. Mothukuri, P. Hegde, and S. G. Koolagudi, "Robust dialect identification system using spectro-temporal gabor features," 10 2018.

[32] M. Alrehaili, T. Alasmari, and A. Aoalshutayri, "Arabic speech dialect classification using deep learning," in *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, 1 2023.

[33] S. Bougrine, A. Chorana, A. Lakhdari, and H. Cherroun, "Toward a Web-based Speech Corpus for Algerian Dialectal Arabic Varieties," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, (Valencia, Spain), pp. 138–146, Association for Computational Linguistics, 2017.

[34] H. Schröter, T. Rosenkranz, A. N. Escalante-B., and A. Maier, "DeepFilterNet: Perceptually motivated real-time speech enhancement," in *INTERSPEECH*, 2023.

[35] C. K. A. Reddy, V. Gopal, and R. Cutler, "DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *2020 International Conference on Acoustics, Speech, and Signal Processing*, pp. 6493–6497, IEEE, Oct. 2020.

[36] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *Interspeech 2021*, pp. 2127–2131, 2021.

[37] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," pp. 18–24, 01 2015.

[38] A. Malek, "Spafe: Simplified python audio features extraction," *Journal of Open Source Software*, vol. 8, no. 81, p. 4739, 2023.

[39] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "TSFEL: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, 2020.

[40] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, 2016.

[41] F. Eyben, M. Wöllmer, and B. Schuller, "opensmile – the munich versatile and fast open-source audio feature extractor," pp. 1459–1462, 01 2010.

[42] Z. Alhakeem, S.-I. Jang, and H.-G. Kang, "Disentangled representations in local-global contexts for arabic dialect identification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, 12 2024.

[43] Y. Korkmaz and A. Boyacı, "A comprehensive turkish accent/dialect recognition system using acoustic perceptual formants," *Applied Acoustics*, vol. 193, 4 2022.

[44] D. Alhelal and T. Inan, "Designing a system to recognize main arabic dialects," *IEEE Access*, vol. 12, 11 2024.

[45] Y. Lei and J. H. L. Hansen, "Dialect classification via text-independent training and testing for arabic, spanish, and chinese," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, 1 2011.

[46] M. Nanmalar, P. Vijayalakshmi, and T. Nagarajan, "Literary and colloquial tamil dialect identification," *Circuits, Systems, and Signal Processing*, vol. 41, 3 2022.

[47] H. Zhang, X. Liu, and Y. Shao, "Chinese dialect tone's recognition using gated spiking neural p systems," *Journal of Membrane Computing*, vol. 4, 11 2022.

[48] H. Zhang, X. Liu, and Y. Shao, "Chinese dialect tone's recognition using gated spiking neural p systems," *Journal of Membrane Computing*, vol. 4, 11 2022.

[49] W. Lin, M. Madhavi, R. K. Das, and H. Li, "Transformer-based arabic dialect identification," 11 2020.

[50] C. Bartz, T. Herold, H. Yang, and C. Meinel, "Language identification using deep convolutional recurrent neural networks," 8 2017.

[51] R. Rahmawati and D. P. Lestari, "Java and sunda dialect recognition from indonesian speech using gmm and i-vector," 1 2017.

[52] P. Hegde, N. B. Chittaragi, S. K. P. Mothukuri, and S. G. Koolagudi, "Kannada dialect classification using cnn," vol. 11987, 1 2020.

[53] Z. Qi, Y. Ma, M. Gu, Y. Jin, S. Li, Q. Zhang, and Y. Shen, "End-to-end chinese dialect identification using deep feature model of recurrent neural network," in *2018 IEEE 4th International Conference on Computer and Communications*, 10 2018.

[54] M. Wan, J. Ren, M. Ma, Z. Li, R. Cao, and Q. Gao, "Deep neural network based chinese dialect classification," 10 2021.

[55] R. S. A. Podila, G. S. S. Kommula, R. K, S. Vekkot, and D. Gupta, "Telugu dialect speech data set creation and recognition using deep learning techniques," in *2022 IEEE 19th India Council International Conference (INDICON)*, 12 2022.

[56] S. Kibria, M. S. Rahman, M. R. Selim, and M. Z. Iqbal, "Acoustic analysis of the speakers' variability for regional accent-affected pronunciation in bangladeshi bangla: A study on sylheti accent," *IEEE Access*, vol. 8, 2 2020.

[57] A. Arronte Alvarez and E. S. Abdelaal Issa, "Learning intonation pattern embeddings for arabic dialect identification," *Unknown Journal*, 8 2020.

[58] M. Najafi, S. Khurana, S. Shon, A. Ali, and J. Glass, "Exploiting convolutional neural networks for phonotactic based dialect identification," 4 2018.

[59] H. C. Das and U. Bhattacharjee, "Assamese dialect identification using semi-supervised learning," 6 2022.

[60] P. Sarmah and L. Dihingia, "Assamese dialect identification from vowel acoustics," *Lecture Notes in Networks and Systems*, vol. 238, 1 2022.

[61] Kasiprasad.Mannepalli, P. Sastry, and V.Rajesh, "Accent detection of telugu speech using prosodic and formant features," vol. 2015, 1 2015.

[62] A. Roy Choudhury, N. B. Chittaragi, and S. G. Koolagudi, "Dialect recognition system using excitation source features," 1 2018.

[63] R. Kethireddya, S. R. Kadiri, and S. V. Gangashetty, "Exploration of temporal dynamics of frequency domain linear prediction cepstral coefficients for dialect classification," *Applied Acoustics*, vol. 188, 12 2021.

[64] R. K. Thandil, K. P. M. Basheer, and V. K. Muneer, "End-to-end multi-dialect malayalam speech recognition using deep-cnn, lstm-rnn, and machine learning approaches," vol. 163, 10 2023.

[65] R. S. A. Podila, G. S. S. Kommula, R. K, S. Vekkot, and D. Gupta, "Telugu dialect speech data set creation and recognition using deep learning techniques," 12 2022.

[66] S. Shon, A. Ali, Y. Samih, H. Mubarak, and J. Glass, "Adi17: A fine-grained arabic dialect identification dataset," 5 2020.

[67] J. Sun, G. Zhou, H. Yang, and M. Wang, "End-to-end tibetan ando dialect speech recognition based on hybrid ctc/attention architecture," 11 2019.

[68] X. Xu, H. Dinkel, M. Wu, and K. Yu, "A crnn-gru based reinforcement learning approach to audio captioning," 11 2020.

[69] S. Darjaa, R. Sabo, M. Trnka, M. Rusko, and G. Múcsková, "Automatic recognition of slovak regional dialects." File available at: `8. AutomaticRecognitionofSlovakRegionalDialects.pdf`.

[70] S. Bougrine, H. Cherroun, and A. Abdelali, "Spoken arabic algerian dialect identification," 5 2018.

[71] U. Singh, A. Gupta, D. Bisharad, and W. Arif, "Foreign accent classification using deep neural nets," *Journal of Intelligent & Fuzzy Systems*, vol. 38, 1 2020.

[72] M. Tzudir, M. Bhattacharjee, P. Sarmah, and S. R. M. Prasanna, "Low-resource dialect identification in ao using noise robust mean hilbert envelope coefficients," in *2022 National Conference on Communications (NCC)*, 3 2022.

[73] A. Alashban, M. Qamhan, A. Meftah, and Y. Alotaibi, "Spoken language identification system using convolutional recurrent neural network," *Applied Sciences*, vol. 12, p. 9181, 09 2022.

[74] X. Miao and I. McLoughlin, "Lstm-tdnn with convolutional front-end for dialect identification in the 2019 multi-genre broadcast challenge," 2019.

[75] S. Kakouros, K. Hiovain, M. Vainio, and J. Šimko, "Dialect identification of spoken north sámi language varieties using prosodic features," in *Speech Prosody 2020*, pp. 625–629, 2020.

[76] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, "Exploiting spectro-temporal locality in deep learning based acoustic event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, p. 26, Sept. 2015.

[77] M. Najafian, S. Khurana, S. Shan, A. Ali, and J. Glass, "Exploiting convolutional neural networks for phonotactic based dialect identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5174–5178, 2018.

[78] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015.

[79] R. vander Merwe, "Triplet entropy loss: Improving the generalisation of short speech language identification systems," 12 2020.

[80] J. Yi, Z. Wen, J. Tao, H. Ni, and B. Liu, "Ctc regularized model adaptation for improving lstm rnn based multi-accent mandarin speech recognition," *J Sign Process Syst*, 1 2017.

[81] P. S. Hossain, A. Chakrabarty, K. Kim, and M. J. Piran, "Multi-label extreme learning machine (mlelms) for bangla regional speech recognition," *Appl. Sci.*, vol. 12, 5 2022.

[82] S. Shon, W.-N. Hsu, and J. Glass, "Unsupervised representation learning of speech for dialect identification," 12 2018.

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?



The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

final report

ORIGINALITY REPORT

4%	3%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ar.iub.edu.bd Internet Source	1%
2	Rani Oomman Panicker, Kaushik S. Kalmady, Jeny Rajan, M.K. Sabu. "Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods", Biocybernetics and Biomedical Engineering, 2018 Publication	<1%
3	www.mdpi.com Internet Source	<1%
4	arxiv.org Internet Source	<1%
5	ijmrset.com Internet Source	<1%
6	ollamabasic.blogspot.com Internet Source	<1%
7	scholar.uoc.ac.in Internet Source	<1%
8	www.researchgate.net Internet Source	<1%
9	Amit Kumar Tyagi. "Data Science and Data Analytics - Opportunities and Challenges", CRC Press, 2021	<1%

	Publication	
10	Dev, Antora. "Enhancing Epileptic Seizure Detection With Advanced Wavelet Transforms and Hybrid Deep Learning Models: A Comparative Study Using EEG and MEG Data", Idaho State University, 2024 Publication	<1 %
11	projekter.aau.dk Internet Source	<1 %
12	link.springer.com Internet Source	<1 %
13	serp.ai Internet Source	<1 %
14	www.geeksforgeeks.org Internet Source	<1 %
15	J.S. Hawker, J.L. Ozbun, H. Ozaki, E. Greenberg, Jack Preiss. "Interaction of spinach leaf adenosine diphosphate glucose α-1,4-glucan α-4-glucosyl transferase and α-1,4-glucan, α-1,4-glucan-6-glycosyl transferase in synthesis of branched α-glucan", Archives of Biochemistry and Biophysics, 1974 Publication	<1 %
16	xplorestaging.ieee.org Internet Source	<1 %
17	Fabio Vesperini, Leonardo Gabrielli, Emanuele Principi, Stefano Squartini. "Polyphonic Sound Event Detection by Using Capsule Neural Networks", IEEE Journal of Selected Topics in Signal Processing, 2019 Publication	<1 %



Advancing Bengali Dialect Identification (DiD) Through The BengDiDa Dataset & Dialect Classification

By

Md Zakaria Kabir, Md. Zannat Chowdhury Sagar

Student ID: **1921371, 1920392**

Spring, 2025

The student modified the thesis final report as per the recommendation made by his or her academic supervisor and/or panel members during final viva, and the department can use this version for archiving.

Signature of the Supervisor

Md Asif Bin Khaled

Senior Lecturer

Department of Computer Science & Engineering

School of Engineering, Technology & Sciences

Independent University, Bangladesh