AI-FICATION 2025

শব্দতরী: Where Dialects Flow into Bangla

Team DejaView

Ruwad Naswan
Shadab Tanjeed
Abrar Zahin Raihan

November 22, 2025

Problem Statement & Dataset Overview

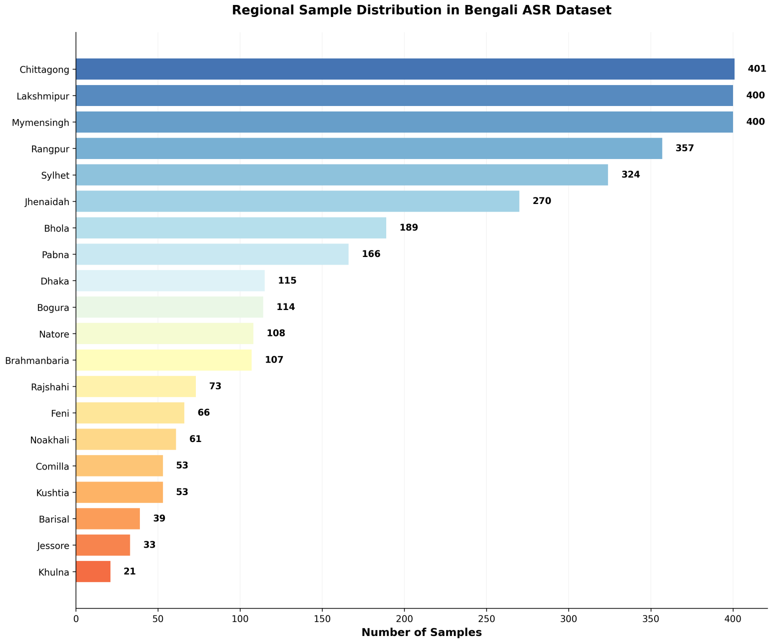
Challenge: Transcribe 20 regional Bangladeshi dialects into standard Bangla text with high accuracy despite phonetic variations and diverse acoustic conditions.

Dataset Statistics

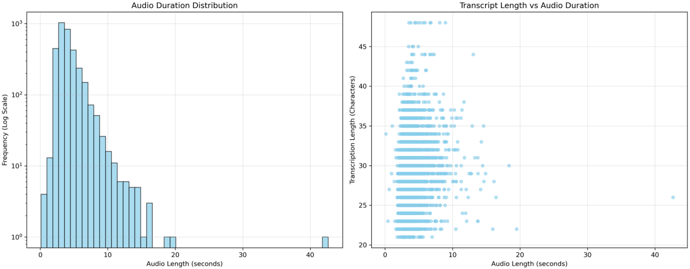
- **Training:** 3,350 audio files
- **Test:** 450 audio files
- **Dialects:** 20 regional variations
- **Format:** 16 kHz, mono WAV
- **Total Duration:** 3.90 hours
- **Avg Duration:** 4.2 seconds/sample

Metric	Value
Vocabulary Size	590
Avg Words/Sample	5.4
Min Duration	0.1s
Max Duration	42.7s

Exploratory Data Analysis (EDA)



Exploratory Data Analysis (EDA)



Observation: Most audio samples are 3-6 seconds with corresponding transcript lengths of 20-40 characters.

Basic Statistics

Duration Analysis

- Mean: 4.2s, Median: 3.7s
- 90% of samples: 3–6s range
- Long-tail distribution (***skewness***: 4.44)

Transcript Statistics

- Avg characters: 29.7
- Avg words: 5.4
- Punctuation ratio: 3.4%

Key Findings

Finding	Impact	Action
Dialect imbalance	Bias towards major dialects	Balanced sampling
Short utterances	Context limitation	Sequence modeling
Noise variations	Recognition errors	Noise augmentation
OOV dialectal words	Transcription gaps	LLM refinement

Data Preprocessing & Feature Engineering

Audio Preprocessing:

- **Input Format:** 16 kHz, mono WAV (standardized)
- **Denoising:** Dynamic spectral gating via `librosa`
- **Normalization:** Relative -3 dB level standardization
- **Padding:** 3.5s silence added to short clips ($<$10s)
- **Zero-length Filtering:** Clips $<$1s eliminated

Text Preprocessing:

- **Transcript Quality:** Pre-cleaned standard Bangla text
- **No Additional Processing:** Foreign words and noise already handled
- **Character Validation:** UTF-8 Bangla Unicode verified

Feature Engineering: Log-Mel spectrograms extracted directly from preprocessed audio for Whisper model input

Key Insight: Minimal text preprocessing needed; primary improvements achieved through audio-level denoising, normalization, and padding strategies.

Balanced Sampling Strategy

Problem

Challenge: *Severe class imbalance across 20 regional dialects*

- *Largest region: 431 samples (Chittagong)*
- *Smallest region: 21 samples (Khulna)*
- *Risk: Model bias towards over-represented dialects*

Solution

Weighted Sampling Approach

- *Calculate inverse frequency weights for each region*
- *Formula:* $Weight_{region} = \frac{\text{Total Samples}}{\text{Number of Classes} \times \text{Region Count}}$
- *Under-represented dialects receive higher sampling probability*
- *Ensures balanced representation during training*

Impact: Khulna (21 samples) receives 20× higher weight than Chittagong (431 samples), ensuring fair dialect representation

External Dataset

Problem

Challenge: *Several dialects had very limited samples (<100) - Risk of poor generalization for minority dialects*

Solution

RegSpeech12 Dataset Integration

- *Source: Regional speech with regional dialect transcriptions*
- *Conversion: Regional text → Standard Bangla using Gemini 2.5 Flash*
- *Process: Automated dialectal standardization for training compatibility*

External Samples Added

Region	Samples
Noakhali	70
Barisal	68
Comilla	62
Chittagong	30
Sylhet	30
Rangpur	30

External Dataset - Distribution

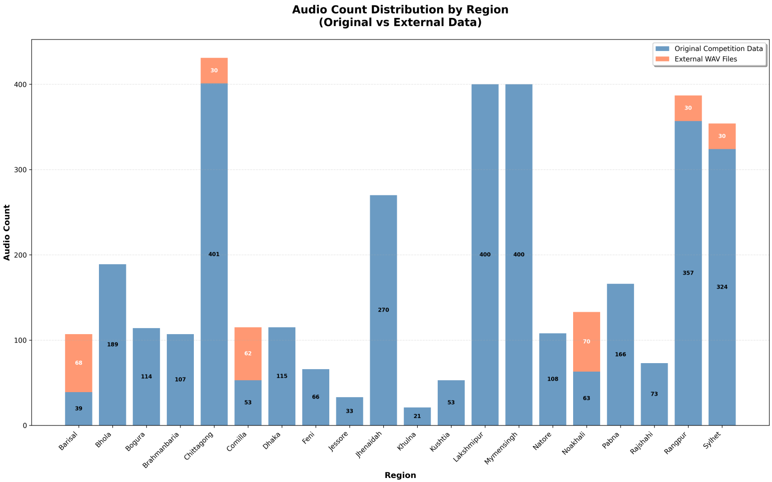


Figure: Audio dataset distribution by region after adding external data

Data Augmentation

Motivation: Limited computational resources prevented adding more external samples; augmentation used to enhance dialectal robustness

Augmentation Techniques

- Time stretching
- Pitch shifting
- Noise injection
- Volume adjustment

Implementation Details

- Probability per technique: $\mathrm{p} = 0.3$
- Applied during training only
- Maintains audio quality
- Preserves phonetic content

Impact: Augmentation combined with external data significantly improved model robustness across all dialects, especially for under-represented regions.

Baseline Model: BengaliAI Whisper Medium (open-source on HuggingFace)
Pre-trained on standard Bengali audio → standard Bengali text

4 Model Variants

Model Variant	Training Approach
Model 1	Frozen Decoder Fine-tuning
Model 2	Full Fine-tuning (Standard → Standard)
Model 3	Regional Classifier Adapter
Model 4	Full Fine-tuning (Regional → Regional)

Frozen Decoder Fine-tuning

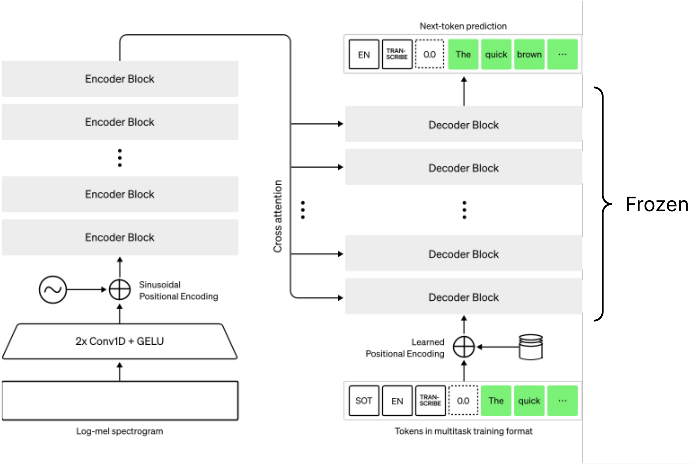


Figure: Whisper model architecture with frozen decoder

Regional Classifier Adapter

Model Architecture

- BengaliAI Whisper Regional ASR Medium
- Encoder-level regional conditioning
- Multi-task setup: ASR + Region Classification

Methodology

- **Region Embedding:** Each region mapped to a 64-d learnable vector.
- **Projection to Whisper Space:** 64-d embedding is linearly projected to **1024-d** to match encoder hidden states.
- **Adapter Injection:** Projected regional vector is added to Whisper encoder outputs → gives encoder regional awareness.
- **Stabilization:** LayerNorm applied since adapter outputs start unbalanced vs pretrained encoder states.
- **Region Classifier Head:** Mean-pooling over time → single embedding → linear layer predicts region label.
- **Multi-task Loss:**

$$\text{Total Loss} = \text{Loss}_{\mathrm{ASR}} + \alpha \cdot \text{Loss}_{\mathrm{Region}}$$

Regional Classifier Adapter

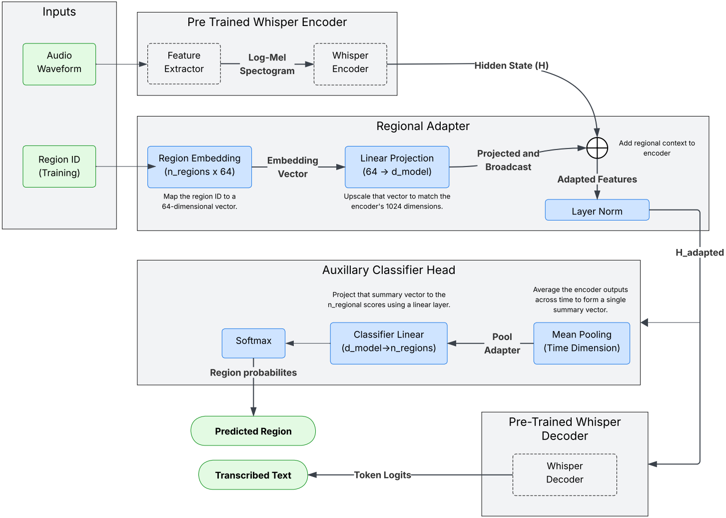


Figure: Regional Classifier Adapter architecture for Whisper Medium

Regional Classifier Adapter

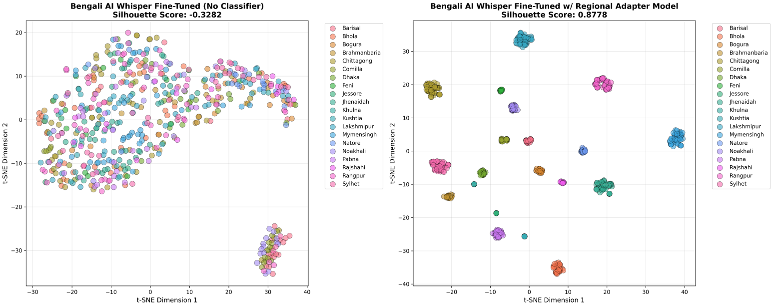


Figure: Regional Classifier Adapter showing t-SNE embeddings before and after adaptation

Full Fine-Tuning Approaches

Full Fine-Tuning (Standard Bangla)

- BengaliAI Whisper ASR Medium
- Pre-trained: Standard Bangla audio → Standard Bangla text

Full Fine-Tuning (Regional Bangla)

- BengaliAI Whisper ASR Regional Medium
- Pre-trained: Regional Bangla audio → Regional Bangla text (Ben10)

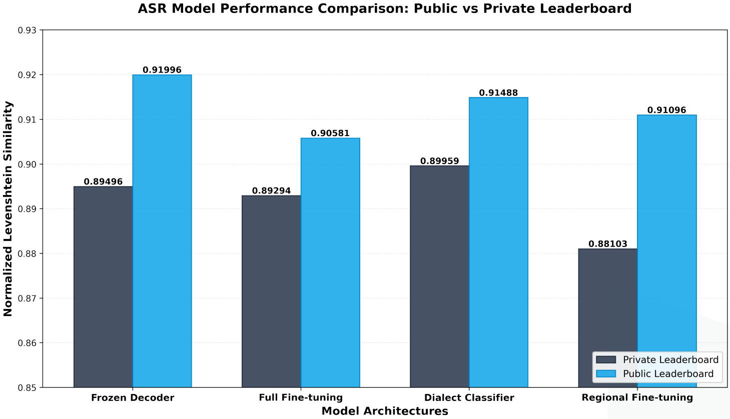
Training Approach: Full fine-tuning on competition dataset with external data, augmentation techniques, and weighted sampling for dialect balance

Evaluation Metrics & Results

Definition

Normalized Levenshtein Similarity (NLS)

$$\mathrm{NLS}(r, p) = 1 - \frac{\mathrm{LevenshteinDistance}(r, p)}{\max(|r|, |p|)}$$



Ensemble of Models

- **Method:** Recognizer Output Voting Error Reduction (ROVER)
- **Approach:** Weighted voting selecting prediction most similar to all others
- **Models:** 4 variants of fine-tuned Whisper Medium

Ensemble Results

Leaderboard	NLS Score
Public Test	0.93509
Private Test	0.91782

Model Performance Analysis

t-SNE Embedding Analysis: Errors scattered randomly without clustering, indicating no systematic failure patterns or acoustic confusion regions

Common Error Patterns: Model occasionally struggles with OOV (out-of-vocabulary) dialectal words, leading to spelling mistakes and mishearings

Deployment & Future Directions

Deployment: All model variants deployed and publicly available on Kaggle for community use and reproducibility

Future Directions

- **Expand Dataset:** Collect more regional dialectal data to address low-resource language challenges
- **Address Imbalances:** Resolve dialect and gender imbalances through targeted data collection and balanced sampling strategies to prevent model bias
- **ASR-LLM Projection Coupling:** Explore synchronous ASR-LLM integration using lightweight projection layers to transfer acoustic information from ASR decoder states to LLM, enabling acoustically-grounded text generation without full model retraining

Thank You!

Any Questions?

Team DejaView