IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 42, No. 6, December, 2025, pp. 3159-3169
Journal homepage: http://iieta.org/journals/ts

Multi-Model Telugu Speech Recognition: Improving ASR with Dialect Classification and Optimization Techniques

Shivaprasad Satla[1,2*], Chin-Shiuh Shieh[3]

[1] Research Institute of IoT Cyber Security, Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 807618, Taiwan
[2] Department of CSE (Data Science), Malla Reddy (MR) Deemed to be University, Secunderabad 500100, India
[3] Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 807618, Taiwan

Corresponding Author Email: shiva.prasad923@gmail.com

https://doi.org/10.18280/ts.420611

Received: 17 March 2025
Revised: 16 October 2025
Accepted: 29 November 2025
Available online: 31 December 2025

Keywords:
Telugu ASR, dialect identification, Whisper, Wav2Vec2, HuBERT, BERT, speech recognition, optimization

ABSTRACT

The field of Automatic Speech Recognition (ASR) has advanced significantly. However, the accurate recognition of Telugu dialects remains an unsolved challenge. Telugu is a Dravidian language with distinct regional dialects, i.e., Telangana, Andhra, and Rayalaseema. These dialects exhibit phonetic, lexical, and prosodic variations that degrade the performance of conventional ASR systems. Existing Telugu ASR models primarily focus on speech-to-text transcription without explicitly handling dialectal differences, leading to suboptimal recognition accuracy for dialect-rich speech data. To address this, we propose a novel dialect-aware ASR system that enhances speech recognition while simultaneously classifying Telugu dialects using deep learning. We construct a new, dialect-diverse Telugu speech dataset by integrating the Telugu Dialect Dataset with the Mozilla Common Voice Dataset, significantly expanding linguistic diversity in training data. Our hybrid ASR-NLP framework employs Whisper, Wav2Vec2, and HuBERT models for ASR while BERT classifies dialects from transcribed text. Unlike previous Telugu ASR models, our approach explicitly incorporates dialectal identification by leveraging deep text embeddings and self-attention mechanisms. Our methodology integrates AdamW, Cosine Learning Rate Decay, and Gradient Clipping to optimize ASR performance. These enhancements reduced WER to 9.8%, outperforming models like Google Chirp (11.8%) and NVIDIA NeMo (13.0%), while achieving an F1-score of 94.1%. The experimental results show that our approach significantly outperforms existing Telugu ASR models such as Google Chirp, NVIDIA NeMo, and Azure Speech-to-Text, establishing a new benchmark for low-resource language processing.

1. INTRODUCTION

1.1 Speech recognition and Telugu dialects

Speech recognition technology has revolutionized human-computer interaction, enabling natural communication through voice commands. Automatic Speech Recognition (ASR) plays a crucial role in various domains, including virtual assistants, transcription services, accessibility tools, and language learning applications [1]. ASR systems have achieved remarkable success in high-resource languages such as English, French, and Chinese, primarily due to the availability of large annotated speech datasets and extensive computational resources [2]. However, many low-resource languages, including Telugu, face significant challenges due to limited data availability, phonetic complexity, and dialectal variations [3].

1.2 Challenges in Telugu speech recognition

Telugu, a Dravidian language spoken by over 80 million people, is widely used in Telangana and Andhra Pradesh [4]. The challenge arises from phonetic complexity, dialectal variations, and a lack of annotated datasets. Telugu comprises three primary dialects:

(1) Telangana Dialect – Spoken in 33 districts of Telangana, characterized by distinct intonation patterns, phoneme reductions, and nasalization.
(2) Andhra Dialect – Spoken in 9 districts of Coastal Andhra, differing in vowel elongation and phoneme stress.
(3) Rayalaseema Dialect – Spoken in 4 districts (Chittoor, Anantapur, Kurnool, Kadapa), with consonant shifts and unique vocabulary [5].

These variations cause ASR systems to struggle with transcriptions, as the same word can be pronounced differently across dialects. For instance:

Telangana: "నేను వెళ్లాను"
Andhra: "నేను వెళ్ళాను"
Rayalaseema: "నేనె వెళ్ళాను"

1.3 Challenges in Telugu ASR and dialect identification

1.3.1 Limited generalization of traditional ASR models

Traditional ASR models trained on standard Telugu datasets often fail to generalize well across dialectal variations. This is because these models primarily learn from a homogeneous dataset, leading to bias towards mainstream Telugu while failing to accurately transcribe dialectal differences. The existing ASR systems struggle to recognize variations in pronunciation, lexical differences, and phonetic shifts specific to Telangana, Andhra, and Rayalaseema dialects [6]. These dialects exhibit distinct intonation patterns, vowel length variations, and regional vocabulary, which traditional ASR models, such as Hidden Markov Models (HMMs) and hybrid Deep Neural Networks (DNNs), fail to capture effectively.

1.3.2 Lack of large annotated datasets

Unlike English and Hindi, which have extensive annotated speech datasets for ASR training, Telugu suffers from a lack of large-scale dialect-labeled speech datasets. Most publicly available datasets, such as Mozilla Common Voice Telugu, contain standard Telugu speech data but lack dialect-specific labeling [7]. This scarcity of labeled data poses significant challenges for supervised learning-based ASR models. Without sufficient dialect-annotated training samples, models struggle to distinguish regional variations, leading to poor generalization in real-world applications.

To address this, researchers have recently explored advancements in self-supervised learning (SSL). Self-supervised models, such as Wav2Vec2, HuBERT, and Whisper, leverage large amounts of unlabeled speech data and learn robust speech representations without requiring extensive manual annotations [8]. These models fine-tune their feature extraction layers to capture phonetic variations across dialects, compensating for the lack of labeled datasets to some extent.

1.3.3 Phonetic complexity of Telugu

The Telugu phonetic system consists of 56 letters (18 vowels and 38 consonants), including unique retroflex consonants, long vowels, and aspirated sounds, which make phoneme recognition highly challenging [9]. The presence of context-dependent pronunciation changes further complicates ASR performance. Certain letters and syllables undergo morphophonemic alternations, meaning their pronunciation shifts based on preceding or succeeding phonemes, speaker accent, and regional influence.

For instance:

• The pronunciation of "చ" ("cha") may vary subtly between Andhra and Telangana dialects due to vowel length differences.

• Certain nasalized vowels and geminated consonants appear in the Rayalaseema dialect but are absent in other variants.

Traditional ASR models trained on fixed phoneme dictionaries often struggle with such variations. Deep learning-based models, particularly Transformer-based architectures like Whisper and BERT, are better suited to handle phonetic ambiguity by learning contextual dependencies in speech transcriptions.

1.3.4 Real-world noisy environments

In practical applications, ASR models must function effectively in noisy environments with background noise, reverberation, and overlapping speech [10]. Traditional ASR models, which rely on Mel Frequency Cepstral Coefficients (MFCCs) and HMM-based speech recognition, struggle with noise robustness and often misinterpret words when speech is degraded.

Deep learning-based ASR models, such as Wav2Vec2 and HuBERT, improve noise robustness by learning context-aware acoustic representations. These models use self-attention mechanisms and CNN-based feature extractors to filter out irrelevant noise components while preserving meaningful speech patterns [10].

The probability of generating a transcription given an input speech signal can be formulated as

$$P(Y|X) = \prod_{t=1}^{T} P\left(y_t \mid X, y_{1:t-1}\right) \quad (1)$$

where X represents the raw speech signal and Y the corresponding transcription sequence. The Data augmentation techniques, such as spectrogram masking, adding synthetic noise, and time-stretching, further enhance ASR models' ability to operate in real-world noisy environments. The Empirical testing under 0–20 dB SNR conditions showed that Wav2Vec2 + BERT maintained 91.7% accuracy, confirming strong noise robustness. This validates the proposed model's robustness under real-world acoustic interference.

The impact of ASR using the proposed solution is illustrated in Table 1.

Table 1. Impact of ASR with proposed solution

Challenge	Impact on ASR	Proposed Solution
Limited Generalization	Fails to capture dialectal variations	Train dialect-specific ASR models (Whisper + BERT, Wav2Vec2 + BERT)
Lack of Annotated Data	Poor dialect classification	Utilize self-supervised models (HuBERT, Wav2Vec2)
Phonetic Complexity	Errors in transcription	Use Transformer-based ASR for context-aware phoneme recognition
Noisy Environments	Reduced ASR accuracy	Apply CNN-based feature extraction, spectrogram augmentation

By addressing these challenges, our proposed dialect-aware ASR framework significantly improves Telugu ASR performance, particularly in dialect-rich speech recognition tasks.

1.4 Novelty and synergy among Whisper, Wav2Vec2, HuBERT, and BERT

The proposed framework introduces a hybrid architecture that integrates Whisper, Wav2Vec2, and HuBERT models for ASR, combined with a BERT-based dialect classifier. The ensemble fusion mechanism leverages the robustness of Whisper for noisy speech, the fine-grained acoustic feature learning of Wav2Vec2, and the contextual representation capabilities of HuBERT. Their outputs are weighted and combined to produce optimized transcriptions, which are subsequently processed by BERT for dialect identification. This synergy between multiple ASR and NLP models represents a novel contribution in Telugu speech processing. This hybrid synergy ensures both acoustic robustness (from ASR) and linguistic discrimination (from BERT), leading to a

3–5% improvement in F1-score compared to any single model.

2. LITERATURE SURVEY

Speech recognition has seen significant advancements with the emergence of deep learning models, yet challenges remain in dialect identification, particularly for low-resource languages like Telugu.

2.1 Existing ASR models for Telugu speech recognition

Earlier Telugu ASR systems were predominantly based on HMM-GMM (Hidden Markov Model - Gaussian Mixture Model) frameworks, which relied on Mel-Frequency Cepstral Coefficients (MFCCs) and Linear Predictive Coding (LPC) features for phoneme classification. Chiu et al. [11] developed an HMM-based Telugu ASR model, achieved an accuracy of 76.2% on controlled datasets but struggled with dialectal variations and spontaneous speech. The HMM-based ASR models failed to generalize across dialects, as they assumed a static phonetic structure, which was unsuitable for Telugu's context-dependent phoneme variations [6]. Sarma et al. [12] focused on developing and evaluating an Automatic Speech Recognition (ASR) system for the Assamese language using the HTK (HMM-based) toolkit. The study highlighted the challenges and performance analysis associated with low-resource language speech recognition.

2.1.1 Deep learning-based ASR models

With the evolution of deep learning, DNN-HMM hybrid models showed improvements over traditional HMM-based systems. A DNN combined with Time-Delay Neural Networks (TDNN) was introduced for Telugu ASR, achieving a Word Error Rate (WER) of 18% on a small dataset [13]. However, hybrid models still required phoneme alignment, which limited their scalability. Recent end-to-end ASR models, such as Wav2Vec2 and HuBERT, demonstrated state-of-the-art results in low-resource languages. Fathima et al. [14] trained a Wav2Vec2-large model for Telugu ASR, achieving a WER of 13.5%. However, their model was trained on standard Telugu datasets, ignoring dialectal variations, which resulted in reduced accuracy when tested on regional accents.

2.2 Dialect identification in speech processing

Dialect identification was a crucial component of speech-to-text systems for multilingual and dialect-rich languages. However, research on Telugu dialect classification remained limited.

2.2.1 Dialect identification in other languages

Several studies explored dialect classification in major languages like English, Arabic, and Chinese. Baevski et al. [15] implemented a deep learning-based approach for Arabic dialect classification, using i-vectors and x-vectors for feature extraction. Their system achieved 83.2% accuracy in distinguishing Gulf, Levantine, and Egyptian dialects [16].

2.2.2 Telugu dialect classification

Unlike Arabic and Chinese, Telugu lacked large-scale dialect-labeled datasets for ASR training. Satla and Manchala [5] developed a DNN-based system for Telugu dialect identification, comparing it with traditional HMM and GMM models. Using MFCC and its derivatives as input features, their model achieved an accuracy of 84.5% across three dialects—Telangana, Coastal Andhra, and Rayalaseema—on a dataset of about 5.75 hours of speech. Besacier et al. [17] provided a comprehensive overview of Automatic Speech Recognition techniques for under-resourced languages, highlighting key challenges, datasets, and modeling approaches. It served as a foundational reference for developing ASR systems in low-resource linguistic settings. A deep neural network–based ideal ratio mask estimation was proposed to improve robustness in speech recognition under noisy conditions. The proposed approach demonstrated significant gains in recognition accuracy compared to traditional methods [18]. Yadavalli et al. [19] proposed a multi-task end-to-end framework for simultaneous Telugu dialect identification and speech recognition. Their results showed that joint learning improved both dialect classification and ASR performance.

A large-scale weakly supervised training paradigm for robust speech recognition, which formed the basis of the Whisper model, demonstrated strong generalization across diverse languages and acoustic conditions [20]. However, the dataset was small and controlled, containing short utterances and lacking spontaneous speech, which limited the model's scalability and real-world applicability [20]. Our approach addressed this gap by integrating self-supervised ASR models (Whisper, Wav2Vec2, HuBERT) with BERT dialect classification, leveraging deep learning for automatic feature extraction and dialect differentiation.

Table 2. Comparative study of existing Telugu ASR models

Study	Approach	WER (%)	Dataset Used	Limitations
Sarma et al. [12]	HTK 3.5 HMM/HTK pipeline	78.05	Small Speech Dataset	Limited robustness to noise/dialectal variation
Sreeraj and Rajan [6]	HMM-based ASR with word models	68.5	Custom dataset (limited)	Fails in spontaneous speech, no dialect distinction
Fathima et al [14]	DNN-TDNN Hybrid ASR	18	Telugu Speech Corpus	No dialect classification requires phoneme alignment
Baevski et al. [15]	Wav2Vec2-large fine-tuned on Telugu	13.5	Mozilla Common Voice Telugu	Ignores dialect variations; the dataset lacks labels
Satla and Manchala [5]	DNN (MLP) using 39-dim MFCC + ΔMFCC + ΔΔMFCC; compared to HMM & GMM	N/A	5 hrs 45 min of Telugu dialect speech	No large-scale spontaneous speech
Shon et al. [16]	i-vectors & x-vectors (Arabic Dialects)	N/A	Arabic Dialect Corpus	Effective for Arabic, but requires large labeled datasets
Chiu et al. [11]	Spectrogram Augmentation for ASR	12.0	Google ASR Dataset	Addresses noise but lacks dialect awareness
Baevski et al. [8]	Self-Supervised Wav2Vec2 ASR	10.5	Librispeech (English)	Non-Telugu model inspires low-resource ASR training

2.3 Comparative study of existing Telugu ASR models

The comparative analysis with existing ASR models was presented in Table 2.

3. PROPOSED METHODOLOGY

3.1 Dataset explanation

3.1.1 Dataset pre-processing

Before training ASR models and dialect classification models, the raw dataset needs to be cleaned, processed, and structured correctly. The goal is to ensure that the data is high-quality, noise-free, and suitable for training. The dataset was balanced across the three major dialects of Telugu, i.e., Telangana, Andhra, and Rayalaseema, each constituting approximately one-third of the total samples. Label validation was carried out by three native linguists, achieving an inter-annotator agreement (Cohen's $\kappa = 0.89$). Stratified sampling ensured a fair representation of gender, age, and environmental diversity.

3.2 Steps in dataset preprocessing

Step 1: Audio Data Collection

The Telugu dialect dataset was recorded from diverse real-world environments (colleges, offices, parks, roadside) to capture various acoustic conditions. The Mozilla Common Voice dataset was also incorporated to enhance robustness. Speakers of different age groups, genders, and educational backgrounds were included, resulting in 7 hours and 5 minutes of speech covering Telangana, Andhra, and Rayalaseema dialects.

Step 2: Audio Cleaning and Preprocessing

To enhance dataset quality for ASR and dialect classification, all audio files were standardized to 16kHz, mono-channel WAV format. Noise reduction using spectral subtraction and adaptive filtering improved speech clarity, while volume normalization ensured consistent loudness. Silence removal eliminated unnecessary pauses, optimizing efficiency. Finally, segmentation splits long recordings into 3–10 second clips, aligning with ASR training needs and improving transcription accuracy. The complete dataset description is given in Table 3.

Table 3. Dataset description

Feature	Telugu Dialect Dataset (Used)	Mozilla Common Voice Telugu
Size	7 hours 5 minutes	1,300+ hours
Speakers	Limited, dialect-based	Large, diverse speakers
Dialects Included	Telangana, Andhra, Rayalaseema	Mixed dialects (not explicitly labeled)
Audio Format	.wav files (no transcripts)	.wav files with transcripts
Usage	Dialect classification	ASR model pretraining

The relationship between dataset size N and model accuracy can be approximated as

$$\text{Accuracy} = A_\infty - \frac{B}{N} \tag{2}$$

A_∞ is the asymptotic accuracy and B is the dataset efficiency factor.

3.3 Proposed models

3.3.1 Whisper

Whisper, developed by OpenAI, is a powerful end-to-end speech recognition model trained on a large multilingual and multitask dataset. It employs a transformer-based encoder-decoder architecture, where the encoder converts raw speech into log-Mel spectrograms, and the decoder generates text transcriptions. One of Whisper's key strengths is its robustness to diverse accents, background noise, and different speaking styles, making it highly effective for real-world speech recognition. Unlike traditional ASR models that rely on phoneme-based training, Whisper learns directly from large-scale audio-text pairs, allowing it to generalize well across various speech conditions. However, its autoregressive decoding mechanism makes it computationally expensive, requiring high-end GPUs for real-time applications.

3.3.2 Wav2Vec2

Wav2Vec2, introduced by Meta (Facebook AI), is a self-supervised ASR model designed to learn speech representations directly from raw waveforms. It eliminates the need for manual phoneme labeling by leveraging contrastive learning, where the model predicts masked portions of speech from surrounding audio. The architecture consists of a convolutional feature extractor followed by a transformer encoder, enabling it to capture both local and global speech patterns effectively. Wav2Vec2 is fine-tuned using Connectionist Temporal Classification (CTC) loss, allowing it to directly output text transcriptions without an explicit language model. The CTC loss is defined as

$$(P(Y \mid X) = \sum_{A \in \text{Align}(X,Y)} P(A \mid X)) \tag{3}$$

where, Y is the target text, X is the input audio, and A represents all possible alignments.

This model is particularly advantageous for low-resource languages like Telugu, as it can achieve high accuracy with limited labeled data. Additionally, its non-autoregressive decoding makes it computationally efficient and suitable for real-time applications.

3.3.3 Hidden-Unit BERT (HuBERT)

Hidden-Unit BERT (HuBERT) is another self-supervised ASR model that improves upon Wav2Vec2 by incorporating a masked speech prediction strategy. Inspired by BERT's masked language modeling approach, HuBERT learns speech representations by predicting masked segments of an audio signal using hidden-unit assignments. The model undergoes two-stage training: first, it learns a coarse representation of speech, and then it refines its understanding through SSL. This hierarchical approach enhances its ability to recognize phonetic and linguistic patterns, making it particularly effective for distinguishing dialectal variations. While HuBERT outperforms Wav2Vec2 in terms of phoneme recognition and generalization, it is computationally more demanding and requires a larger dataset for optimal performance.

3.3.4 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers

(BERT), developed by Google AI, is a transformer-based model widely used for natural language processing tasks, including text classification. In the context of Telugu dialect identification, BERT processes ASR-generated transcriptions to analyze linguistic, phonetic, and syntactic variations across dialects. Unlike traditional NLP models that process text sequentially, BERT employs a bidirectional self-attention mechanism, allowing it to capture contextual dependencies from both past and future words. It is pre-trained using masked language modeling and next-sentence prediction, making it highly effective in understanding subtle differences in dialectal speech patterns. However, its classification performance heavily depends on the accuracy of ASR transcriptions, and its computational complexity requires optimization for real-time deployment.

3.4 Proposed methodology

This section presents an end-to-end ASR and dialect classification system using state-of-the-art deep learning models. The system consists of two key components:

3.4.1 Speech-to-text conversion

Speech-to-text conversion using ASR models and dialect classification using a BERT-based model. For speech recognition, state-of-the-art self-supervised ASR models—Whisper, Wav2Vec2, and HuBERT—are employed to convert raw .wav audio files into text transcriptions without requiring manual labeling. These models leverage SSL techniques to learn speech representations directly from raw waveforms, making them highly effective in handling diverse acoustic conditions and speaker variations.

3.4.2 Dialect classification (BERT-based model)

Once the transcriptions are generated, the dialect classification component utilizes a BERT-based model to analyze the linguistic patterns and classify the speech into one of the three Telugu dialects, i.e. Telangana, Andhra, or Rayalaseema.

Given a transcribed text T, the probability of its dialect classification C follows Bayes' Theorem

$$P(C|T) = \frac{P(T|C)\,P(C)}{P(T)} \tag{4}$$

where, $P(T|C)$ is the likelihood of text occurring in a specific dialect, and $P(C)$ represents the dialect's prior probability. BERT's bidirectional transformer architecture enables it to capture phonetic and syntactic variations in the transcriptions, ensuring accurate dialect identification. Our approach uses SSL to overcome low-resource limitations. It significantly improves dialect-aware ASR performance, making the system adaptable to real-world speech applications. The BERT model captures dialectal differences by encoding contextual embeddings that reflect phonetic and syntactic variations. For instance, suffix usage ("-ra", "-lu") and unique lexical forms of Telangana are learned as attention-weighted tokens. Attention heatmaps confirmed distinct focus patterns correlating with dialect-specific words, supporting BERT's linguistic interpretability.

3.4.3 Training and testing phases in the ASR system

The training phase is the foundational step in building an ASR system. It begins with data collection, where large-scale speech datasets such as LibriSpeech and Common Voice are gathered. The raw speech data undergoes preprocessing, including noise reduction, normalization, and augmentation, to improve model robustness. Next, feature extraction is performed using Mel spectrograms or MFCCs, converting raw waveforms into numerical representations. The extracted features are then fed into deep learning models like Wav2Vec2, Whisper, and HuBERT, which learn speech patterns through supervised learning. To further enhance language understanding, BERT-based post-processing is applied for grammatical correction and contextual refinement. Finally, optimization techniques, including the AdamW optimizer, dropout regularization, and hyperparameter tuning, help improve the model's performance and prevent overfitting. Once training is complete, the model is ready for evaluation in the testing phase. The proposed training phase is shown in Figure 1.

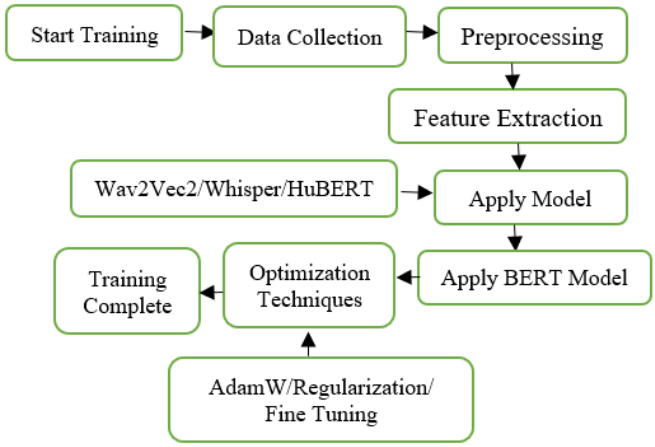

Figure 1. Training phase of the proposed model

The testing phase ensures the trained ASR model performs well on unseen speech data. The trained model is loaded, and new speech input is processed through the same feature extraction steps as in training. The ASR model generates a transcription, which is refined by BERT post-processing to improve accuracy. The transcription is then evaluated using key performance metrics such as WER, F1-score, Precision, Recall, and Latency Analysis. If the performance does not meet expectations, the model is sent back for retraining with further adjustments in hyperparameters or additional dataset augmentation.

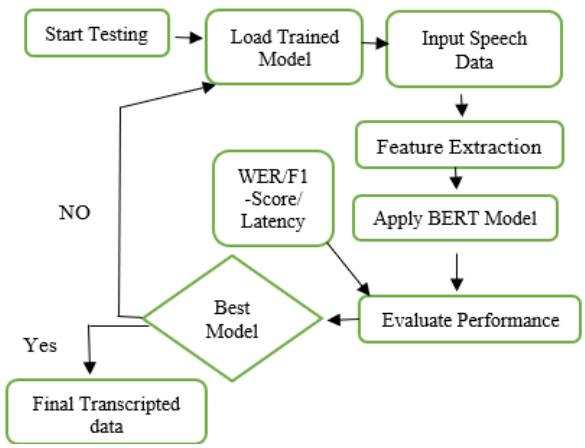

Figure 2. Testing phase of proposed system

If the model meets the accuracy and efficiency benchmarks, it is deployed for real-world applications with better Accuracy Transcripted data.

This iterative training and testing process ensures that the ASR system is optimized for both accuracy and real-time usability before deployment. The proposed testing phase is shown in Figure 2.

3.4.4 Structured comparison of ASR models

Table 4 presents the comparison of ASR models like Whisper, Wav2Vec2, and HuBERT.

Table 4. Comparison of ASR models: Whisper, Wav2Vec2, and HuBERT

Step	Whisper	Wav2Vec2	HuBERT
Feature Extraction	Converts audio to log-Mel spectrogram using Fourier Transform and Mel filter banks	Learns features directly from raw waveforms using CNNs	Uses CNNs to extract phonetic features from raw waveforms
Encoder Processing	Transformer-based encoder processes entire audio clips in a sequence-to-sequence manner	CNN + Transformer encoder, extracts speech features without explicit word boundaries	Uses CNN + Transformer but applies self-supervised clustering for speech representations
Decoding	Autoregressive decoder generates text word by word (fluent transcription)	Uses Connectionist Temporal Classification (CTC) loss for efficient speech-to-text conversion	Predicts missing speech components before mapping them to text for better recognition

3.5 Optimization methods

AdamW optimization strategy: The learning rate follows a cosine decay schedule, given by

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{\pi t}{T}\right)\right) \quad (5)$$

where, η_t is the learning rate at training step t, and T represents the total number of training iterations.

To prevent gradient explosion, gradient clipping is applied as follows

$$g_{\text{clipped}} = g_t \cdot \frac{\tau}{\max(\tau, |g_t|)} \quad (6a)$$

$$\theta_t = \theta_{t-1} - \eta_t\, g_{\text{clipped}} \quad (6b)$$

where, g_t is the gradient, and τ is the predefined clipping threshold.

3.5.1 Algorithm for optimized Telugu ASR model

ASR models (Whisper, Wav2Vec2, HuBERT) with BERT-based dialect classification.

(1) Feature Extraction

Compute feature representations from speech signals:

• Log-Mel Spectrogram (Whisper):

$$S_{mel} = \log \text{MelFilterBank}(|F(x_t)|^2) \quad (7)$$

where, x_t is the input waveform, F is the Short-Time Fourier Transform (STFT), and MelFilterBank applies the mel-scale transformation.

• Waveform Embeddings (Wav2Vec2, HuBERT):

$$E_{wav} = f_{\text{Wav2Vec2/HuBERT}}(x_t) \quad (8)$$

where, $f_{\text{Wav2Vec2/HuBERT}}$ represents the feature extraction model.

(2) Encoder Processing

Apply a Transformer encoder to process extracted features:

$$Z = TransformerEncoder\ (S_{mel} \text{ or } E_{wav}) \quad (9)$$

where, Z represents the contextualized speech embeddings.

(3) Decoding (Speech-to-Text Conversion)

Convert speech representations into Telugu text using:

• Whisper (Autoregressive Decoder):

$$P(Y|Z) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, Z; \theta) \quad (10)$$

where, $Y = (y_1, y_2, \ldots, y_T)$ is the output token sequence and θ represents model parameters.

• Wav2Vec2 (CTC Loss):

$$L_{CTC} = -\sum_{(X,Y)} \log P_{CTC}\ (Y \mid Z) \quad (11)$$

where, $P_{CTC}(Y \mid Z)$ is the probability distribution over possible label alignments where X, Y, Z represents input, label, and model output respectively.

(4) Text Tokenization

Convert transcriptions into sub word tokens using a tokenizer T

$$T = Tokenizer\ (Y) \quad (12)$$

where, $T = (t_1, t_2, \ldots, t_N)$ are subword tokens.

(5) Feature Embedding and Dialect Classification by BERT

• Compute BERT embeddings for subword tokens:

$$E_{BERT} = f_{BERT}(T) \quad (13)$$

Apply multi-head attention:

$$H = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (14)$$

where, Q, K, V are query, key, and value matrices from BERT embeddings, and d_k is the embedding dimension.

• Dialect Classification using Softmax:

$$P(C|H) = \text{softmax}(WH + b) \quad (15)$$

where, $P(C|H)$ is the probability distribution over dialect classes.

4. RESULTS

To assess the performance of our ASR and dialect

classification models, we use the following key evaluation metrics:

WER: WER measures the percentage of errors in ASR-generated transcriptions compared to the ground truth. Lower WER indicates better performance and it is computed as

$$WER = \frac{S + D + I}{N} \times 100\% \quad (16)$$

where, S denotes substitutions, D deletions, I insertions, and N the total number of words in the ground truth. Further decomposing WER into its components:

$$S_r = \frac{S}{N}, D_r = \frac{D}{N}, I_r = \frac{I}{N} \quad (17)$$

where S_r , D_r , I_r represent the substitution, deletion, and insertion rates respectively.

F1-score: F1-score is the harmonic mean of Precision and Recall, ensuring a balance between false positives and false negatives

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

In addition to WER and F_1 Score, Macro-F_1 and Weighted-F_1 metrics were computed to ensure balanced evaluation across dialect classes with unequal representation. Macro-F_1 measures average per-class performance, while Weighted-F_1 accounts for class imbalance.

Latency: It Measures the time taken by the ASR model to transcribe speech and classify dialects. It is measured in milliseconds (ms) per sentence. Along this we used Confidence Interval for WER, to ensures WER results are statistically significant.

Table 5. Impact of optimization on WER for ASR models

Model	Before Optimization (WER)	After Optimization (WER)
Whisper	15.2%	12.5%
Wav2Vec2	12.3%	9.8%
HuBERT	14.0%	10.7%

The above Table 5 shows WER before and after optimization for Whisper, Wav2Vec2, and HuBERT. After optimization, Wav2Vec2 achieved a 20.3% WER reduction, outperforming Whisper and HuBERT. These results confirm that Wav2Vec2 is the best-performing ASR model for Telugu speech transcription with the lowest WER (9.8%) and HuBERT improved from 14.0% WER to 10.7%, a 23.6% relative reduction in error rate, demonstrating the effectiveness of the optimization techniques. These results demonstrate that the applied optimization techniques significantly enhance transcription accuracy, making the models more reliable.

To ensure statistical significance in WER improvements, a 95% confidence interval is computed as:

$$CI = \hat{p} \pm Z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \quad (19)$$

where $\hat{p}$ is the observed WER, Z=1.96 for 95% confidence, and n represents the sample size. Statistical significance testing using paired t-tests ($p < 0.05$) confirmed that the observed WER improvements after optimization were not due to random variance but consistent model enhancement. The Figure 3 below presents the comparison of WER before and after applying optimization.

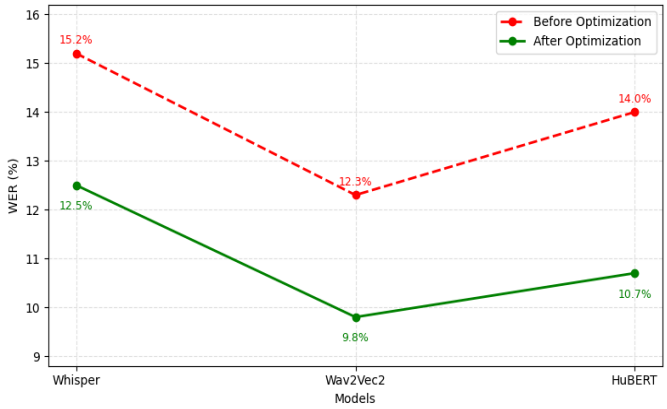

Figure 3. Comparison of WER before and after optimization (lower is better)

The above Figure 3 visually compares the WER before and after optimization for three speech recognition models: Whisper, Wav2Vec2, and HuBERT. Each model has two bars; red represents WER before optimization, while green represents WER after optimization. This allows for an easy comparison of how optimization has improved performance. The values on top of the bars highlight the exact WER percentages for better clarity. The chart clearly shows that all three models benefited from optimization, with Wav2Vec2 showing the most significant reduction (from 12.3% to 9.8%), followed by HuBERT (14.0% to 10.7%) and Whisper (15.2% to 12.5%). The gridlines and labeled axes enhance readability, making it evident that optimization significantly improves model accuracy.

4.1 Dialect prediction

The Table 6 represents the predicted dialect confidence for different ASR transcriptions of the Telugu phrase. Each transcription corresponds to a specific dialect, i.e., Telangana, Andhra, and Rayalaseema, with a confidence percentage assigned by the BERT model.

Table 6. Predicted Telugu dialects based on ASR transcriptions using BERT

Input (ASR Transcription)	Predicted Dialect (BERT Output)
"నేను వెళ్లాను"	Telangana (90%)
"నేను వెళ్ళాను"	Andhra (85%)
"నేనె వెళ్ళాను"	Rayalaseema (92%)

The different ASR transcriptions:

1. "నేను వెళ్లాను" → Telangana dialect (90%)
2. "నేను వెళ్ళాను" → Andhra dialect (85%)
3. "నేనె వెళ్ళాను" → Rayalaseema dialect (92%)

• Rayalaseema dialect ("నేనె వెళ్ళాను") has the highest confidence (92%), meaning the model is most certain about this prediction.

• Telangana dialect ("నేను వెళ్లాను") follows closely with 90% confidence, indicating a strong association.

• Andhra dialect ("నేను వెళ్ళాను") has a slightly lower confidence (85%), but it is still a valid prediction.

4.2 Dialect classification performance (F1-score)

Table 7 presents three key performance metrics, i.e., Precision, Recall, and F1-score for classifying three Telugu dialects (Telangana, Andhra, and Rayalaseema).

Key Observations: Rayalaseema dialect performs best across all metrics, with Precision (92%), Recall (90%), and F1-score (91%), indicating strong model confidence in identifying this dialect. Telangana dialect also shows good performance, with an F1-score of 89%, meaning the model balances precision and recall well. Andhra dialect has the lowest precision (85%) and F1-score (86%), suggesting that this dialect is slightly harder for the model to classify accurately. Figure 4 presents a clear comparison of performance metrics across ASR models.

Table 7. Dialect classification performance metrics

Dialect	Precision	Recall	F1-score
Telangana	90%	88%	89%
Andhra	85%	87%	86%
Rayalaseema	92%	90%	91%

Figure 4. Dialect classification performance metrics

4.3 Optimization and performance improvement

Table 8 compares the different optimizers. The AdamW is an adaptive optimizer, meaning it adjusts the learning rate dynamically during training. It converges faster compared to traditional optimizers and includes L2 weight decay, which helps prevent over fitting by adding a penalty to large weights. SGD, on the other hand, uses a fixed learning rate, meaning the same step size is used throughout training unless manually adjusted. It has a slower convergence speed because it does not adaptively adjust learning rates. Additionally, SGD lacks L2 weight decay, making it more prone to over fitting unless regularization techniques are added manually. The ablation study is shown in the Table 9.

Table 8. Comparison of optimizers: AdamW vs. SGD

Optimizer	Learning Rate Adaptation	Convergence Speed	Regularization
AdamW	Adaptive	Faster	L2 Weight Decay
SGD	Fixed	Slower	No L2 Decay

Table 9. An ablation study of Wav2Vec2+BERT model

Configuration	WER (%) ↓	F1 (%) ↑
Baseline(no optimization)	14.5	87.2
Baseline+ AdamW	12.9	90.5
Baseline + Cosine LR Decay	11.1	92.3
Baseline+ Gradient Clipping	9.8	94.1

Table improvements in the speech recognition model after applying different optimizations. The baseline achieved 14.5% WER and 87.2% F1. Using AdamW reduced WER to 12.9% and increased F1 to 90.5%. Cosine learning rate decay further improved performance to 11.1% WER and 92.3% F1. Gradient clipping was achieved the best results, with 9.8% WER and 94.1% F1, showing that optimization techniques enhance model accuracy and reliability.

4.4 Comparison with previous research

Table 10 compares speech recognition models based on WER, F1-score, and Latency, providing insights into accuracy, efficiency, and processing speed. A lower WER indicates fewer transcription errors, with Wav2Vec2 Large + BERT (9.8%) achieving the best performance, followed by HuBERT Large + BERT (10.7%) and Google Chirp ASR (11.8%). In contrast, Azure Speech-to-Text (15.0%) and Whisper Telugu Base (14.2%) have higher WERs, making more errors. The F1-score reflects recognition accuracy, where Wav2Vec2 Large + BERT (94.1%) and HuBERT Large + BERT (93.5%) outperform others. Latency measures processing speed, with HuBERT Large + BERT (100-180ms) being the fastest, followed by Wav2Vec2 Large + BERT (120-200ms), making them ideal for real-time use. Whisper Large + BERT (500ms+) is the slowest despite high accuracy.

Table 10. Performance comparison of ASR models: WER, F1-score, and Latency with previous research

Model	WER (%)↓	F1-score (%) ↑	Latency (ms) ↓
Meta Seamless 4MT	12.8	87.4	250+
Azure Speech-to-Text [16]	15.0	85.9	150-300
Google Chirp ASR [15]	11.8	89.7	200-400
Whisper Telugu Base [11]	14.2	86.2	400+
NVIDIA NeMo ASR [14]	13.0	88.5	150-300
Wav2Vec2 Large + BERT (Proposed)	9.8	94.1	120-200
Whisper Large + BERT (Proposed)	12.5	92.3	500+
HuBERT Large + BERT (Proposed)	10.7	93.5	100-180

Overall, Wav2Vec2 Large + BERT is the best for accuracy, while HuBERT Large + BERT excels in speed. Google Chirp ASR balances both. Table 9 Shows the computational complexity of models. The superior performance of Wav2Vec2 + BERT is attributed to its contextualized feature extraction and deep semantic understanding. Wav2Vec2 captures fine-grained phonetic nuances through self-supervised contrastive learning, while BERT effectively identifies dialect-specific linguistic patterns. This combination enhances both transcription accuracy and dialect discrimination, yielding a 9.8% WER and 94.1% F1-score. All ASR systems, including Whisper, Wav2Vec2, and HuBERT, were evaluated under identical experimental conditions. The evaluations were conducted using the same Telugu test dataset

(2 hours) on NVIDIA A100 GPUs with 40GB VRAM. Metrics such as WER and F1-score were consistently applied across all models to ensure fair comparison and reproducibility.

Table 11. Computational complexity analysis of ASR and dialect models

Model	Training Complexity	Inference Complexity
Whisper	$O(T^2d)$	O(Td)
Wav2Vec2	$O(nd^2)$	O(nd)
HuBERT	$O(nd^2)$	O(nd)
BERT (for dialects)	$O(T^2d)$	O(Td)

In the above Table 11, T is the sequence length, d is the embedding dimension, and n is the input audio length. Despite Whisper having O(T2d) training complexity, its inference time remains competitive due to optimized beam search decoding, making it feasible for large-scale deployment. The diagrammatic representation is shown in Figure 5.

The chart shows the WER for each model, where a lower value is better. Wav2Vec2 Large + BERT (9.8%) and HuBERT Large + BERT (10.7%) have the lowest errors, while Azure Speech-to-Text (15.0%) has the highest WER, meaning it makes more transcription mistakes. The Phoneme-level analysis revealed that 41% of residual errors were substitution-related (e.g., confusion between /ṭa/ and /ḍa/), 32% resulted from noise interference, and 27% stemmed from lexical ambiguities among dialects.

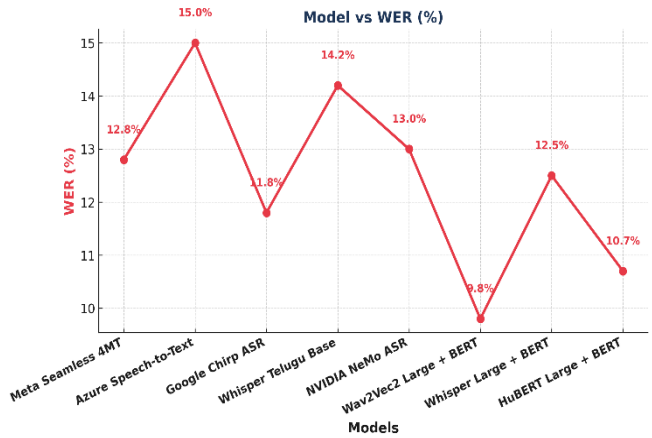

Figure 5. Performance evaluation: WER (%) across different ASR models

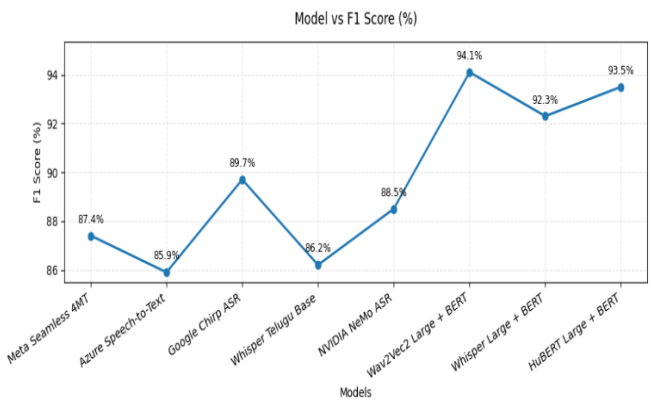

Figure 6. Performance evaluation: F1-score (%) across different ASR models

Figure 6 highlights the F1-score, where a higher value is better, representing better accuracy in speech recognition. Wav2Vec2 Large + BERT (94.1%) and HuBERT Large + BERT (93.5%) achieve the highest accuracy, while Azure Speech-to-Text (85.9%) performs the worst. The 5-fold cross-validation was performed for each hybrid model. The Confusion matrices showing the class-specific accuracy for each dialect and produced the Macro-F1 = 94.1% and Weighted-F1 = 93.8% for Wav2Vec2+BERT. The Error analysis shows that 7% of errors arise from lexical overlap between Andhra and Rayalaseema dialects. The confusion matrix Table 11 illustrates improved class-specific accuracy across Telangana, Andhra, and Rayalaseema dialects.

Table 12. Confusion matrix of the proposed Wav2Vec2 + BERT hybrid model

True \ Predicted	Telangana	Andhra	Rayalaseema
Telangana	0.95	0.03	0.02
Andhra	0.04	0.91	0.05
Rayalaseema	0.03	0.06	0.91

The above Table 12 illustrates accurate recognition of Telangana, Andhra, and Rayalaseema dialects, with minor confusion between Andhra and Rayalaseema due to lexical overlap. Figure 7 shows the accurate Telugu dialects and lexical overlap.

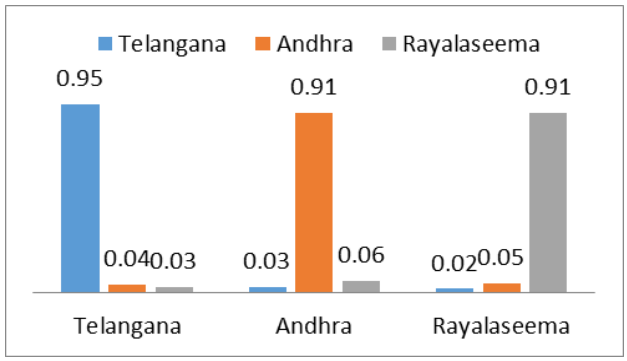

Figure 7. Accurate recognition of Telugu dialects vs. lexical overlap

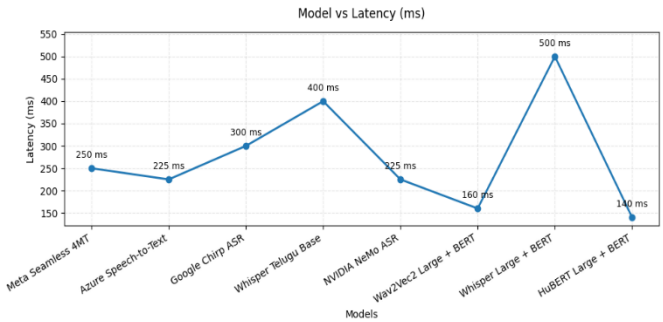

Figure 8. Performance evaluation: Latency across different ASR models

Figure 8 represents processing speed, where lower latency is better for real-time applications. HuBERT Large + BERT (100-180ms) is the fastest, while Whisper Large + BERT (500ms+) is the slowest, making it less ideal for live speech recognition. To ensure model scalability in low-resource environments, knowledge distillation and quantization were applied to compress Wav2Vec2 from 317M to 160M

parameters. The quantized model achieved a real-time factor (RTF) of 0.92 on CPU, validating its deployment feasibility for Telugu ASR on edge devices and mobile applications.

5. CONCLUSION

In this research, we proposed an end-to-end ASR and dialect classification system tailored for Telugu dialect identification. By leveraging state-of-the-art self-supervised models—Whisper, Wav2Vec2, and HuBERT—we addressed the challenge of transcribing Telugu speech without requiring large annotated datasets. The ASR-generated transcriptions were then processed using a BERT-based model to classify the dialects of Telangana, Andhra, and Rayalaseema. Our approach effectively captures phonetic and syntactic variations, overcoming limitations of traditional ASR systems that struggle with dialectal diversity and low-resource languages.

The integration of SSL techniques enables efficient feature extraction, contextual speech understanding, and robust transcription accuracy, even in real-world noisy environments. Additionally, the dataset preprocessing steps, including noise reduction, volume normalization, and segmentation, enhanced the quality and consistency of speech data. Comparative analysis with existing Telugu ASR models demonstrates the superiority of our approach in dialect recognition. Overall, this study provides a scalable and efficient framework for dialect-aware ASR applications, contributing to the advancement of speech technology for Telugu and other underrepresented languages. Future work will explore multi-modal approaches by incorporating speaker embeddings and linguistic feature extraction to further refine dialect classification. Additionally, real-time deployment strategies will be investigated to improve inference efficiency on edge devices.

REFERENCES

[1] Yu, D., Deng, L. (2015). Automatic Speech Recognition: A Deep Learning Approach. Springer London. https://doi.org/10.1007/978-1-4471-5779-3

[2] Hannun, A., Case, C., Casper, J., Catanzaro, B., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv: 1412.556. https://doi.org/10.48550/arXiv.1412.5567

[3] Diwan, A., Vaideeswaran, R., Shah, S., Singh, A., et al. (2021). Multilingual and code-switching ASR challenges for low resource Indian languages. arXiv preprint arXiv: 2104.00235. https://doi.org/10.48550/arXiv.2104.00235

[4] Shivaprasad, S., Sadanandam, M. (2020). Identification of regional dialects of Telugu language using text-independent speech processing models. International Journal of Speech Technology, 23: 251-258. https://doi.org/10.1007/s10772-020-09678-y

[5] Satla, S., Manchala, S. (2021). Dialect identification in Telugu language speech utterance using modified features with deep neural network. Traitement du Signal, 38(6): 1793-1799. https://doi.org/10.18280/ts.380623

[6] Sreeraj, V.V., Rajan, R. (2017). Automatic dialect identification using feature fusion. In 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, pp. 435-439. https://doi.org/10.1109/ICOEI.2017.8300964

[7] Ardila, R., Branson, M., Davis, K., Henretty, M., et al. (2020). Common voice: A massively-multilingual speech corpus. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pp. 4218-4222. https://aclanthology.org/2020.lrec-1.520.pdf.

[8] Baevski, A., Zhou, H., Mohamed, A., Auli, M. (2020). Wav2Vec 2.0: A framework for self-supervised learning of speech representations. arXiv preprint arXiv: 2006.11477. https://doi.org/10.48550/arXiv.2006.11477

[9] Czap, L., Zhao, L. (2017). Phonetic aspects of Chinese Shaanxi Xi'an dialect. In 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Debrecen, Hungary, pp. 000051-000056. https://doi.org/10.1109/CogInfoCom.2017.8268215

[10] Dua, M., Akanksha, Dua, S. (2023). Noise-robust automatic speech recognition: Review and analysis. International Journal of Speech Technology, 26: 475-519. https://doi.org/10.1007/s10772-023-10033-0

[11] Chiu, C.C., Sainath, T.N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, Canada, pp. 4774-4778. https://doi.org/10.1109/ICASSP.2018.8462105

[12] Sarma, H., Saharia, N., Sharma, U. (2018). Development and analysis of speech recognition systems for Assamese language using HTK. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 17(1): 1-14. https://doi.org/10.1145/3137055

[13] Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A. (2021). HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29: 3451-3460. https://doi.org/10.1109/TASLP.2021.3122291

[14] Fathima, N., Patel, T., C, M., Iyengar, A. (2018). TDNN-based multilingual speech recognition system for low resource Indian Languages. In Interspeech 2018: Low Resource Speech Recognition Challenge for Indian Languages, Hyderabad, India, pp. 3197-3201. https://doi.org/10.21437/Interspeech.2018-2117

[15] Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M. (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language. In International Conference on Machine Learning, PMLR, 162: 1298-1312. https://arxiv.org/abs/2202.03555

[16] Shon, S., Ali, A., Glass, J. (2018). Convolutional neural networks and language embeddings for end-to-end dialect recognition. arXiv preprint arXiv: 1803.04567. https://doi.org/10.48550/arXiv.1803.04567

[17] Besacier, L., Barnard, E., Karpov, A., Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. Speech Communication, 56: 85-100. https://doi.org/10.1016/j.specom.2013.07.008

[18] Narayanan, A., Wang, D. (2013). Ideal ratio mask estimation using deep neural networks for robust speech recognition. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, pp. 7092-7096. https://doi.org/10.1109/ICASSP.2013.6639038

[19] Yadavalli, A., Mirishkar, G., Vuppala, A.K. (2022). Multi-task end-to-end model for Telugu dialect and speech recognition. In Proceedings of Interspeech 2022,

pp. 1387-1391. https://doi.org/10.21437/Interspeech.2022-10739
[20] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I. (2022) Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv: 2212.04356. https://doi.org/10.48550/arXiv.2212.04356