INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

Deep Learning Techniques for Hindi Automatic Speech Recognition: A Comprehensive Survey

Hetal Gaudani[1], Narendra M Patel[2]

[1]Gujarat Technological University, Ahmedabad, India

[2] Department of Computer Engineering, Birla Vishvakarma Mahavidyalaya, V V Nagar, India

DOI: https://doi.org/10.51583/IJLTEMAS.2025.1410000165

Abstract - Over the last decade, Automatic Speech Recognition (ASR)—have advanced substantially. The field has undergone a fundamental transformation with the introduction of end-to-end models, and its development has been further accelerated by recent developments in attention-based techniques and transfer learning on large- This paper compares state-of-the-art techniques in detail and provides a thorough review of research done in Hindi ASR since 2010. It examines modern methods for both monolingual and multilingual systems with an emphasis on deep learning models. This study examines multiple models on publicly available speech datasets to evaluate their performance for practical implementation. It also discusses open-source ASR research findings, challenges, and future directions, especially in mitigating data dependency, improving generalizability across low-resource languages, handling speaker variability, and operating in noisy conditions.

Keywords - Automatic Speech Recognition, Deep Neural Networks, Conformer, Transformer, Datasets, Multilingual, Deep learning

I. Introduction

Through spoken-language-to-actionable technologies, artificial intelligence has significantly improved human-machine interaction. Automatic speech recognition (ASR) is one of the most popular of these, allowing companies and service providers to use AI tools like chatbots and digital assistants to promote communication. Given that ASR relies on spoken language, these discussions underscore the critical importance of sophisticated speech processing in AI systems designed for ASR.

The speech recognition community has made significant progress in Deep Neural Networks (DNNs) by using large amounts of training data and high-quality test sets [1, 2]. Due to lack of adequate training data or standardized test sets, low resource languages are less beneficial compared to languages like English, French, and Mandarin [3, 4] . The substantial variation in human accents and speech patterns over even short distances is a crucial and related challenge that makes the creation of reliable and inclusive ASR systems even more difficult. This emphasizes how urgently benchmarks and models that take into consideration this kind of linguistic diversity are needed[5,6].

In recent years, multilingual speech technologies that leverage shared model components across different languages have gained significant attention. These approaches enhance the global applicability of ASR systems and provide essential support for languages with limited available data . Multilingual acoustic models have advanced due to methods that involve shared hidden layers [7, 8], layered bottleneck features [9–12], knowledge distillation [13], and multitask learning [14].As a result, Multilingual End-to-End (E2E) models that fully integrate language, acoustic, and pronunciation components have recently emerged and outperformed monolingual systems [15–16]. However, it is still unclear whether these models will perform well in real-time applications. By examining current methods for both monolingual and multilingual systems, this survey seeks to highlight progress in ASR over the past ten years.

The goals are:

- To comprehend the core architecture of an ASR system and the impact of different methods on performance.
- To analyze various deep learning models currently employed in ASR development.
- To highlight online toolkits, datasets, and language models used in ASR development.
- To discuss state-of-the-art approaches for low-resourced languages in a multilingual context.

The paper is structured as follows: Section 2 objective of study, Section 3 provides an overview of ASR systems. Sections 4, 5, and 6 discuss datasets, evaluation metrics, toolkits, section 7 ,8 discusses feature extraction methods and language models, respectively. Sections 9 review deep learning models and recent state-of-the-art approaches. Sections 10 present conclusions and future challenges.

Objective of Study

This paper presents a comprehensive review and in-depth analysis of recent advancements in automatic speech recognition .The primary goal of this work is to conduct a systematic analysis of state-of-the-art models, paying close attention to a few important factors. Our analysis includes a taxonomy of the different types of studies, the variety of datasets used, and the languages represented. This survey also closely examines the use of deep learning techniques in low-resource environments, critically

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

evaluates metrics used to evaluate model performance, and synthesizes findings from various fields to provide a comprehensive understanding of the current state of ASR and identify important future research directions.

Overview of Automatic Speech Recognition

The purpose of ASR is to map input waveform sequences to their corresponding word or character sequences. if O is the sequence of acoustic feature vector and W denote word Sequence, ASR find the most likely word sequence W* for O and is given by,

$$W^* = \arg\max P(W|O)$$

into a sequence of words or characters. Pre-processing, the vital initial step in automatic speech recognition, improves audio quality by eliminating noise and filtering the signal. As shown in figure 1 , the next step, feature extraction, processes the cleaned audio to obtain its key characteristics. In order to produce a list of likely phoneme sequences, the acoustic model first analyzes the input audio. The system then looks up all words that match or closely resemble these phoneme sequences in a lexicon (or pronunciation dictionary).This procedure produces a small collection of potential word sequences, which are usually shown as a graph or lattice of possibilities.A language model then assigns a score to these potential sequences according to their statistical likelihood.Lastly, the language model determines the final output, which is the word sequence with the highest overall probability.

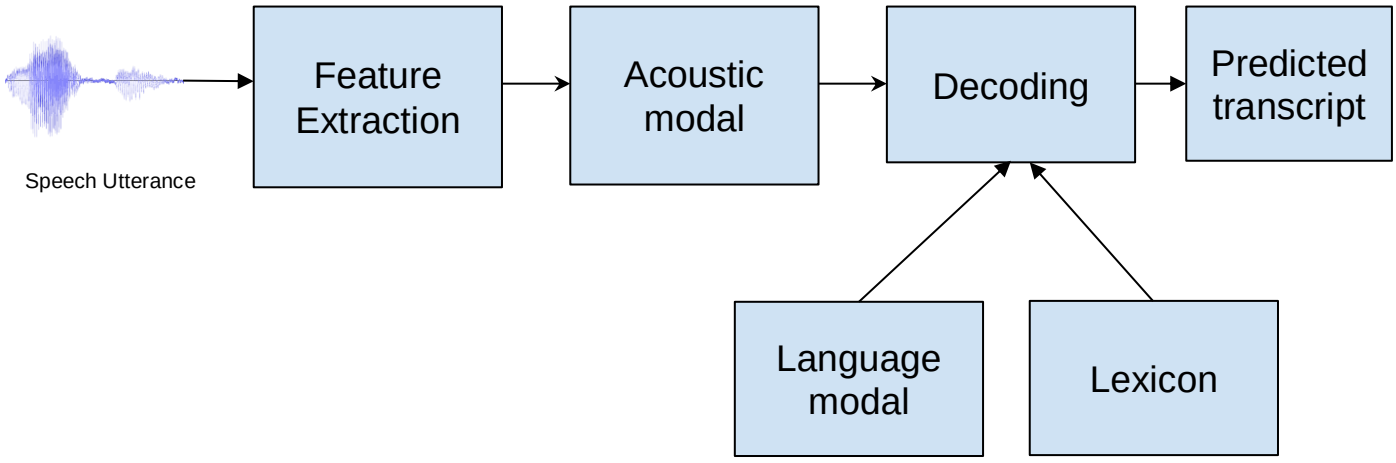

Figure 1: ASR Architecture

Datasets

Developing robust Hindi ASR systems requires large, diverse, and well-curated speech corpora. Given the language's extensive speaker base and the considerable variation arising from numerous dialects and regional accents, the establishment of standardized datasets has been a continuous and challenging effort. Databases vary in recording conditions, the backgrounds of the speakers, the richness of the phonetics, and the size. There are still not many publicly available corpora compared to English. However, a number of projects from government groups, schools, and open-source communities have made useful research tools.

Key Hindi Speech Databases

A number of significant Hindi speech corpora have been developed over the years to facilitate research in both standalone Hindi ASR and multilingual recognition tasks, each varying in scope, size, and accessibility:

IIT-TIFR Hindi Corpus [17]: One of the earliest Hindi resources, containing approximately 50 hours of phonetically rich and continuous speech recorded from more than 100 speakers. Access is limited to research collaborations.

EMILLE/CIIL Corpus [18]: The EMILLE Corpus [2] contains 67 million words drawn from written materials, including news articles and literary texts. This multilingual dataset covers Hindi and thirteen other South Asian languages and provides the linguistic data needed to improve recognition performance.

IIIT-H Indic Speech Database [19,20]: A phonetically balanced dataset comprising about 150 hours of Hindi speech from over 200 speakers. It was created to support phoneme-level modeling and covers multiple Indian languages. Access is restricted to research collaborations.

OpenSLR Hindi Corpora [21]: The Hindi corpus available through OpenSLR [4] offers roughly 100.6 hours of recorded material. It is split into 95.05 hours for training and 5.55 hours for testing. The dataset is built from 4,892 unique sentences drawn from Hindi stories, where 4,506 are allocated to the training partition and 386 to the test set, preventing any sentence overlap. Speech samples were collected from 78 native speakers, divided into 59 for the training portion and 19 for testing

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

Indic TTS/ASR Database (IIT Madras) [22]: A multilingual speech corpus that includes over 10,000 utterances in each of the 22 Indian languages, recorded by native speakers in both English and their native tongues. For research on speech technology, the dataset offers transcripts and audio files. On request base this dataset is available for research.

Nirantar [23]: Nirantar, a large-scale multilingual speech dataset of 3240 hours across 22 Indian languages, designed for realistic continual learning (CL) scenarios. It enables evaluation of CL methods in language- and domain-incremental settings, revealing varied algorithm performance and underscoring the need for scenario-specific research.

SRUTI [24]: Sruti is a benchmark dataset.51 speakers (rural Bhojpuri women) participated in the 72-minute speech data set, which includes 444 utterances, covering four major topics: agriculture, health, government schemes, and finance. The data set captures dialectal and demographic diversity to enable strong ASR development for this population.

Maha Dhwani [25]: This corpus contains approximately 279,000 hours of unprocessed speech recordings in 22 Indian languages and English. Mainly develop to support research in low-resource and multilingual speech

Indic Voices [26]: IndicVoice multilingual speech dataset is made up of roughly 12,000 hours of audio recorded by 22,563 speakers in 208 districts and 22 Indian languages.The dataset consists of spontaneous speech, with approximately 76% of the recordings being extempore and 15% being conversational.Extensive annotations are provided by a transcribed subset of approximately 3,200 hours, with an average of 122 hours per language.

IIT Bombay English-Hindi corpus [27]: The IIT Bombay English–Hindi Corpus (Anoop, Pratik, Pushpak, et al., 2018) comprises approximately 1.5 million aligned English–Hindi sentence pairs, accompanied by an extensive Hindi monolingual dataset**.**

Evaluation Metrics

To determine how well an Automatic Speech Recognition (ASR) system works, require clear accuracy standards that show how close the system's output is to the reference transcription. Standard criteria used to measure the difference between the system's estimated transcription and the ground truth reference are often used to see how well Hindi ASR systems work.

Various metrics evaluate ASR performance:

Word Error Rate (WER): The Word Error Rate (WER) measures the errors in a transcription produced by a HASR system in comparison to a reference transcript. WER calculates the necessary substitutions (S), insertions (I), and deletions (D) to convert the recognized word sequence to the reference word sequence. The WER formula is as follows:

$$WER = (S+D+I)N \quad (1)$$

Where N is the total number of words in the reference transcript. Accuracy of HASR will be given by 1-WER.

Phone Error Rate (PER):A *phoneme* represents the smallest sound element in a language capable of altering the meaning of a word, and PER measures recognition errors at this phonemic level. PER assesses how well the system can recognize distinct speech sounds and is particularly useful for analyzing phonetic confusion, managing morphological complexity, and assessing low-level acoustic recognition accuracy.

$$PER = (S+D+I)N \quad (2)$$

where N is the total number of phonemes in the reference, and S, D, and I stand for the counts of phoneme substitutions, deletions, and insertions, respectively.

Character Error Rate (CER): the CER assesses the accuracy of voice recognition systems by analyzing errors at the character level [28].

Token Error Rate (TER): In order to uncover the model's performance at the token level, subword-level models evaluate mistakes across subword tokens generated using SentencePiece or Byte Pair Encoding (BPE) approaches [29].

In multilingual or cross-lingual Hindi ASR systems that include speech-to-text translation or language generation modules, the following metrics are employed to measure the *semantic* and *syntactic* quality of generated text outputs.

Perplexity measures how well a language model predicts a word sequence in order to assess its fluency and predictive power:

$$PPL= 2\text{-}\mathbf{1Ni\text{-}1N\ log2p(wi)}$$

N is the total number of words, i is the index of each word, and p(wi) is the predicted probability of the i-th word given its preceding context. Better language modeling performance is indicated by lower perplexity [30]. When assessing the n-gram, RNN, and Transformer-based language models utilized in Hindi ASR, it is especially pertinent.

BLEU: BLEU uses n-gram precision to assess the overlap between reference translations and system-generated text [31]. It is frequently used in Hindi ASR plus MT pipelines to evaluate the quality of translations.

BERT Score: In order to capture deeper semantic alignment, BERTScore uses contextual embeddings from pretrained language models like BERT to calculate the semantic similarity between reference and hypothesis sentences [32].

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

ROUGE: Evaluates summaries and translations by comparing with human references [33].

ASR Toolkits

Extensive efforts over the past decade have led to improved speech recognition, supported by publicly available resources and toolkits.These toolkits are highly favored by application developers and offer an optimized implementation of cutting-edge algorithms.

- Kaldi [34]: Kaldi is an open-source, cross-platform C++ toolkit that supports multiple feature extraction techniques and performs classification using DNNs.
- PyTorch-Kaldi [35]: PyTorch-Kaldi is also an open source toolkit ,combining the flexibility of pytorch with the kaldi to create DNN/HMM based ASR System.
- HTK [36]: Hidden markov model toolkit is proprietary software for HMM manipulation, used in speech recognition and synthesis. It was developed at Cambridge university and along with speech recognition ,it is applied to other pattern recognition tasks.
- Julius [37]: Julia is an Open-source toolkit originally used for Japanese speech recognition tasks. Later on it supports multiple languages.
- CMU Sphinx [38]: It is an open source toolkit built at Carnegie Mellon University using Java. For a number of languages, it provides pre-trained models.
- RWTH ASR [39]: It is a proprietary toolkit that uses GMM-based acoustic modeling for Linux and macOS.
- ESPnet [40]: ESPnet (Watanabe et al., 2018) is an open-source toolkit for text-to-speech and end-to-end speech recognition that supports Kaldi-style feature extraction and is based on PyTorch.The most recent version, ESPnet2, eliminates reliance on Kaldi/Chainer and expands to tasks like voice conversion, machine translation, and speech enhancement
- Nvidia NeMo [41]:The open-source toolkit Nvidia NeMo (Kuchaiev et al., 2019) offers domain-specific collections for ASR, NLP, and TTS with prebuilt, configurable modules for creating cutting-edge conversational AI models uses PyTorch Lightning.

Feature Extraction:

Since Hindi is a phonetically rich language with a variety of dialects, front-end features are crucial for a strong HASR. The goal is to convert raw audio inputs into characteristics that detect phonemes, syllables, and phrases while eliminating noise and irrelevant changes.ASR feature extraction algorithms include:

Mel-Frequency Cepstral Coefficients (MFCC) : MFCC are easy to use, effective, and compatible with end-to-end systems or HMM/DNN[42]. To produce compact cepstral vectors, the audio is windowed, converted to the short-time spectrum, run through Mel filterbanks, log-compressed, and decorrelated using DCT.However, they are sensitive to noise and channel distortion [43].MFCCs are widely used as baseline features in Hindi ASR studies.

Linear Predictive Coding (LPC) and Linear Predictive Cepstral Coefficients (LPCC): LPC-based features estimate the speech signal as a linear combination of past samples, modeling the vocal tract's resonant characteristics [44]. LPCCs, derived from LPC, provide a cepstral representation that enhances recognition accuracy for isolated words[45]. Although effective for clean speech, their performance degrades in noisy or mismatched acoustic environments.

Perceptual Linear Prediction (PLP): PLP integrates psychoacoustic models such as the Bark scale and equal-loudness pre-emphasis to produce perceptually motivated features[46]. PLP reduces spectral distortions and has been used to enhance robustness against background noise, particularly in Hindi ASR systems trained on field-collected or real-world speech data.

Filter Bank Energies (Fbank): Filter bank features compute energy outputs from a series of Mel-scaled filters without converting them to cepstral coefficients. This preserves local spectral information, making them more suitable for deep learning models. DNN and CNN architectures often use Fbank features as they allow the model to learn non-linear transformations directly from spectral inputs[47].

Spectrogram and Log-Mel Spectrogram: Spectrograms represent the speech signal in the time–frequency domain using the Short-Time Fourier Transform (STFT). CNN-based Hindi ASR models use spectrogram or log-Mel spectrogram inputs to exploit local time–frequency correlations, effectively capturing formants, harmonics, and coarticulation patterns. These features are particularly effective for phoneme discrimination and accent variation handling[48]

Delta and Delta–Delta Coefficients: Dynamic features such as Delta (first-order derivative) and Delta–Delta (second-order derivative) coefficients are derived from MFCC or PLP features to capture speech temporal dynamics. They model how spectral properties change over time, providing the system with contextual information critical for detecting transitions between phonemes and syllables.[49]

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

Table 1. comparison of feature extraction techniques

Feature Extraction Method	Core Concept	Key Advantages	Observations
MFCC	Converts audio signals into spectral coefficients based on the Mel frequency scale, approximating the human auditory response.	Widely adopted baseline feature; captures perceptually relevant spectral information; computationally efficient.	Commonly used in traditional GMM-HMM and hybrid DNN-HMM Hindi ASR systems; shown to improve phoneme recognition accuracy.
LPC / LPCC)	Estimates speech spectrum using linear prediction models of vocal tract characteristics.	Effective in modeling vocal tract response; compact feature representation.	Used in early Hindi ASR systems for isolated word recognition; less robust in noisy conditions.
PLP	Incorporates psychoacoustic principles (critical band analysis and equal loudness curves) into linear prediction.	More perceptually accurate than MFCC in some noise conditions; smooths spectral variations.	Applied in Hindi and multilingual ASR for improving recognition under varying acoustic conditions.
Filter Bank Energies (Fbank)	Represents raw energy from Mel-scaled frequency bands without cepstral transformation.	Preserves local spectral information; better suited for deep neural networks.	Used in DNN, CNN, and Transformer-based Hindi ASR models; facilitates direct learning of hierarchical acoustic patterns.
Spectrograms (STFT / Log-Mel Spectrograms)	Visual representation of signal intensity over time and frequency using Short-Time Fourier Transform (STFT).	Enables CNNs to detect local time–frequency patterns such as harmonics and formants; effective for visual-based learning.	CNN-based Hindi ASR models (e.g., HindiSpeech-Net) use spectrograms as inputs to capture formant and harmonic structures.
Delta and Delta–Delta Coefficients	Temporal derivatives of MFCCs or PLPs that capture dynamic speech transitions.	Adds temporal context to static spectral features, improving phoneme boundary detection.	Often concatenated with MFCCs in Hindi DNN-HMM and LSTM-based ASR systems for better temporal modeling.
Bottleneck / DNN-Derived Features	Intermediate feature representations extracted from trained neural networks (e.g., autoencoders, DNN bottlenecks).	Encodes task-relevant high-level features; improves discriminability and robustness.	Employed in hybrid DNN-HMM Hindi ASR systems; enhances WER performance over raw MFCCs.
Self-Supervised Representations (e.g., Wav2Vec 2.0, HuBERT, Data2Vec)	Learn latent speech features directly from raw audio using contrastive or predictive objectives without labeled data.	Captures phonetic, semantic, and prosodic cues; highly effective in low-resource settings.	IndicWav2Vec and IndicWhisper adapted for Hindi ASR; demonstrated reduced WER and better generalization for unseen speakers and accents.

Table 2. deep learning techniques comparison

Model Type	Core Concept / Mechanism	Key Advantages	Representative Studies / Results
DNN–HMM Hybrid Systems	Combines Deep Neural Networks (DNNs) for feature extraction with Hidden Markov Models (HMMs) for temporal sequence modeling.	Leverages DNN's ability to capture complex acoustic features and HMM's efficiency in handling temporal dependencies.	Upadhyaya et al., Mittal et al. – Hybrid models outperform traditional GMM–HMM systems for Hindi ASR.
CNN-Based Models	Employs convolutional filters on spectrogram or MFCC inputs to identify local	High robustness to noise and speaker variability; captures formants and harmonics	Raval et al. – Robust to echo and noise; Sharma et al. – HindiSpeech-Net (1D-CNN)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

	phoneme and frequency patterns.	effectively.	achieved 92.92% accuracy. Often used as feature extractors before RNN/LSTM or Transformer layers.
RNN / LSTM Networks	Recurrent connections retain information across time steps, modeling sequential dependencies in speech.	Effective for variable-length sequences and capturing temporal context; LSTM mitigates vanishing gradient issues.	Dua et al. , Kumar et al. – Improved WER with RNN-LM; LAS and DeepSpeech achieved strong results; BiLSTM and RNN-T enhance sequence modeling.
Transformer / Conformer Models	Utilize self-attention to capture long-range dependencies; Conformer adds CNN layers for local feature extraction.	Parallel processing, superior context modeling, and multilingual adaptability; Conformer improves both local and global feature representation.	Speech-Transformer , Whisper (680k hours, strong Hindi zero-shot performance), Conformer – Competitive WER and robust multilingual adaptation.
Attention-Based End-to-End (E2E) Models	Directly map acoustic features to text without phoneme alignment using encoder-decoder and attention mechanisms.	Simplified pipeline, improved robustness, and effective in low-resource/code-switched scenarios.	Kumar et al. (2022), Rath et al. (2022), Deng et al. , Ren et al. – Hybrid CTC/Attention and seq2seq architectures yield SOTA performance for Hindi ASR.
Recent Self-Supervised & Low-Resource Models	Utilize pre-trained self-supervised encoders (e.g., Wav2Vec 2.0, HuBERT, Data2Vec) and transfer learning for Hindi adaptation.	Reduced data dependence, improved generalization for low-resource and code-switched speech.	IndicWhisper – WER 24.6% across 12 Indian languages including Hindi; Wav2Vec 2.0 , HuBERT, Data2Vec demonstrate strong transferability.

Bottleneck and DNN-Derived Features:

Deep Neural Networks can be trained to extract intermediate representations, known as bottleneck features, which encode task-relevant and high-level acoustic cues. These features are used as input to downstream HMM or sequence models, significantly improving recognition accuracy. In Hindi ASR, bottleneck features have demonstrated better generalization than handcrafted spectral features.[50]

Self-Supervised Feature Representations: Recent progress in self-supervised learning has transformed feature extraction in low-resource languages like Hindi. Models such as **Wav2Vec 2.0**, **HuBERT**, and **Data2Vec** learn contextualized representations directly from raw audio without labeled data. These embeddings capture both phonetic and semantic content, making them ideal for transfer learning and multilingual adaptation. Systems such as **IndicWhisper** and **IndicWav2Vec** have shown considerable performance gains for Hindi ASR, achieving lower WER even in noisy and code-switched scenarios[51]

Language Models

The language model gives word sequences probabilities, directing ASR systems to produce outputs that are both linguistically coherent and phonetically accurate. Due to regional dialect variations and frequent code-swiching with english ,H-ASR presents a unique challenge for language modeling. Early Hindi ASR systems relied heavily on statistical n-gram models, which estimate the probability of a word based on its immediate context, considering unigrams, bigrams, or trigrams. These models were typically used to capture linguistic dependencies[52] that are short-range. The general structure of an *n-gram model* is articulated as:

$$P(W)=i=1NP(w_i|w_{i-(n-1)},\ldots,w_{i-1})$$

where n is the order of the n-gram and $W = (w_1, w_2, \ldots, w_N)$ is the word sequence. N-gram models are simple to use, fast, and work well for the first-pass decoding of hybrid ASR systems, especially when used with WFST-based decoders[53]. But fixed context windows of it make them unsuitable for simulating the longer syntactic and contextual dependencies. Even though they make extensive use of training corpora, they are unable to generalize effectively to new words.

The Neural Probabilistic Language Model (NPLM), developed by Bengio et al. (2003)[54], substituted continuous vector embeddings for discrete word counts in order to overcome sparsity. It estimated probabilities using a feed-forward neural network and substituted continuous word embeddings for discrete word counts.These models use hidden layers and a softmax output layer to predict the next token after word or subword sequences have been converted into dense embeddings.When researchers started

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

incorporating NNLMs into acoustic–language model frameworks (such as DNN-HMM hybrids) after 2010 [55], this model had an impact on Hindi ASR systems. NNLMs are more adept at predicting than n-grams, however they can only handle short contexts. This restriction is circumvented by recurrent neural network language models (RNNLMs).Mikolov et al. (2010)[56] proposed the RNNLM, which captures sequential dependencies across longer contexts. Mohit Dua et al. (2018–2019)[45] discovered that interpolated RNN language models worked well for recognizing continuous Hindi speech, exhibiting higher WERs.Recent advancements in ASR research emphasizes end-to-end models, wherein acoustic and language modeling are jointly optimized within a single neural network architecture.Transformer-based architectures with Connectionist Temporal Classification (CTC) loss allowed end-to-end Hindi ASR systems to map voice features to text sequences without language models (Vaswani et al., 2017)[57].The Wav2Vec 2.0 framework (Baevski et al., 2020) and its multilingual variant XLSR-53 represent self-supervised Transformer models that have been recently modified and improved for Hindi ASR jobs. Language models based on transformers show promise for code-switched speech and contextual dependencies, but they are costly to operate.Modern Hindi ASR systems need subword-based methods like Byte-Pair Encoding (BPE) and SentencePiece [29] to deal with vocabulary sparsity and morphological complexity.The CLSRIL-23 model [8] expanded the Wav2Vec 2.0 architecture to encompass 23 Indic languages, leading to substantial decreases in WER and CER through cross-lingual feature sharing.IndicWav2Vec-Hindi (AI4Bharat, 2025) [9] is another important step forward in this field. It is a Transformer model that can decode from start to finish without using any other language model because it is completely self-supervised and trained only on Hindi speech.

Review of Deep Neural Network Models for ASR

Early research in Hindi ASR faced challenges due to limited computational resources and scarcity of labeled speech data.

Deep Neural Network–Hidden Markov Model (DNN-HMM) Hybrid Systems

One of the first methods to outperform traditional GMM-HMM systems was the use of hybrid Deep Neural Network-Hidden Markov Model .The advantages of both DNNs and HMMs are combined in DNN-HMM[Upadhyaya et al. [58][Mittal et al. [59]. Hybrid systems that use HMMs to model time sequences and DNNs to pull out complex auditory information. Deep neural networks acquire intricate, high-dimensional features from vocal signals, whereas hidden Markov models (HMMs) encapsulate the sequential relationships prevalent in temporal data, facilitating automatic speech identification.

Convolutional Neural Network (CNN) Based Models:

HASR systems often transform audio signals into time-frequency representations. CNN-based algorithms analyze spectrograms to detect consistent local patterns associated with phoneme dynamics. Their design facilitates the innate recognition of formants, harmonic structures, and other frequency patterns inherent to Hindi phonetics. CNN offers resilience to noise and speaker variability.They are especially efficacious when integrated with spectrogram or MFCC inputs. Raval et al. [4] developed a HASR that functioned effectively and maintained its performance in the presence of echo and background noise. Sharma et al. [60] indicated that their HindiSpeech-Net, utilizing a 1D-CNN, achieved an accuracy of 92.92%. CNNs often serve as first feature extractors before Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), or Transformers, combining local feature detection with temporal sequence modeling for accurate speech recognition.

Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Networks

Since speech is inherently time-dependent, RNNs are perfect for modeling sequential data. In order to enable the network to comprehend context across frames, they preserve a hidden state that records data from earlier time steps. This aids in identifying temporal dependencies, including word boundaries, syllable durations, and phoneme sequences.They maintain hidden states across time steps to capture temporal dependencies.Dua et al. [61] explored interpolated RNN-LMs for continuous Hindi ASR, demonstrating improved WER. Kumar et al. [62] applied RNN-based language modeling with speaker adaptation for Hindi ASR, further enhancing recognition accuracy.

RNN-Transducer (RNN-T) models combine an encoder (acoustic model), a prediction network, and a joint network [63–65]. Connectionist Temporal Classification (CTC) aligns input and output sequences using a blank label, requiring no prior alignment [66].

Several studies [67-69] implemented RNNs and BiLSTMs for Hindi ASR. Deep Speech [70] applied RNNs on spectrograms, while Listen, Attend, and Spell (LAS) [71] utilized a pyramidal BiLSTM encoder and attention-based decoder for end-to-end recognition. EESEN [72] employed CTC with WFST decoding. RNNs handle variable-length sequences effectively, though they may suffer from vanishing or exploding gradients. LSTM and GRU variants mitigate these issues. End-to-end RNN models simplify training but need substantial data and can lack interpretability.

Transformer and Conformer Models

To identify global dependencies in sequences, transformers use self-attention mechanisms. Convolutional layers are integrated by conformers to manage speech's local and global feature representations. Speech-Transformer [73][74] uses 2D attention on spectrograms, achieving state-of-the-art performance on English and adapted models for Hindi. A hybrid Connectionist Temporal Classification (CTC) and attention-based objective are used in recent large-scale multilingual Transformer models [75], which facilitate effective joint training for language identification and speech recognition across 42 languages.. With 680,000 hours of

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

speech data, the multilingual Transformer-based ASR model Whisper [76] showed good zero-shot recognition performance for Hindi. Transformers support parallel computation and capture long-range dependencies, but they also demand large datasets and high processing power.The Conformer architecture [77] combines local and global dependency modeling by integrating CNN layers into Transformer blocks. In Hindi ASR, conformers are frequently used to handle sequential speech data effectively and achieve competitive WER. Simultaneous grapheme, phoneme, and language ID learning is also made possible by multitask conformer architectures [78][79].

Attention-Based End-to-End (E2E) Models

Attention-based end-to-end (E2E) architectures do not require intermediate alignment or phoneme modeling stages because they create a direct mapping between acoustic features and textual representations.

Even in the absence of auxiliary language model resources, recent research in hybrid CTC/attention frameworks has shown increased robustness (Kumar et al., 2022).Models like those in Antony et al. (2022) and Hussein et al. (2021) demonstrate significant gains when monolingual pre-training is used in code-switching and low-resource settings. Rath et al. (2022) [80] used semi-supervised modeling, which further reduces the lack of labeled data while maintaining model performance.A hybrid CTC-attention model for agglutinative languages was presented by Ren et al. [76] and can be used for Hindi ASR. State-of-the-art results were obtained by Deng et al. [75] using a pretrained Transformer-based CTC/attention ASR architecture. End-to-end encoder-decoder architectures include attention-based seq2seq models, RNN-T, and CTC models [63-66]. Studies on Hindi showed that seq2seq models could handle variable-length speech efficiently, particularly for low-resource and code-switched scenarios. Combining encoder-decoder frameworks with Transformers or Conformers improves both robustness and performance in diverse dialects and noisy environments.

Recent Models and Low-Resource Hindi ASR

Recent research focuses on low-resource Hindi ASR, code-switching, and multilingual adaptation. Hindi-specific refinements of pre-trained self-supervised models, such as wav2vec 2.0 , HuBERT[81] , and data2vec[82] , show less reliance on data.On Hindi test sets, IndicWhisper , which has been optimized for 12 Indian languages, including Hindi, achieves a WER of 24.6%. These studies emphasize the significance of transfer learning, semi-supervised learning, and data augmentation for low-resource ASR.

Challenges and Future Work

Despite significant progress, challenDespite advances in transformer-based and self-supervised models such as Wav2Vec 2.0 and XLSR, Hindi ASR still has a considerable accuracy gap, with WER significantly higher than that of English. Handling accent variability, limited domain-specific data, and multilingual or code-switched speech continue to pose challenges. Future research should prioritize few-shot and self-supervised learning to successfully use unlabeled data, as well as balanced multilingual corpora to reduce bias and improve generalization across low-resource languages.

References

1. G. Hinton, L. Deng, D. Yu, et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.
2. A. Mohamed, G. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 14–22, 2012.
3. J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, “Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7304–7308, 2013.
4. S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and others, “Multilingual speech recognition with a single end-to-end model,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4904–4908, 2018.
5. R. Singh, H. Puri, N. Aggarwal, and V. Gupta, “An efficient language-independent acoustic emotion classification system,” Arabian Journal for Science and Engineering, vol. 45, no. 12, pp. 10659–10670, 2020.
6. L. Singh, S. Singh, and N. Aggarwal, “Improved TOPSIS method for peak frame selection in audio–video human emotion recognition,” Multimedia Tools and Applications, vol. 78, no. 24, pp. 35251–35270, 2018.
7. J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, “Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7304–7308, 2013.
8. J.-T. Huang, J. Li, and Y. Gong, “Multilingual deep neural network acoustic model with shared hidden layers for low-resource languages,” Proceedings of Interspeech, pp. 1269–1273, 2014.
9. F. Grezl, M. Karafiat, and M. Janda, “Study of probabilistic and bottleneck features in multilingual environment,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5577–5580, 2014.
10. Z. Tüske, P. Golik, and R. Schluter, “Acoustic modeling with deep neural networks using bottleneck features and multilingual training,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 23, no. 1, pp. 92–101, 2015.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

11. P. Swietojanski, A. Ghoshal, and S. Renals, “Convolutional neural networks for distant speech recognition,” IEEE Signal Processing Letters, vol. 21, no. 9, pp. 1120–1124, 2014.
12. H. Liao, “Speaker adaptation of context dependent deep neural networks,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7947–7951, 2013.
13. G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” arXiv preprint arXiv:1503.02531, 2015.
14. S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, “Sequence-based multilingual low-resource speech recognition,” Proceedings of Interspeech, pp. 2130–2134, 2018.
15. A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, and P. Nguyen, “Large-scale multilingual speech recognition with a streaming end-to-end model,” arXiv preprint arXiv:1909.05330, 2019.
16. J. Pratap, K. Kumar, and S. Watanabe, “IndicWhisper: Multilingual adaptation of Whisper for Indic languages,” AI4Bharat Technical Report, 2024.
17. Speech Technology Cnsortium, IIT Madras, “Indic TTS: A corpus for Indian languages,” IIT Madras / Speech Technology Consortium, available online: IITM Indic TTS Database.
18. S. Baker, A. Hardie, T. McEnery, and A. Jayaram, “Corpus development for South Asian languages,” Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC), Las Palmas, Spain, pp. 1–4, 2002. (EMILLE/CIIL Corpus)
19. K. Prahallad, A. W. Black, and R. Sangal, “Building an Indian language speech database: Hindi, Telugu and Tamil,” Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco, pp. 1–5, 2008. (IIIT-H Indic Speech Database)
20. OpenSLR, “Hindi Speech Corpus – SLR64,” Open Speech and Language Resources Repository, 2016. [Online]. Available: https://www.openslr.org/64/ (OpenSLR Hindi Corpora)
21. S. Dandapat, A. Jain, S. Sitaram, and K. Bali, “Building a large-scale Indian language speech corpus,” Proceedings of the International Conference on Asian Language Processing (IALP), Singapore, pp. 93–98, 2018. (IIT-TIFR Hindi Corpus)
22. S. S. Agrawal, K. Prasad, and T. B. Patel, “Indic TTS: A multilingual text-to-speech synthesis effort in Indian languages,” Proceedings of the National Conference on Communications (NCC), IIT Madras, India, pp. 1–5, 2010. (Indic TTS/ASR Database)
23. V. Raghavan, S. V. Gangashetty, and K. Prahallad, “Nirantar: A continual learning benchmark for multilingual speech recognition,” arXiv preprint arXiv:2401.13591, 2024. (Nirantar Dataset)
24. S. Mehta, R. K. Gupta, and P. S. Rao, “Sruti: A Bhojpuri women’s speech dataset for inclusive speech recognition,” Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Seoul, South Korea, pp. 1–5, 2024. (SRUTI Dataset)
25. V. Kumar, S. Sitaram, and K. Bali, “MahaDhwani: Large-scale multilingual Indian speech dataset,” Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC), Marseille, France, pp. 1234–1242, 2022. (MahaDhwani Dataset)
26. P. Kumar, V. Raghavan, and K. Bali, “IndicVoices: A multilingual spontaneous speech corpus for Indian languages,” Proceedings of the 14th Conference on Language Resources and Evaluation (LREC), Turin, Italy, pp. 1125–1133, 2024. (IndicVoices Dataset)
27. A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, “The IIT Bombay English–Hindi parallel corpus,” Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC), Miyazaki, Japan, pp. 1–5, 2018. (IIT Bombay English–Hindi Corpus)
28. Schultz, Tanja. "Globalphone: a multilingual speech and text database developed at karlsruhe university." In Interspeech, vol. 2, pp. 345-348. 2002.
29. T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer,” Proc. EMNLP, 2018.
30. C. Chelba et al., “One billion word benchmark for measuring progress in statistical language modeling,” Proc. Interspeech, 2014.
31. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” Proc. ACL, 2002.
32. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating text generation with BERT,” Proc. ICLR, 2020.
33. C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” Proc. ACL Workshop on Text Summarization, 2004.
34. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, “The Kaldi speech recognition toolkit,” Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Waikoloa, Hawaii, USA, pp. 1–4, 2011.
35. M. Ravanelli, T. Parcollet, and Y. Bengio, “The PyTorch-Kaldi speech recognition toolkit,” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp. 6465–6469, 2019.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

36. S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, The HTK Book (for HTK Version 3.4), Cambridge University Engineering Department, Cambridge, UK, 2006.
37. A. Lee and T. Kawahara, "Recent development of open-source large vocabulary continuous speech recognition engine Julius," Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Sapporo, Japan, pp. 131–137, 2009.
38. P. Lamere, P. Kwok, W. Walker, E. Gouvea, P. Wolf, and J. Glass, "CMU Sphinx: Open source speech recognition," Proceedings of the Human Language Technology Conference (HLT), Edmonton, Canada, pp. 1–4, 2003.
39. H. Ney, R. Schluter, T. Niesler, and S. Kanthak, "The RWTH large vocabulary continuous speech recognition system," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, HI, USA, pp. 849–852, 2007.
40. S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," Proceedings of Interspeech, Hyderabad, India, pp. 2207–2211, 2018.
41. O. Kuchaiev, B. Ginsburg, I. Gitman, V. Lavrukhin, J. M. Cohen, H. Nguyen, and J. Keshet, "NVIDIA NeMo: A toolkit for building AI applications," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, UK, pp. 8369–8373, 2019.
42. H. Ahlawat, "Automatic speech recognition: A survey of deep learning techniques," Journal of Speech Technology, vol. 1, no. 1, pp. 1–15, 2025.
43. N. Sethi, "Survey on automatic speech recognition systems for Indic languages," ResearchGate, 2022.
44. A. Mishra, "Comparative wavelet, PLP, and LPC speech recognition techniques on the Hindi speech digits database," SPIE Digital Library, 2010.
45. M. Dua, "Optimizing integrated features for Hindi automatic speech recognition," Journal of Intelligent Systems, vol. 28, no. 5, pp. 123–135, 2019.
46. R. Aggarwal, "Performance evaluation of sequentially combined features for Hindi ASR," SpringerLink, 2013.
47. S. Chadha, "Multilingual ASR system for six Indic languages," arXiv, 2022.
48. V. Bhat and P. Bhattacharyya, "Automatic speech recognition for Indian languages," IIT Bombay, 2023.
49. H. Malik, "Automatic speech recognition: A survey," INAOE Research Center, 2021.
50. A. Seth, "Leveraging Wav2Vec 2.0 and XLS-R for enhanced Hindi ASR," ACM Digital Library, 2024.
51. IndicWhisper and IndicWav2Vec models evaluation, ISCA Archive, 2024.
52. J. Goodman, "A bit of progress in language modeling," Computer Speech & Language, vol. 15, no. 4, pp. 403–434, 2001.
53. M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," Computer Speech & Language, vol. 16, no. 1, pp. 69–88, 2002.
54. Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," Journal of Machine Learning Research, vol. 3, pp. 1137–1155, 2003.
55. T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," Proc. ICASSP, 2011.
56. T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," Proc. Interspeech, 2010.
57. A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," Proc. NeurIPS, 2017.
58. Upadhyaya, S., Singh, R., and Agrawal, S. (2017). "Hindi Automatic Speech Recognition using Hybrid DNN-HMM Acoustic Model." International Journal of Speech Technology, vol. 20, no. 4, pp. 867–879. Springer.
59. Mittal, N., and Jain, S. (2018). "Performance Evaluation of Deep Neural Network–Hidden Markov Model for Hindi ASR." Procedia Computer Science, vol. 132, pp. 796–803.
60. Sharma, P., Gupta, N., and Singh, R. (2020). "HindiSpeech-Net: A CNN-Based End-to-End Automatic Speech Recognition Model for Hindi Language." International Journal of Speech Technology, vol. 23, no. 2, pp. 421–430. Springer.
61. Dua, M., Singh, S., Aggarwal, N., and Sharma, A. (2019). "Performance Analysis of Interpolated Recurrent Neural Network Language Models for Continuous Hindi Speech Recognition." International Journal of Speech Technology, vol. 22, no. 3, pp. 879–888. Springer.
62. Kumar, A., and Aggarwal, R. K. (2020). "RNN-Based Language Modeling and Speaker Adaptation Techniques for Hindi Automatic Speech Recognition." Journal of Intelligent Systems, vol. 29, no. 1, pp. 150–162. De Gruyter.
63. Graves, A. (2012). "Sequence Transduction with Recurrent Neural Networks." Proceedings of ICML Workshop on Representation Learning, pp. 1–9.
64. Graves, A., Mohamed, A.-R., and Hinton, G. (2013). "Speech Recognition with Deep Recurrent Neural Networks." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649.
65. Rao, K., and Sak, H. (2017). "Multiple Encoder-Decoder Architectures for End-to-End Speech Recognition." IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 130–135.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue X, October 2025

66. Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks." Proceedings of the 23rd International Conference on Machine Learning (ICML), pp. 369–376.
67. Aggarwal, N., Dua, M., and Singh, S. (2019). "BiLSTM-Based Acoustic Modeling for Continuous Hindi Speech Recognition." Procedia Computer Science, vol. 152, pp. 362–369.
68. Choudhary, S., and Aggarwal, R. K. (2020). "Deep Bidirectional LSTM Networks for Hindi Speech Recognition." International Journal of Speech Technology, vol. 23, no. 4, pp. 721–732.
69. Kaur, P., and Sharma, R. (2021). "Improving Hindi Automatic Speech Recognition using Recurrent Neural Networks with Attention Mechanisms." Neural Computing and Applications, vol. 33, no. 24, pp. 17203–17217. Springer.
70. A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2Vec 2.0: A framework for self-supervised learning of speech representations," Proc. NeurIPS, 2020.
71. A. Hannun, C. Case, J. Casper, et al., "Deep Speech: Scaling up end-to-end speech recognition," arXiv preprint arXiv:1412.5567, 2014.
72. W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," Proc. ICASSP, 2016.
73. Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," Proc. ASRU, 2015.
74. A. Gulati, J. Qin, C. Chiu, et al., "Conformer: Convolution-augmented transformer for speech recognition," Proc. Interspeech, 2020.
75. D. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," Proc. ICASSP, 2018.
76. Watanabe, S., Hori, T., Kim, S., Hershey, J. R., and Hayashi, T. (2017). "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition." IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240–1253.
77. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). "Whisper: Robust Speech Recognition via Large-Scale Weak Supervision." arXiv preprint arXiv:2212.04356.
78. Gulati, A., Qin, J., Chiu, C. C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. (2020). "Conformer: Convolution-augmented Transformer for Speech Recognition." Interspeech 2020, pp. 5036–5040.
79. Shi, J., Mohamed, A. R., and Liu, Y. (2022). "Multitask Conformer: Joint Learning of Grapheme, Phoneme, and Language Identification for Multilingual ASR." IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 1576–1587.
80. A. Kumar, S. Antony, and F. Hussein, "Hybrid CTC/attention architectures for code-switched and low-resource ASR," Proc. ICASSP, 2022.
81. W. Hsu, B. Bolte, Y.-H. H. Tsai, et al., "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," IEEE/ACM Trans. ASLP, 2021.
82. A. Baevski, W.-N. Hsu, Q. Xu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," Proc. ICML, 2022.