Egyptian Informatics Journal 29 (2025) 100615

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

End-to-end neural automatic speech recognition system for low resource languages

Sami Dhahbi [a], Nasir Saleem [b], Sami Bourouis [c], Mouhebeddine Berrima [d,*], Elena Verdú [e]

[a] Applied College of Mahail Aseer, King Khalid University, Muhayil Aseer, 62529, Saudi Arabia
[b] Department of Electrical Engineering, Faculty of Engineering and Technology, Gomal University, Dera Ismail Khan, Pakistan
[c] Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
[d] Unit of Scientific Research, Applied College, Qassim University, Buraydah, Saudi Arabia
[e] Universidad Internacional de La Rioja, Logroño, Spain

ARTICLE INFO

Keywords:
Speech recognition
Deep learning
Low-resource language
E2E learning
Data augmentation
Synthetic speech

ABSTRACT

The rising popularity of end-to-end (E2E) automatic speech recognition (ASR) systems can be attributed to their ability to learn complex speech patterns directly from raw data, eliminating the need for intricate feature extraction pipelines and handcrafted language models. E2E-ASR systems have consistently outperformed traditional ASRs. However, training E2E-ASR systems for low-resource languages remains challenging due to the dependence on data from well-resourced languages. ASR is vital for promoting under-resourced languages, especially in developing human-to-human and human-to-machine communication systems. Using synthetic speech and data augmentation techniques can enhance E2E-ASR performance for low-resource languages, reducing word error rates (WERs) and character error rates (CERs). This study leverages a non-autoregressive neural text-to-speech (TTS) engine to generate high-quality speech, converting a series of phonemes into speech waveforms (mel-spectrograms). An on-the-fly data augmentation method is applied to these mel-spectrograms, treating them as images from which features are extracted to train a convolutional neural network (CNN) and a bidirectional long short-term memory (BLSTM)-based ASR. The E2E architecture of this system achieves optimal WER and CER performance. The proposed deep learning-based E2E-ASR, trained with synthetic speech and data augmentation, shows significant performance improvements, with a 20.75% reduction in WERs and a 10.34% reduction in CERs.

1. Introduction

Automatic speech recognition (ASR) is a part of daily life, with hands-free assisting ASR systems such as Apple CarPlay in cars, Apple Siri and Samsung Bixby in cell phones, Microsoft Cortana in computers, and Amazon Echo and Google Home in virtual home assistants. However, most of these technologies are designed for languages with abundant digital resources. Low-resource languages, facing constraints such as a scarcity of large training data and phonetic definitions and language models, encounter difficulties in implementing such advanced technologies. Additionally, low-resource languages contend with issues like diverse language types, a shortage of skilled native speakers for data collection, and numerous dialects. While ASR can facilitate interaction in specific circumstances [1,2], its importance transcends mere assistance and becomes crucial. For instance, in regions with low literacy rates and limitations associated with under-resourced languages, ASR proves to be an ideal solution [3–6]. The widespread adoption of communication platforms, such as smartphones, in the developing world and their increasing presence in rural areas offers a unique opportunity to create a voice-based application. Such an application could play a crucial role in addressing the low literacy levels prevalent in these regions. Individuals in areas with limited literacy often communicate in local languages, sometimes referred to as under-resourced due to their lack of formal written grammar and vocabulary. Many people in these regions may not comprehend languages with abundant resources, such as English. Developing ASR systems for languages with fewer resources appears to be a promising solution to this challenge. However, creating an ASR system faces the obstacle of limited training data. This study focuses on end-to-end ASR for low-resource languages. For instance, Urdu, despite being the 10th most spoken language globally, qualifies as a low-resource language. The scarcity of benchmark datasets in Urdu has compelled researchers to employ increasingly innovative methods to overcome this challenge.

Deep neural networks (DNNs) [7] used in language modelling for ASR systems have yielded substantial performance enhancements in

* Corresponding author.
E-mail address: m.berrima@qu.edu.sa (M. Berrima).

https://doi.org/10.1016/j.eij.2025.100615
Received 3 August 2024; Received in revised form 25 December 2024; Accepted 11 January 2025
Available online 28 January 2025

high-resource languages like English and Mandarin [8–10]. However, achieving similar improvements in low-resource languages requires extensive training data. Deep Learning (DL) ASR systems [11] for languages with limited resources often necessitate the integration of additional training resources, such as cross-lingual acoustic models [12], or in-domain synthetic acoustic data. These measures are essential to achieve word error rates (WERs) comparable to those observed with classical Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) ASR paradigms. Several studies, including [13–15], have demonstrated that text-to-speech (TTS)-based synthetic data can facilitate end-to-end (E2E) ASR models in learning new languages. This is crucial for incorporating ASR features into new applications. Recent findings from [14] highlight practical challenges in language extension strategies, particularly in learning to identify new words without compromising the recognition of previously learned ones. The study explores various techniques to address this challenge, such as mixing actual and synthetic data with weighted sampling and applying different regularizations to each framework. Generative data-free quantization for ASR leverages generative models (e.g., GANs or VAEs) to create synthetic data resembling speech inputs, enabling model compression without requiring the original training data. This approach reduces the precision of weights and activations (e.g., to 8-bit) while preserving model performance, making ASR systems more efficient for deployment in resource-constrained environments [16–20].

The usefulness of synthetic speech data for training ASR systems has advanced significantly in recent years. In the training of acoustic-to-word speech recognition models, a Tacotron-2-based TTS engine [21] is employed to synthesize speech for new vocabulary [22]. Subsequent research [13] utilizes a multi-speaker TTS to enhance auditory variety in synthetic data by integrating speaker embeddings into the Tacotron-2 framework. In a later study [23], global style token (GST)-inspired embeddings are applied to update the Tacotron-2 version, improving the acoustical diversity of synthetic data, with GST proving superior to i-vector-driven embeddings. The study also demonstrates that combining TTS-based synthetic data, language model methods, and the general data augmentation technique (SpecAugment, [24]) is generally independent and effective. Another suggestion [25] proposes that end-to-end TTS with speaker presentations using a variational autoencoder (VAE) can increase the variety of speech in low-resource data. These prior studies underscore the importance of enhancing acoustic diversity in synthetic data for ASR model development, particularly in low-resource contexts. Based on this research, the present study aims to reduce the need for ASR model training on large datasets generated by TTS synthetic data. Current technologies can potentially achieve comparable results, but each has its own set of limitations. In semi-supervised learning (SSL) [26,27], a machine transcribes spoken audio instead of a human, requiring the collection, distribution, and storage of audio signals to construct machine-transcribed labels. In federated learning (FL) [28,29], numerous network updates are necessary when many devices collaborate to train a single global model, leading to increased costs for users updating their devices to use new features. Given the significance of language model strategies as alternate methods for improving end-to-end ASR performance using text-only data [30–32], this study focuses on WER and CER improvement by employing TTS synthetic data and a standard signal processing-based data augmentation.

This study proposes an end-to-end ASR (E2E-ASR) system designed for low-resource languages, utilizing synthetic speech, data augmentation, and transfer learning. Initially, we employ a non-autoregressive neural text-to-speech (TTS) engine, FastSpeech [33,34], known for its efficiency and effectiveness in speech synthesis. This model generates high-quality, clean speech, producing mel-spectrograms as an output. The neural TTS engine directly translates a sequence of phonemes into mel-spectrograms, capturing sound information akin to human perception. To enhance the ASR model, we introduce a data augmentation method that generates additional training data. This on-the-fly augmentation is applied to the mel-spectrograms, leveraging the distinct informative patterns crucial for recognition. As speech patterns typically remain consistent across speakers, these patterns prove effective in building a robust ASR system. The E2E-ASR treats mel-spectrograms as images, employing a convolutional neural network (CNN) architecture to learn features and improve word error rates (WERs) and character error rates (CERs). Additionally, this study incorporates transfer learning to maximize ASR performance. The key contributions can be summarized as follows: first, proposing an E2E-ASR system for low-resource languages using synthetic speech, data augmentation, and transfer learning; second, implementing an efficient and fast speech synthesis model, FastSpeech, to create mel-spectrograms with additional data augmentation, thereby increasing the number of samples for training the E2E-ASR.

The remaining study is structured as follows: Section 2 presents related ASR literature. Section 3 presents the proposed E2E-ASR system. Section 4 provides experiments. The results and discussions are outlined in Section 5. Section 6 presents concluding remarks and future directions.

2. Related literature

ASR models based on HMM and GMM dominated the speech recognition field for decades [35–37]. The HMM, a stochastic model, has a fixed and specified number of states established during the training process. The hidden state quantity in an input speech can vary according to the state. The HMM is predicated on the assumption that the input speech signal can be characterized as a parametric random selection with well-defined and precise parameters.

Deep learning models have evolved into a vital feature of TTS and ASR [38–41], as well as other speech analysis and recognition challenges, resulting in substantial advances in speech technology in recent years. In ASR systems, DNN-based acoustic models have almost overtaken HMMs and GMMs. Recently, studies in the area of speech technology have focused on convolutional neural networks (CNNs), which are successfully used in E2E models that are built using raw speech datasets. A speech recognition system is constructed by incorporating a limited-weight-sharing approach into the CNN framework. The findings revealed that the CNN decreased the WERs by 6%–10% when compared to the traditional techniques [42–44]. In another study, the CNN modelling technique was employed to recognize predetermined keywords in a spoken command recognition system [42]. This method used several files, including spoken instructions from Google's TensorFlow and AIY teams. The vanilla, single-layer SoftMax-based DNN and CNN were applied. The system using CNN achieved 95.1% accuracy. End-to-end acoustic modelling with a CNN [43] was employed using TIMIT, WSJ, MP-DE, and MP-F databases to perform the recognition task. The method produced a 1.9% WER, which was lower than the other available approaches. A study [45] used pre-trained wav2vec2.0 models to address low-resource speech recognition tasks in multiple spoken languages to test their language applicability. The study achieved 20% relative improvements in six low-resource languages.

A study [46] examined the performance of an E2E RNN-T model. The model comprises an encoder built from a CTC acoustic model and a decoder trained substantially from an RNN language model trained only on text data. Text and pronunciation data can be utilized to increase the performance of E2E ASR. The study observed that pre-training the RNN-T encoder using CTC increases WER by 5% compared to a 5-layer encoder while using a deeper 8-layer encoder instead of a 5-layer encoder improves WER by 10%. An E2E ASR [47] processes twice as fast as real-time on a Google Pixel phone and increases WER by more than 20% over a powerful embedded baseline model for both voice search and dictation tasks. Many studies on speech recognition in different languages have been attempted, but only a handful have been

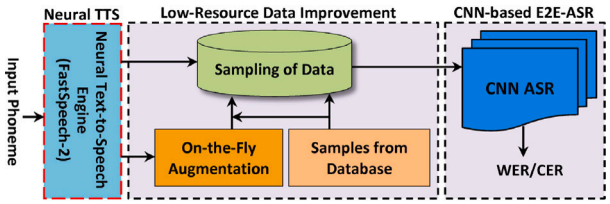

Fig. 1. Schematic of the Proposed ASR System for Low Resource Languages.

applied to low-resource languages. Mamyrbayev et al. [46] reported a Kazakh ASR system based on a DNN model using Kaldi tools. Later, in 2021, the study [48] reported an E2E ASR model for recognizing Kazakh language speech using an RNN-T, which has a similar network as an encoder–decoder with an attention framework. A study proposed MixSpeech [49], an effective data augmentation approach for low-resource ASR. MixSpeech trained an ASR model by using a weighted mix of two different speech features (like mel-spectrograms or MFCC) as inputs and recognizing both text sequences using the same combination weight. MixSpeech is applied to two prominent E2E ASR models, LAS (Listen, Attend, and Spell) and Transformer, and experiments are done on several low-resource datasets, including TIMIT, WSJ, and HKUST.

A study [50], for the Uzbek language and its dialects, suggested an E2E DNN-HMM automatic speech recognition model and a hybrid connectionist temporal classification (CTC)-attention network. The study [51] applied the CNN-DTW (dynamic time warping) technique to a low-resource language (Luganda), where the CAE (correspondence autoencoder) was used as input to the CNN-DTW model, and showed that this arrangement produced better results. A study [52] combined the MFCC feature extraction method with FDLP (frequency domain linear prediction) and proposed an MFCC-FDLP hybrid system for the Punjabi ASR system front-end feature extraction. Based on transfer learning and multilingual speech recognition on the E2E framework, a study [53] suggested an E2E ASR model for the low-resource Lhasa dialect. The approach started by making a monolingual E2E ASR system for the Lhasa dialect. Multiple source languages were used to start the ASR model, and transfer learning was used to compare how the source languages affected the Tibetan ASR model. There is also a proposal for a multilingual E2E ASR that uses different ways to start and units at different levels. Another study [54] suggested that a low-resource Sanskrit language could use self-supervised learning to do E2E ASR. Acoustic representations are learned using the wav2vec2.0 framework in an E2E deep learning technique. Using the Mozilla DeepSpeech architecture, an E2E method is proposed for developing ASR with the low-resource Tamil language [55]. For training, online computing resources were used, which made it possible to use the same methods for research in languages with few resources. The proposed ASR model achieved the highest WER of 24.7%, compared to 55% for Google speech-to-text. In the OpenASR21 Challenge, a study [56] explored and examined a set of wav2vec pre-trained models for E2E ASRs in 15 low-resource languages. The performance of pre-trained models using diverse pre-training audio data and architectures (wav2vec2.0, HuBERT, and WavLM) in low-resource languages is studied. An external E2E language model was used in a study [57] to look at better ways to adapt to low-resource languages in the context of transfer learning. First, a language-independent ASR system is built with a common vocabulary for all languages and a unified sequence-to-sequence (S2S) architecture. During the adaptation stage, language model fusion transfer is used when an external language model is added to the decoder network of the attention-based S2S model so that the language context of the target language can be used.

3. Proposed ASR for low resource language

This section presents the proposed E2E-ASR and its related modules. Fig. 1 demonstrates the proposed E2E-ASR system. It consists of a neural text-to-speech engine (FastSpeech [34]), data augmentation applied to the synthetic speech and a convolutional neural network-based ASR model for speech recognition. A multi-style training scheme is used to train the CNN model. The features for training the ASR are obtained after mixing speech from the limited database with TTS-based synthetic+augmented mel-spectrograms. The sampling weights optimize the ratio of real-to-synthetic speech during CNN training. This approach effectively combines low-resource speech with synthetic+augmented data in every batch, allowing the ASR model to see and train on both types of data. The E2E-ASR utilizes transfer learning to maximize the benefits of already available datasets (LibriSpeech [58] in our case).

3.1. Neural TTS engine

This study applies a neural TTS (as shown in Fig. 2(a)) to produce mel-spectrograms of clean synthetic speech. The TTS contains an encoder, variance adaptor, and predictor modules. Due to the increased variability and detailed representation (such as phases) observed in a speech waveform in comparison to a mel-spectrogram, there exists a greater difference in information between the input and output. Therefore, the waveform decoder in the original FastSpeech is omitted in our E2E-ASR model. The process begins with the encoder, which transforms the phoneme embeddings into phoneme-hidden sequences. The variance adaptor augments the hidden sequence by incorporating various types of variance information, including duration, pitch, and energy. Lastly, the mel-spectrogram decoder takes the adapted hidden sequence and generates a mel-spectrogram sequence. For the encoder, we use a feed-forward transformer that comprises a self-attention layer and a 1D-convolution layer stacked together as a fundamental structure (Fig. 2(b)).

The variance adaptor incorporates variance-related details (energy, pitch, and duration) into the phoneme sequences, as illustrated in Fig. 2(c). This additional information can enable the prediction of different versions of the speech, thus addressing the challenge of mapping a single input to multiple outputs in TTS. The variance adaptor contains (i) a duration predictor, (ii) a pitch predictor, and (iii) an energy predictor, respectively. The model structure for the predictors of duration, pitch, and energy is similar but with different model parameters. It comprises a two-layer 1D-convolutional network with ReLU activation, followed by layer normalization and a dropout layer. Additionally, there is an extra linear layer that projects the hidden states into the output sequence. The duration predictor uses a hidden sequence of phonemes as its input and forecasts the length of each phoneme in terms of the number of corresponding mel-frames. This information is converted to the logarithmic domain for easier prediction. Mean square error (MSE) loss optimizes the performance of the predictor by minimizing the errors. The duration predictor can be represented as:

$$\hat{d}_i = f(x_i; \theta_d) \tag{1}$$

where $\hat{d}_i$ is the predicted duration of the ith phoneme or syllable, x_i is the corresponding input feature vector, and θ_d are the learnable parameters of the duration predictor. The function $f(\cdot)$ is implemented by using CNN. The purpose of the length regulator LR (shown in Fig. 2(d)) is to address the problem of inconsistent length between the phoneme and spectrogram sequences in the variance adaptor. Typically, a phoneme sequence is shorter than its corresponding mel-spectrogram sequence, with multiple mel-spectrograms representing a single phoneme. This duration of mel-spectrograms that correspond to a phoneme is known as phoneme duration. To adjust for this, a length regulator expends the phoneme sequences by a factor of the phoneme duration d, resulting in the total length of the hidden states

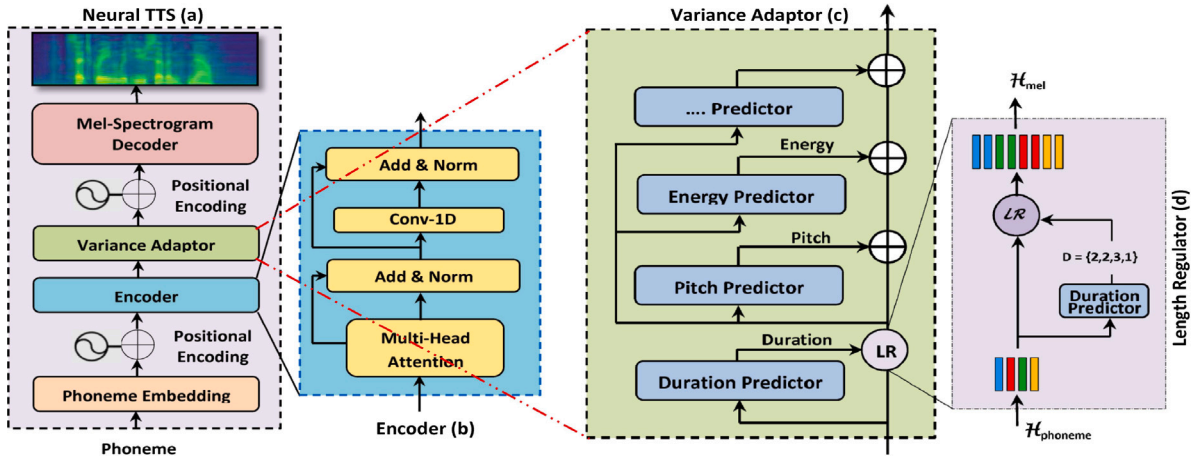

Fig. 2. The Neural TTS Framework. (a) FastSpeech [34] without Waveform Decoder, (b) Encoder structure, (c) Variance Adaptor, and (d) Length Regulator (LR).

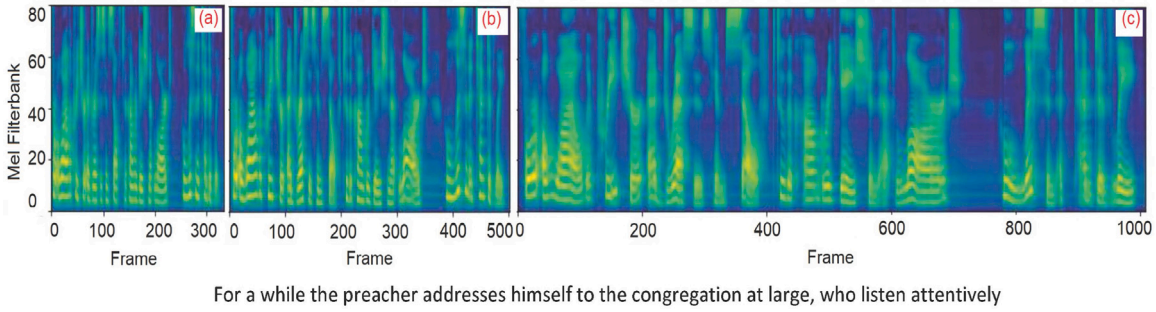

Fig. 3. The mel-spectrograms with α= [1.5x(a), 1.0x(b) and 0.5x(c)].

equal to a mel-spectrogram. The hidden states of the phoneme sequences and the phoneme duration sequence are denoted as $H_{phoneme} = [h_1, h_2, h_3, \ldots, h_m]$ and $D_{phoneme} = [d_1, d_2, d_3, \ldots, d_n]$, where m and n indicate the length of phoneme and mel-spectrogram sequences. The length regulator (LR) is given as:

$$H_{mel} = \mathfrak{LR}(H_{phoneme,D,\alpha}) \tag{2}$$

Where α indicates the hyperparameter that determines the length of the expanded sequence H_{mel}, hence controlling the speed of the voice. For illustration, for a given $H_{phoneme} = [h_1, h_2, h_3, h_4]$ with corresponding phoneme duration $D_{phoneme} = [2, 2, 3, 1]$, the expanded sequence H_{mel}, Eq. (2) becomes $[h_1, h_1, h_2, h_2, h_3, h_3, h_3, h_1]$ when $\alpha = 1$ (normal). For $\alpha = 1.3$ (slow) and $\alpha = 0.5$ (fast), the duration sequences are $D_{phoneme} = [2.6, 2.6, 3.9, 1.3]$ and $D_{phoneme} = [1, 1, 1.5, 0.5]$, respectively, and illustrated in Fig. 3 for reference. The input text to the TTS is Urdu text translated as "For a while, the preacher addresses himself to the congregation at large, who listen attentively".

The pitch predictor predicts the pitch contours. For better predictions of the variations in the pitch contours, the CWT (continuous wavelet transform) decomposes the pitch series into a pitch spectrogram. The pitch predictor uses these pitch spectrograms as a training target, optimized by minimizing the MSE. The pitch predictor during the inference generates pitch spectrograms, which are then converted to pitch contours by applying inverse CWT. For a given pitch counter function F_0, the CWT converts it into a pitch spectrogram $Z(\tau, t)$ as:

$$Z(\tau, t) = \tau^{\frac{1}{2}} \int_{-\infty}^{+\infty} F_0(y)\psi(\frac{y-t}{\tau})dy \tag{3}$$

Where ξ denotes the Mexican hat mother wavelet, $F_0(y)$ denotes the pitch values in position y whereas t denotes the position and τ denotes the scale of the wavelet. By using inverse CWT, the primary F_0 can be recovered from $Z(\tau, t)$:

$$F_0(t) = \int_{-\infty}^{+\infty} \int_{0}^{+\infty} Z(\tau, t)\tau^{\frac{-5}{2}} \psi(\frac{x-t}{\tau})dx d\tau \tag{4}$$

The F_0 can be represented as 10 different components if F_0 is decomposed into 10 scales, given as:

$$Z_k(t) = Z(2^{k+1}\tau_0, t)(k + 2.5)^{\frac{-5}{2}} \tag{5}$$

Where $k = 1, \ldots, 10$, $\tau_0 = 5msec$. For 10 wavelet components $\hat{Z}_k(t)$, the pitch counter $\hat{F}_0$ can be recomposed as:

$$\hat{F}_0(t) = \sum_{k=1}^{k=10} Z_k(t)(k + 2.5)^{\frac{-5}{2}} \tag{6}$$

The energy predictor predicts the original values of energy and optimizes the energy predictor with the MSE loss function. Since energy is not as highly variable as pitch, the energy is not transformed using CWT.

The architectures for predicting variations (pitch, duration, and energy) follow similar structures with different parameters. The architectures consist of a 2-layered 1D-CNN with ReLU, where each layer is followed by normalization and dropout regularization, as illustrated in Fig. 4(b). In addition, there is an extra linear layer for converting the hidden states into the output sequence. For the mel-spectrogram decoder, this study employs a feed-forward transformer block that comprises a self-attention layer and a 1D-convolution layer stacked together as a fundamental structure (as in Fig. 4(a)).

3.2. On-the-fly speech augmentation

Speech augmentation increases the effective size of existing speech data, even when the volume of training data is inadequate [24]. This

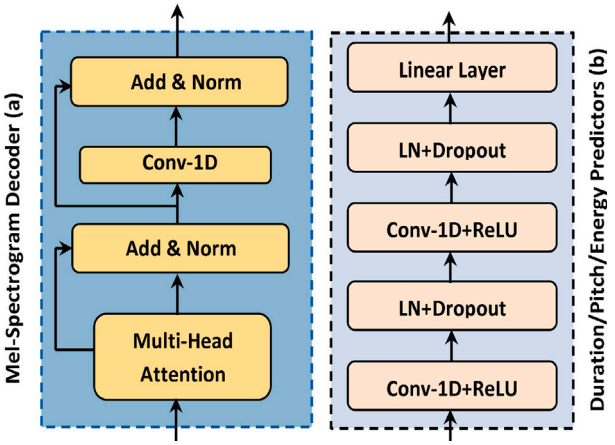

Fig. 4. Mel-Spectrogram Decoder and structures of the Duration/Pitch/Energy predictors.

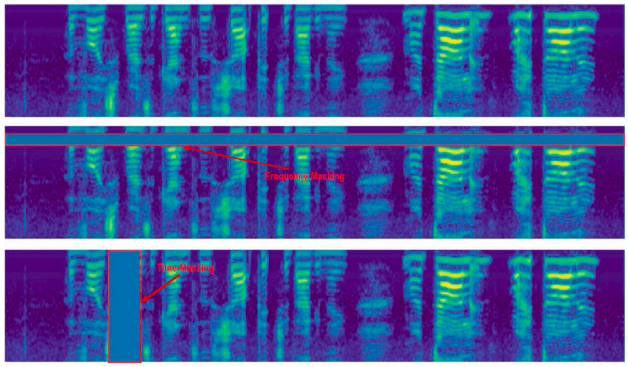

Fig. 5. Individual augmentations applied to a mel-spectrogram. Frequency masking augmentation (middle mel-spectrogram) and Time masking augmentation (bottom mel-spectrogram).

technique proves to be effective in enhancing the performance of ASR systems. For ASRs, augmentation typically involves modifying the audio waveforms in various ways, such as altering the speed or adding background noise. This enhances the dataset's effective size, given that various augmented iterations of an individual input are introduced to the network during the training process. Additionally, this approach promotes the network's resilience by compelling it to acquire pertinent features. Conventional audio augmentations are computationally expensive and sometimes require additional data. Thus, in this study, an on-the-fly augmentation technique is employed to directly apply transformations to the mel-spectrograms, which aids the network in acquiring useful features. Speech augmentation consists of frequency and time masking. The deformation policies concerning frequency and time are:

- To apply frequency masking, a consecutive block of mel frequency channels (f_0, $f_0 + f$) of length f is masked, where the value of f is randomly selected from a uniform distribution between 0 and the frequency mask parameter F, and f_0 is chosen randomly from the interval [0, v-f], where v represents the total number of mel frequency channels.
- To apply time masking, a block of t consecutive time steps (t_0, t_0+t) is masked, where the value of t is selected randomly from a uniform distribution between 0 and the time mask parameter T, and t_0 is randomly selected from the interval [0, τ-t], where τ represents the total number of time steps.

Examples of the individual augmentations applied to a mel-spectrogram can be seen in Fig. 5.

3.3. CNN as low-resource ASR

This study employs a CNN as an ASR system, as shown in Fig. 6. The CNN model uses skip connections to prevent gradient decay and a flatten layer for recognizing low-resource speech. A mel-spectrogram serves as an input RGB image to CNN. Before feeding, the Mel-spectrogram is first reshaped to (200×200) pixels. The CNN for ASR in this study comprises 10 convolutional layers (Conv-2D), using 16 filters that gradually increase to 256 filters. Each convolutional set is followed by a Max pooling layer with a (2×2) pooling size, whereas the kernel size is set to (3×3). The output tensors after all convolutional sets follow dimensions of $(148 \times 148 \times 16)$, $(72 \times 72 \times 32)$, $(34 \times 34 \times 64)$, $(15 \times 15 \times 128)$, and $(5 \times 5 \times 256)$, respectively, with the filter size ranging from 16 to 256. A spatial dropout regularization is used to prevent overfitting due to data scarcity. A 50% spatial dropout rate is used to extract a 2D feature map from the mel-spectrogram instead of randomly dropping individual pixels. A standard dropout is applied after the final max-pooling layer. All convolutional layers use ReLU activation, but a softmax activation is applied after the flattened layer for one-of-many classification. The Bidirectional Long Short-Term Memory (BLSTM) layers [59] are added to capture the temporal dependency. The output of the CNN is fed into the BLSTM layer, which is particularly effective in capturing temporal patterns. In our configuration, we use two BLSTM layers followed by the fully connected layer; however, the specific number of layers may vary based on the experimental requirements. Each BLSTM layer consists of 832 cells and 1024 neurons, with 512 LSTM units in each direction. A batch size of 256 samples is used to train the CNN. To optimize the weights, the ADAM optimizer is applied. The cross-entropy loss function specifies the target labels used in CNN. In our case, using word-level transcriptions allows us to capture higher-level linguistic information.

Transfer learning is employed in E2E-ASR to leverage the advantages of high-resource data and improve the model performance on low-resource speech data. This involves the model learning important visual features of high-resource speech, which then refines its understanding when presented with low-resource mel-spectrograms. For this task, clean speech data is obtained from the LibriSpeech dataset, and no low-resource speech samples are used to train the ASR model. After ASR training on clean speech, transfer learning is applied. The ASR model is retrained using speech data from low-resource languages. During transfer learning, the weights and biases of the first four convolutional sets (Conv2D-1 to Conv2D-8, as depicted in Fig. 7) are kept frozen except for the last convolutional set (Conv2D-9 and Conv2D-10). The weights and biases of the last convolutional set are updated, thereby learning the patterns of low-resource speech signals. These frozen neurons instruct the optimizer to stop updating the hyperparameters of Conv2D-1 to Conv2D-8 and force the model to update the hyperparameters of the remaining convolutional neurons, thereby minimizing loss. A 70% dropout rate is applied during this procedure. Fig. 7 illustrates the frozen layers (non-trainable convolutional neurons, represented as a red box) and unfrozen layers (trainable convolutional neurons, represented as a green box) for transfer learning.

4. Experiments

4.1. Datasets

The experiments use datasets of the IARPA Babel program,[1] IEEE Hindi speech dataset,[2] the Kaggle Urdu dataset,[3] and Turkish dataset.[4]

[1] https://catalog.ldc.upenn.edu/LDC2016S08
[2] https://ieee-dataport.org/open-access/speech-dataset-hindi-language
[3] https://www.kaggle.com/datasets/hazrat/urdu-speech-dataset
[4] https://huggingface.co/datasets/emre/Open_SLR108_Turkish_10_hours

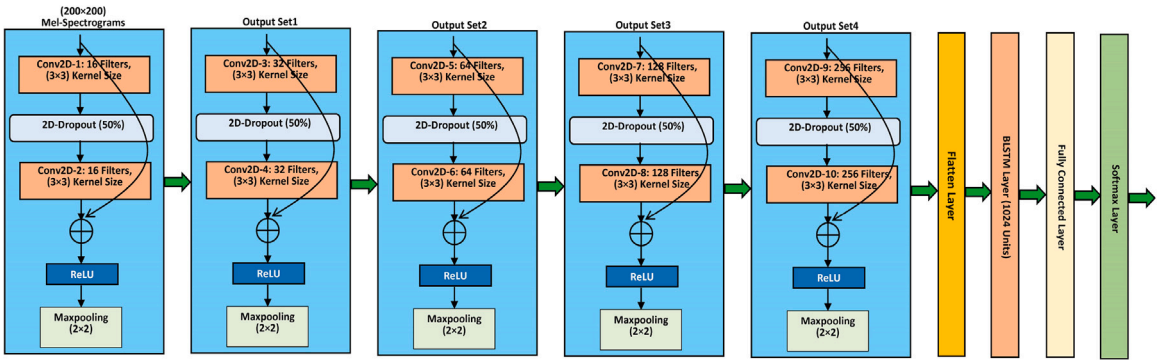

Fig. 6. Architecture of CNN-based Low-Resource ASR.

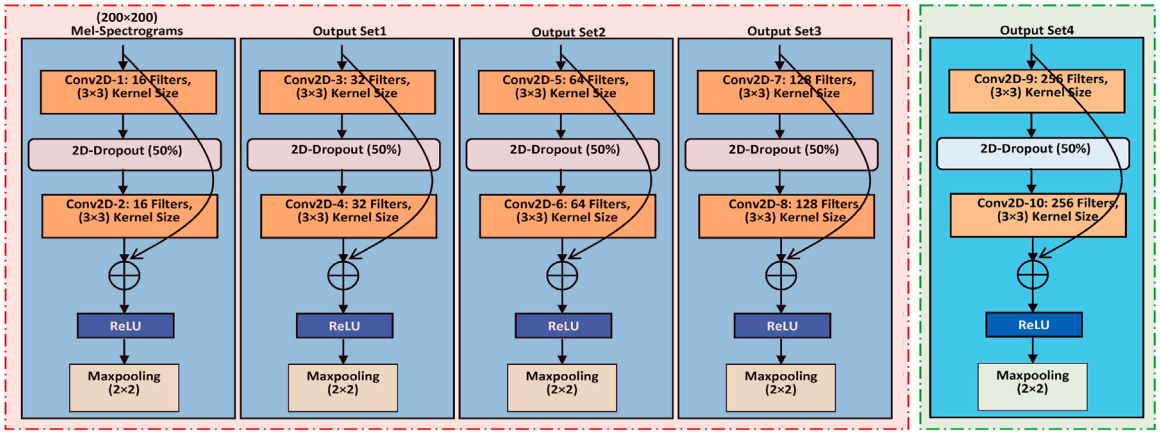

Fig. 7. Transfer learning with Frozen and Unfrozen convolutional neurons.

Table 1
Details of the CNN architecture.

Conv2D-1	Conv2D-2	Conv2D-3	Conv2D-4	Conv2D-5	Conv2D-6	Conv2D-7	Conv2D-8	Conv2D-9	Conv2D-10
Filters: 16, Kernel: (3 × 3) Maxpooling: (2 × 2)		Filters: 32, Kernel: (3 × 3) Maxpooling: (2 × 2)		Filters: 64, Kernel: (3 × 3) Maxpooling: (2 × 2)		Filters: 128, Kernel: (3 × 3) Maxpooling: (2 × 2)		Filters: 256, Kernel: (3 × 3) Maxpooling: (2 × 2)	
Flatten (1 × 6400); Final convolutional embedding length (1 × 60)									
BLSTM layers: 832 cells and 1024 neurons									

For transfer learning, this study uses the clean training set from LibriSpeech [58]. The Babel datasets created by IARPA include conversational telephone speech in 28 languages that were recorded in diverse settings. These recordings were made in real-life situations and under different conditions, like on the street while talking on a mobile phone. However, some of the languages have only a small amount of data collected using a distant microphone. The quantity of transcribed audio data differs based on the language and recording conditions. Turkish (10 h of training speech data) and Bengali (62 h of training data) are two low-resource languages selected from the IARPA Babel datasets. The IEEE Hindi database contains 2 h of training data, whereas the Kaggle Urdu dataset contains 1.3 h of training data. However, these low-resource databases are enhanced with synthetic and augmented speech samples to reduce WERs.

4.2. Pre-processing data

The transcripts for the neural TTS are selected from the Intelligence Advanced Research Projects Activity (IARPA) for different low-resource languages (Bengali, Turkish, and Hindi). The input text transcripts are first converted into a numerical representation that is fed into a deep neural network (DNN). This involves converting words into vectors to represent the input sequences. The text data is cleaned and preprocessed by tokenization, lowercasing, removing special characters, and handling language-specific refinement. The neural TTS model learns the mapping from text to speech to capture a natural speech waveform. However, the process begins with the encoder, which transforms the phoneme embeddings into phoneme-hidden sequences. The variance adaptor augments the hidden sequence by incorporating various types of variance information, including duration, pitch, and energy. Lastly, the mel-spectrogram decoder takes the adapted hidden sequence and generates a mel-spectrogram sequence. The Mel-spectrograms are first reshaped to (200×200) pixels before feeding to CNN.

4.3. Architecture details

The experiments use a Convolutional Neural Network (CNN) as an ASR system, as discussed in Section 3(c). By processing speech signals as a series of mel-spectrograms (visual representations of speech), CNN can learn to recognize patterns and features in the speech that can be used to transcribe spoken words. The Mel-spectrograms with (200×200) pixels are applied to the CNN. The BLSTM layers are added to capture the temporal dependency. The network architecture of the CNN ASR is provided in Table 1. The CNN for ASR in this study comprises 10 convolutional layers (Conv-2D), using 16 filters that gradually increase

Table 2
TTS configuration.

S #	Hyperparameter	Configuration
1	Phoneme Embedding Dimension	256
2	Encoder Layers	4
3	Encoder Hidden Size	256
4	Encoder Conv1D Kernel	9
5	Encoder Conv1D Filter Size	1024
6	Encoder Attention Heads	2
7	Mel-Spectrogram Decoder Layers	4
8	Mel-Spectrogram Decoder Hidden Size	256
9	Mel-Spectrogram Decoder Conv1D Kernel	9
10	Mel-Spectrogram Decoder Conv1D Filter Size	1024
11	Mel-Spectrogram Decoder Attention Headers	2
12	Encoder/Decoder Dropout	0.1
13	Variance Predictor Conv1D Kernel	3
14	Variance Predictor Conv1D Filter Size	256
15	Variance Predictor Dropout	0.5

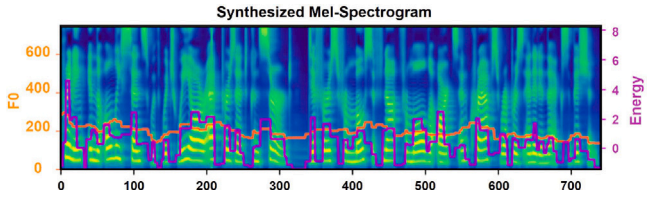

Fig. 8. Example Neural TTS Synthesized Mel-Spectrogram.

to 256 filters. Each convolutional set is followed by a Max pooling layer with a (2×2) pooling size, whereas the kernel size is set to (3×3). The output tensors after all convolutional sets follow dimensions of $(148 \times 148 \times 16)$, $(72 \times 72 \times 32)$, $(34 \times 34 \times 64)$, $(15 \times 15 \times 128)$, and $(5 \times 5 \times 256)$, respectively, with the filter size ranging from 16 to 256. A spatial dropout regularization is used to prevent overfitting due to data scarcity. A 50% spatial dropout rate is used to extract a 2D feature map from the mel-spectrogram instead of randomly dropping individual pixels. A standard dropout is applied after the final max-pooling layer. All convolutional layers use ReLU activation, but a softmax activation is applied after the flattened layer for one-of-many classification.

The neural text-to-speech (TTS) system is composed of an encoder with four feed-forward transformer blocks and a decoder that generates mel-spectrograms. The output linear layer of the decoder converts the hidden states into mel-spectrograms with a dimension of 80. The training of our model employs mean absolute error (MAE) optimization. Each feed-forward transformer block has a phoneme embedding dimension and a hidden self-attention size of 256. The self-attention layer is followed by a 2-layer convolutional network in which there are two attention heads. The kernel sizes for the 1D convolutions in this network are specified as 9 and 1, respectively. The initial layer possesses a dimensionality of 256 for input and 1024 for output, whereas the subsequent layer is configured with dimensions 1024 for input and 256 for output. The vocabulary size for phonemes comprises 76 elements, inclusive of punctuation. Within the variance predictor, the 1D-convolutional kernel sizes are uniformly set to 3, and both layers exhibit input/output dimensions of 256/256. A dropout rate of 0.5 is implemented in this context. The ADAM optimizer is employed with parameters $\beta = 0.9$ and $\beta = 0.98$, coupled with learning rates $\epsilon =$ and 10^{-9}. The complete details are provided in Table 2. Fig. 8 shows a synthesized mel-spectrogram.

5. Results and discussions

This section presents the experimental results for four low-resource databases: Bengali, Hindi, Turkish, and Urdu. The results are evaluated based on word error rate (WER) and character error rate (CER), with lower percentages reflecting superior performance. Table 3 provides an interpretation of various models. Here, TL denotes "Transfer Learning".

Table 3
Interpretation of different models.

	Model name	Interpretation
1	E2E-ASR-Clean	ASR trained by Clean Speech
2	E2E-ASR-Synthetic	ASR trained by Synthetic Speech
3	E2E-ASR-SpecAug	ASR trained by Augmented Speech
4	E2E-ASR-Joint	ASR trained by Clean+Synthetic+Augmented

5.1. Results with no transfer learning

The results obtained for the proposed E2E-ASR are provided in this section for the development and test sets. Also, results with the English language (LibriSpeech) are included; however, no augmentation or synthetic speech is added to the LibriSpeech. The results in Tables 4–7 indicate that some low-resource languages perform poorly due to a lack of clean speech data, such as Urdu. Insufficient training data for the ASR model leads to poor performance. However, better results were obtained for the Turkish language, where sufficient training speech data was available. Synthetic data mixed with Urdu language data significantly reduced the WERs and CERs, and on-the-fly speech augmentation achieved reasonable WERs and CERs in the Urdu language. The combination of synthetic speech and on-the-fly augmentation improved the WERs and CERs for all low-resource languages, particularly for Urdu (which had only 1.3 h of training data) and Hindi (which had only 2 h of training data) due to the lack of sufficient training data for both languages. In the test set (Table 4), the Urdu language shows an improvement in WERs, from 49.31% (E2E-ASR-Clean) to 30.14% (E2E-ASR-Joint), and in CERs, from 22.12% (E2E-ASR-Clean) to 11.36% (E2E-ASR-Joint). However, even with the reduced WERs and CERs, the results for Urdu are still lower compared to other low-resource languages. The Turkish language exhibits improved WERs and CERs, with rates of 28.34% and 10.01% respectively, after the application of speech augmentation. This represents a 13.1% improvement in WERs and a 9.12% improvement in CERs. The WERs and CERs are improved by 19.18% and 11.02% for the Bengali language and 19.25% and 10.98% for the Hindi language, respectively, through the combination of synthetic speech and on-the-fly speech augmentation. The E2E-ASR-Joint significantly reduced the WERs and CERs at the testing set. At the development set after fine-tuning, the model further improves the WERs and CERs, as indicated in Table 5. With the development set, the WERs and CERs with the Urdu language are improved by 2.20% and 2.03% at E2E-ASR-Clean. Similarly, at E2E-ASR-Joint, the WERs and CERs are improved by 2.87% and 1.14% with the Urdu language. The Turkish language performs the best at the development set with E2E-ASR-Clean, improving the WERs by 2.39% and CERs by 1.97%, respectively. Jointly with E2E-ASR-Joint, the WERs and CERs are improved by 3.41% and 0.87% with Bengali, whereas by 2.97% (WERs) and 0.67% (CERs) with the Hindi language.

5.2. Results with transfer learning

Improved training techniques, such as synthetic and augmented data, have shown reasonable performance in E2E-ASR models. However, these models still require large amounts of training data to achieve state-of-the-art performance, which is a serious challenge for low-resource languages. To address this issue, transfer learning is a simple yet effective method to improve WERs and CERs. In this study, we explore the effect of transfer learning on a speech recognition system for four low-resource languages. We first train an E2E-ASR model on the English LibriSpeech dataset without using synthetic or augmented data and then transfer the trained network to fine-tune it using only 3.5 h of training data from the low-resource datasets. The transfer learning approach achieves better performance and requires less training time than training the model from scratch. The transfer learning approach used in this study is explained in Section 3(c). The

Table 4
WER and CER results for Bengali and Hindi without and with transfer learning on testing sets.

Language	Bengali language				Hindi language			
ASR model	ASR: No TL		ASR: With TL		ASR: No TL		ASR: With TL	
Metric	WERs	CERs	WERs	CERs	WERs	CERs	WERs	CERs
E2E-ASR-Clean	45.29	21.04	44.83	20.61	44.12	20.87	43.57	20.35
E2E-ASR-Synthetic	40.38	16.22	39.71	15.58	39.61	16.01	38.89	15.34
E2E-ASR-SpecAug	32.39	12.11	31.66	11.42	31.01	12.10	30.21	11.39
E2E-ASR-Joint	26.11	10.02	25.24	9.23	24.87	09.89	23.88	09.02

Table 5
WER and CER results for Turkish and Urdu without and with transfer learning on testing sets.

Language	Turkish language				Urdu language			
ASR model	ASR: No TL		ASR: With TL		ASR: No TL		ASR: With TL	
Metric	WERs	CERs	WERs	CERs	WERs	CERs	WERs	CERs
E2E-ASR-Clean	41.45	19.13	40.87	18.60	49.31	22.12	48.75	21.65
E2E-ASR-Synthetic	32.78	15.88	32.04	15.20	41.65	18.44	40.94	17.87
E2E-ASR-SpecAug	28.34	10.01	27.51	09.28	35.17	14.21	34.38	13.52
E2E-ASR-Joint	20.22	08.98	19.24	08.09	30.14	11.36	29.21	10.58

Table 6
WER and CER results for Bengali and Hindi without and with transfer learning on development sets.

Language	Bengali language				Hindi language			
ASR model	ASR: No TL		ASR: With TL		ASR: No TL		ASR: With TL	
Metric	WERs	CERs	WERs	CERs	WERs	CERs	WERs	CERs
E2E-ASR-Clean	43.08	19.83	42.74	19.54	42.15	19.77	41.81	19.48
E2E-ASR-Synthetic	38.39	14.17	37.91	13.79	37.70	15.02	37.23	14.64
E2E-ASR-SpecAug	30.42	11.23	29.84	10.76	30.02	11.16	29.44	10.69
E2E-ASR-Joint	22.97	09.15	22.28	08.49	21.90	09.22	21.22	08.56

Table 7
WER and CER results for Turkish and Urdu without and with transfer learning on development sets.

Language	Turkish language				Urdu language			
ASR model	ASR: No TL		ASR: With TL		ASR: No TL		ASR: With TL	
Metric	WERs	CERs	WERs	CERs	WERs	CERs	WERs	CERs
E2E-ASR-Clean	39.06	17.16	38.72	17.87	47.11	20.09	46.77	19.80
E2E-ASR-Synthetic	30.79	14.63	30.28	13.98	38.31	16.96	37.83	16.58
E2E-ASR-SpecAug	26.37	09.24	25.79	08.77	33.08	13.18	32.50	12.71
E2E-ASR-Joint	18.24	08.24	17.55	07.56	27.28	10.22	26.59	09.56

training/validation accuracy and loss are illustrated in Fig. 9.

Tables 4–7 show ASR results after transfer learning for the development and test sets. The findings suggest that Urdu, a low-resource language, shows improved performance on the testing set compared to earlier results. This is attributed to the learned parameters resulting from transfer learning. Inadequate training data for the ASR model leads to suboptimal performance, but this can be improved through transfer learning. Furthermore, transfer learning yields better outcomes for the Turkish language, with sufficient training speech data available. Synthetic data mixed with Urdu language data significantly reduced the WERs and CERs after transfer learning, and on-the-fly speech augmentation achieved reasonable WERs and CERs. The combination of synthetic speech and on-the-fly augmentation as well as transfer learning improved the WERs and CERs for all low-resource languages, particularly for Urdu (which had only 1.3 h of training data) and Hindi (which had only 2 h of training data). With the test set after applying transfer learning, the WERs and CERs with the Urdu language (E2E-ASR-Clean) are reduced by 0.56% and 0.47%. Further, with the development set after applying transfer learning, the WERs and CERs with the Urdu language (E2E-ASR-Synthetic) are reduced by 0.48% and 0.38%. Jointly with synthetic+augmentation (E2E-ASR-Joint) and transfer learning, the average WERs and CERs over all low-resource languages are improved by 0.95% and 0.83% with the test set, whereas by 0.98% (WERs) and 0.91% (CERs) with the development set. The experiments that utilized transfer learning have demonstrated the effectiveness of this approach in enhancing word error rates (WERs) and character error rates (CERs). The results obtained from these experiments affirm the success of the transfer learning strategy.

5.3. Comparison with related studies

Due to the limited number of studies on low-resource automatic speech recognition (ASR), it is challenging to evaluate the performance of the proposed end-to-end ASR (E2E-ASR) system. To address this issue, the models that were trained in a high-resource language like English were chosen for comparison. Although no synthetic speech was utilized during the training, on-the-fly augmentation was applied to the LibriSpeech dataset to enhance the word error rates (WERs) and character error rates (CERs) of the proposed E2E-ASR system. A comprehensive comparison between the proposed model and several prior studies that employed the LibriSpeech corpus is presented in Table 8. The comparison was based on factors such as the number of parameters utilized as well as the achieved CERs and WERs of each model. The proposed ASR model exhibits the best WER value among the works mentioned, with fewer trainable parameters. However, the achieved CER values are marginally less competitive than the residual-BiGRU-ASR baseline, making it a potential area for future research.

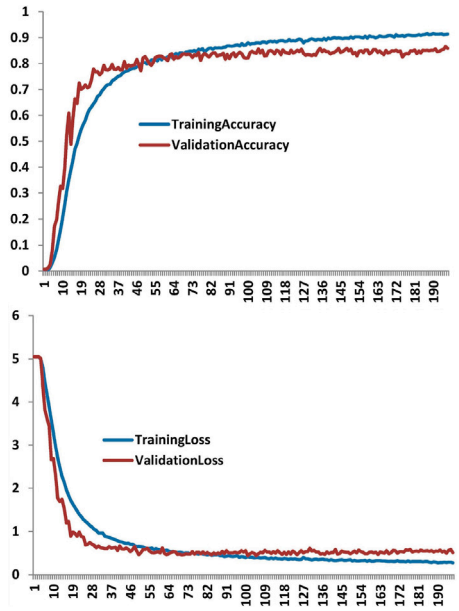

Fig. 9. Training/Validation Accuracy and Loss.

Table 8
Performance comparison on the LibriSpeech (Test-Clean) dataset.

ASR model	Para#	WERs	CERs
E2E-AutoEncoder-ASR [60]	130 M	18.0	8.4
Attention E2E-ASR [61]	70 M	5.40	6.6
GRC-CTC-ASR [62]	35 M	24.6	7.1
Residual BiGRU-ASR [63]	33 M	15.6	**4.7**
Transformer-based ASR [3]	—	37.2	9.4
MLU-E2E ASR [16]	—	30.79	—
VAKY-ASR [64]	—	20.60	5.11
Conformer-based ASR [65]	30.5 M	18.9	5.11
E2E ASR [66]	—	13.55	8.60
E2E-DeepASR (Proposed)	**28 M**	**14.3**	6.2

6. Conclusion

This study suggests an E2E-ASR system for low-resource languages (Bengali, Hindi, Turkish, and Urdu in our case) that uses deep learning, synthetic speech, and data augmentation. The study applies a non-autoregressive neural TTS engine to synthesize high-quality, clean speech and uses FastSpeech, an effective and fast speech synthesis model, to create the mel-spectrogram at the output. The sequence of phonemes serves as input to the neural TTS and directly outputs the mel-spectrogram, providing the model with sound information similar to what a human perceives. Additionally, a data augmentation method generates additional training data for ASR. Therefore, the study applies an on-the-fly data augmentation method to the mel-spectrograms. The E2E-ASR takes mel-spectrograms as images and learns the features using CNN architecture to improve the WERs and CERs. The results conclude that some low-resource languages perform poorly due to a lack of clean speech data, such as Urdu. Insufficient clean training data for the ASR model leads to poor performance. However, better results are obtained for low-resource languages after providing sufficient training speech data. Synthetic data mixed with low-resource language data significantly reduces the WERs and CERs, and on-the-fly speech augmentation achieves reasonable WERs and CERs. The combination of synthetic speech and on-the-fly augmentation improves the WERs and CERs for all low-resource languages, particularly for Urdu (which has only 1.3 h of training data) and Hindi (which has only 2 h of training data). Transfer learning is a simple yet effective method that improves WERs and CERs. After applying transfer learning to the test set, the WERs and CERs of the least low-resource language, Urdu, are reduced by 0.56% and 0.47% with the E2E-ASR-Clean model. Similarly, after applying transfer learning to the development set, the WERs and CERs for the Urdu language are reduced by 0.48% and 0.38% with the E2E-ASR-Synthetic model, respectively. By using a joint approach with synthetic+augmentation (E2E-ASR-Joint) and transfer learning, the average WERs and CERs over all low-resource languages improve by 0.95% and 0.83% with the test set, and by 0.98% (WERs) and 0.91% (CERs) with the development set. The proposed ASR model shows the best WER value among the mentioned research studies and has fewer trainable parameters. However, the achieved CERs are marginally less competitive than a few baselines, which suggests that it is a potential area for future research.

Current end-to-end (E2E) ASR systems face limitations such as the need for large amounts of annotated training data, challenges in handling diverse accents and noisy environments, and difficulty generalizing to low-resource languages. These issues are particularly pronounced for under-resourced languages where speech corpora are scarce. Transfer learning from well-resourced languages can help address these challenges by enabling models trained on large, high-quality datasets to be fine-tuned for low-resource languages. This approach leverages the shared linguistic features and representations learned from resource-rich languages, allowing the E2E-ASR system to generalize better to new languages with minimal data. By using transfer learning, the system can benefit from improved robustness and performance despite the limited availability of training data.

CRediT authorship contribution statement

Sami Dhahbi: Investigation, Formal analysis, Conceptualization. **Nasir Saleem:** Software, Methodology, Conceptualization. **Sami Bourouis:** Writing – original draft, Validation, Software. **Mouhebeddine Berrima:** Writing – review & editing, Data curation. **Elena Verdú:** Writing – review & editing, Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number RGP2/449/45. The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

References

[1] Kheddar H, Hemis M, Himeur Y. Automatic speech recognition using advanced deep learning approaches: A survey. Inf Fusion 2024;102422.
[2] Kheddar H, Himeur Y, Al-Maadeed S, Amira A, Bensaali F. Deep transfer learning for automatic speech recognition: Towards better generalization. Knowl-Based Syst 2023;277:110851.
[3] Khare S, Mittal AR, Diwan A, Sarawagi S, Jyothi P, Bharadwaj S. Low resource ASR: The surprising effectiveness of high resource transliteration. In: Interspeech. 2021, p. 1529–33.
[4] Stoian MC, Bansal S, Goldwater S. Analyzing ASR pretraining for low-resource speech-to-text translation. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing. IEEE; 2020, p. 7909–13.
[5] Scharenborg O, Ciannella F, Palaskar S, Black A, Metze F, Ondel L, et al. Building an ASR system for a low-resource language through the adaptation of a high-resource language ASR system: Preliminary results. In: Proc. internat. conference on natural language, signal and speech processing. 2017, p. 26–30.

[6] Saleem N, Gunawan TS, Kartiwi M, Nugroho BS, Wijayanto I. NSE-CATNet: Deep neural speech enhancement using convolutional attention transformer network. IEEE Access 2023.

[7] Saleem N, Gunawan TS, Shafi M, Bourouis S, Trigui A. Multi-attention bottleneck for gated convolutional encoder-decoder-based speech enhancement. IEEE Access 2023.

[8] Zhang P, Huang Y, Yang C, Jiang W. Estimate the noise effect on automatic speech recognition accuracy for mandarin by an approach associating articulation index. Appl Acoust 2023;203:109217.

[9] Yu H, Hu Y, Qian Y, Jin M, Liu L, Liu S, et al. Code-switching text generation and injection in Mandarin-English ASR. 2023, arXiv preprint arXiv:2303.10949.

[10] Yang C-HH, Li B, Zhang Y, Chen N, Prabhavalkar R, Sainath TN, et al. From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition. 2023, arXiv preprint arXiv:2301.07851.

[11] Almadhor A, Irfan R, Gao J, Saleem N, Rauf HT, Kadry S. E2E-DASR: End-to-end deep learning-based dysarthric automatic speech recognition. Expert Syst Appl 2023;222:119797.

[12] Tong S, Garner PN, Bourlard H. Cross-lingual adaptation of a CTC-based multilingual acoustic model. Speech Commun 2018;104:39–46.

[13] Ueno S, Mimura M, Sakai S, Kawahara T. Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing. IEEE; 2019, p. 6161–5.

[14] Zheng X, Liu Y, Gunceler D, Willett D. Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end ASR systems. In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing. IEEE; 2021, p. 5674–8.

[15] Alderazi F, Algosaibi A, Alabdullatif M, Ahmad HF, Qamar AM, Albarrak A. Generative artificial intelligence in topic-sentiment classification for Arabic text: a comparative study with possible future directions. PeerJ Comput Sci 2024;10:e2081.

[16] Qin S, Wang L, Li S, Dang J, Pan L. Improving low-resource tibetan end-to-end ASR by multilingual and multilevel unit modeling. EURASIP J Audio, Speech, Music Process 2022;2022(1):2.

[17] Qin H, Zhang X, Gong R, Ding Y, Xu Y, Liu X. Distribution-sensitive information retention for accurate binary neural network. Int J Comput Vis 2023;131(1):26–47.

[18] Qin H, Ding Y, Zhang X, Wang J, Liu X, Lu J. Diverse sample generation: Pushing the limit of generative data-free quantization. IEEE Trans Pattern Anal Mach Intell 2023;45(10):11689–706.

[19] Qin H, Ma X, Zheng X, Li X, Zhang Y, Liu S, et al. Accurate lora-finetuning quantization of llms via information retention. 2024, arXiv preprint arXiv:2402.05445.

[20] Huang W, Liu Y, Qin H, Li Y, Zhang S, Liu X, et al. Billm: Pushing the limit of post-training quantization for llms. 2024, arXiv preprint arXiv:2402.04291.

[21] Elias I, Zen H, Shen J, Zhang Y, Jia Y, Skerry-Ryan R, et al. Parallel tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling. 2021, arXiv preprint arXiv:2103.14574.

[22] Mimura M, Ueno S, Inaguma H, Sakai S, Kawahara T. Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition. In: 2018 IEEE spoken language technology workshop. IEEE; 2018, p. 477–84.

[23] Rossenbach N, Zeyer A, Schlüter R, Ney H. Generating synthetic audio data for attention-based speech recognition systems. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing. IEEE; 2020, p. 7069–73.

[24] Park DS, Chan W, Zhang Y, Chiu C-C, Zoph B, Cubuk ED, et al. Specaugment: A simple data augmentation method for automatic speech recognition. 2019, arXiv preprint arXiv:1904.08779.

[25] Du C, Yu K. Speaker augmentation for low resource speech recognition. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing. IEEE; 2020, p. 7719–23.

[26] Synnaeve G, Xu Q, Kahn J, Likhomanenko T, Grave E, Pratap V, et al. End-to-end asr: from supervised to semi-supervised learning with modern architectures. 2019, arXiv preprint arXiv:1911.08460.

[27] Ling S, Liu Y, Salazar J, Kirchhoff K. Deep contextualized acoustic representations for semi-supervised speech recognition. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing. IEEE; 2020, p. 6429–33.

[28] McMahan HB, Moore E, Ramage D, y Arcas BA. Federated learning of deep networks using model averaging. 2, 2016, arXiv preprint arXiv:1602.05629.

[29] Lin Y, Han S, Mao H, Wang Y, Dally WJ. Deep gradient compression: Reducing the communication bandwidth for distributed training. 2017, arXiv preprint arXiv:1712.01887.

[30] McDermott E, Sak H, Variani E. A density ratio approach to language model fusion in end-to-end automatic speech recognition. In: 2019 IEEE automatic speech recognition and understanding workshop. IEEE; 2019, p. 434–41.

[31] Variani E, Rybach D, Allauzen C, Riley M. Hybrid autoregressive transducer (hat). In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing. IEEE; 2020, p. 6139–43.

[32] Meng Z, Parthasarathy S, Sun E, Gaur Y, Kanda N, Lu L, et al. Internal language model estimation for domain-adaptive end-to-end speech recognition. In: 2021 IEEE spoken language technology workshop. IEEE; 2021, p. 243–50.

[33] Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, et al. Fastspeech: Fast, robust and controllable text to speech. Adv Neural Inf Process Syst 2019;32.

[34] Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech. 2020, arXiv preprint arXiv:2006.04558.

[35] Hadian H, Sameti H, Povey D, Khudanpur S. Flat-start single-stage discriminatively trained hmm-based models for asr. IEEE/ ACM Trans Audio, Speech, Lang Process 2018;26(11):1949–61.

[36] Mao S, Tao D, Zhang G, Ching P, Lee T. Revisiting hidden Markov models for speech emotion recognition. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing. IEEE; 2019, p. 6715–9.

[37] Islam M, Aloraini M, Aladhadh S, Habib S, Khan A, Alabdulatif A, et al. Toward a vision-based intelligent system: A stacked encoded deep learning framework for sign language recognition. Sensors 2023;23(22):9068.

[38] Kumar Y, Koul A, Singh C. A deep learning approaches in text-to-speech system: a systematic review and recent research perspective. Multimedia Tools Appl 2023;82(10):15171–97.

[39] Eren E, Demiroglu C. Deep learning-based speaker-adaptive postfiltering with limited adaptation data for embedded text-to-speech synthesis systems. Comput Speech Lang 2023;101520.

[40] Zeng T. Deep learning in automatic speech recognition (ASR): A review. In: 2022 7th international conference on modern management and education technology. Atlantis Press; 2022, p. 173–9.

[41] Stevenson E, Rodriguez-Fernandez V, Urrutxua H, Camacho D. Benchmarking deep learning approaches for all-vs-all conjunction screening. Adv Space Res 2023.

[42] Abdel-Hamid O, Mohamed A-r, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. IEEE/ ACM Trans Audio, Speech, Lang Process 2014;22(10):1533–45.

[43] Palaz D, Magimai-Doss M, Collobert R. End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. Speech Commun 2019;108:15–32.

[44] Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K. Speech recognition using deep neural networks: A systematic review. IEEE Access 2019;7:19143–65.

[45] Zhao J, Zhang W-Q. Improving automatic speech recognition performance for low-resource languages with self-supervised models. IEEE J Sel Top Sign Proces 2022;16(6):1227–41.

[46] Mamyrbayev O, Turdalyuly M, Mekebayev N, Alimhan K, Kydyrbekova A, Turdalykyzy T. Automatic recognition of kazakh speech using deep neural networks. In: Intelligent information and database systems: 11th Asian conference, ACIIDS 2019, Yogyakarta, Indonesia, April 8–11, 2019, proceedings, part II. Springer; 2019, p. 465–74.

[47] He Y, Sainath TN, Prabhavalkar R, McGraw I, Alvarez R, Zhao D, et al. Streaming end-to-end speech recognition for mobile devices. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing. IEEE; 2019, p. 6381–5.

[48] Mamyrbayev O, Oralbekova D, Kydyrbekova A, Turdalykyzy T, Bekarystankyzy A. End-to-end model based on RNN-T for Kazakh speech recognition. In: 2021 3rd international conference on computer communication and the internet. IEEE; 2021, p. 163–7.

[49] Meng L, Xu J, Tan X, Wang J, Qin T, Xu B. Mixspeech: Data augmentation for low-resource automatic speech recognition. In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing. IEEE; 2021, p. 7008–12.

[50] Mukhamadiyev A, Khujayarov I, Djuraev O, Cho J. Automatic speech recognition method based on deep learning approaches for uzbek language. Sensors 2022;22(10):3683.

[51] van der Westhuizen E, Kamper H, Menon R, Quinn J, Niesler T. Feature learning for efficient ASR-free keyword spotting in low-resource languages. Comput Speech Lang 2022;71:101275.

[52] Dua M, Kadyan V, Banthia N, Bansal A, Agarwal T. Spectral warping and data augmentation for low resource language ASR system under mismatched conditions. Appl Acoust 2022;190:108643.

[53] Pan L, Li S, Wang L, Dang J. Effective training end-to-end asr systems for low-resource lhasa dialect of tibetan language. In: 2019 Asia-Pacific signal and information processing association annual summit and conference. IEEE; 2019, p. 1152–6.

[54] Anoop C, Ramakrishnan A. CTC-based end-to-end ASR for the low resource Sanskrit language with spectrogram augmentation. In: 2021 national conference on communications. IEEE; 2021, p. 1–6.

[55] Changrampadi MH, Shahina A, Narayanan MB, Khan AN. End-to-end speech recognition of tamil language. Intell Autom & Soft Comput 2022;32(2).

[56] Peterson K, Tong A, Yu Y. OpenASR21: The second open challenge for automatic speech recognition of low-resource languages. In: Proc. Interspeech 2022. 2022, p. 4895–9.

[57] Inaguma H, Cho J, Baskar MK, Kawahara T, Watanabe S. Transfer learning of language-independent end-to-end ASR with language model fusion. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing. IEEE; 2019, p. 6096–100.

[58] Panayotov V, Chen G, Povey D, Khudanpur S. LibriSpeech: an asr corpus based on public domain audio books. In: 2015 IEEE international conference on acoustics, speech and signal processing. IEEE; 2015, p. 5206–10.

[59] Gamal M, Elhamahmy M, Taha S, Elmahdy H. Improving intrusion detection using LSTM-RNN to protect drones' networks. Egypt Inform J 2024;27:100501.
[60] Karita S, Watanabe S, Iwata T, Delcroix M, Ogawa A, Nakatani T. Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing. IEEE; 2019, p. 6166–70.
[61] Moritz N, Hori T, Le Roux J. Triggered attention for end-to-end speech recognition. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing. IEEE; 2019, p. 5666–70.
[62] Wang K, Guan D, Li B. Deep group residual convolutional CTC networks for speech recognition. In: Advanced data mining and applications: 14th international conference, ADMA 2018, nanjing, China, November 16–18, 2018, proceedings 14. Springer; 2018, p. 318–28.
[63] Reza S, Ferreira MC, Machado J, Tavares JMR. A customized residual neural network and bi-directional gated recurrent unit-based automatic speech recognition model. Expert Syst Appl 2023;215:119293.
[64] Chadha HS, Gupta A, Shah P, Chhimwal N, Dhuriya A, Gaur R, et al. Vakyansh: Asr toolkit for low resource indic languages. 2022, arXiv preprint arXiv:2203.16512.
[65] Bhogale K, Raman A, Javed T, Doddapaneni S, Kunchukuttan A, Kumar P, et al. Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages. In: Icassp 2023-2023 IEEE international conference on acoustics, speech and signal processing. IEEE; 2023, p. 1–5.
[66] Shahnawazuddin S, et al. Developing children's ASR system under low-resource conditions using end-to-end architecture. Digit Signal Process 2024;146:104385.