Nat. Volatiles & Essent. Oils, 2021; 8(5): 568 – 580

An efficient Tamil Text to Speech Conversion Technique based on Deep Quality Speech Recognition

Femina Jalin. A[1], Jayakumari. J[2]

[1]Department of Electronics and Communication Engineering Noorul Islam University,TamilNadu, India.
[2]Mar Baselios College of Engineering and Technology, Kerala, India.
E-mail: femina.jalin@gmail.com

Abstract

Our daily lives are impacted by the use of digital signal processing in speech processing. Many applications, such as automation, audio recording, and audio-based help systems, can benefit from text to speech conversion (TTS). Transcribing TTS is possible for many different languages, including those that are not widely spoken. Text-to-speech (TTS) systems generate spoken equivalents from text input. Though the creation of speech is rather complex, introducing naturalness to the speaker's expression is a major challenge in TTS. This paper proposes an efficient TTS conversion with high accuracy for the Tamil language. The deep learning technique called Deep Quality Speech Recognition (DQSR) is developed in this research study for Tamil language TTS. This is due to the fact that the method used for other languages like English will not work when used in Tamil due to adaptable pronunciations that are fully dependent on the language constructs. When compared to the traditional system, the proposed solution improves the framework's precision by 5%.

Keywords: Audio Recording; Deep Quality Speech Recognition; Digital Signal Processing; Tamil Language; Text to speech conversion.

Introduction

While maintaining linguistic content, the voice conversion (VC) technology transforms an exact speaker's voice into a desired target's voice [1]. They are two independent systems, but they share a common goal: creating speech with a objective voice in a variety of situations. According to the definition of a TTS system [2], the VC system is a speech synthesis system that uses linguistic instructions derived from written text. These similarities in purpose but differences in operating context make VC and TTS complement each other and have their own role in a spoken dialogue system, which is unique. TTS is able to generate a vast amount of speech automatically and affordably since text input is straightforward to develop and alter. There are times when a linguistic instruction cannot be written in a way that is intended. While voice as a reference input is more time-consuming and exclusive to generate, the scheme may be easily extended to a language that has never been used before.

According to the nature of the data available for expansion, VC systems are usually classed as parallel or nonparallel systems. According to this system, the parallel corpus is made up of two speakers who speak the same words at the same time. Dynamic temporal warping (DTW) is used to align the parallel speech utterances of source and target to form the training set. We can create a mapping function to convert the acoustic characteristics of one speaker to another using this training set [3]. For parallel VC systems, a variety of approaches have been presented [4, 5]. In parallel VC, only the source and target speakers' data is used, which is one of its advantages. For example, if we want to create a virtual assistant with a certain speaker's voice, we'll need a high planning and preparation for parallel speech acquisition than for non-parallel speech. For this reason, numerous ways have been proposed to construct non-parallel corpus systems for VC [6, 7]. A non-parallel VC can be adapted

from a parallel VC in the perfect space using the maximum-a posteriori (MAP) method [8] or interpolated between multiple parallel models [9] in the usual Gaussian mixture model (GMM) approach.

Using generative adversarial networks (GANs) or variational autoencoders (VAEs), An intermediate linguistic representation extracted from an automated speech recognition (ASR) model can be used to train a non-parallel VC [10, 11]. VC systems, whether parallel or non-parallel, are able to modify the voice, but not the duration of a speech [12-15]. As speaking rate is a speaker attribute, several recent studies have focused on converting speaking rate together with voices by utilizing sequence-to-sequence models [16, 17, 18 and 19]. A speaker-adaptive TTS model was recently developed by Luong et al. employing backpropagation method to accomplish speaker adaptation using untranscribed speech. As a result of our findings, we believe that the projected technology can be used to construct non-parallel VC as well. Here we present a framework for developing an automatic voice recognition system (VCR) from DQSR's untranscribed speech. Our approach is also tested for its ability to adapt and convert using speech utterances from an unknown language.

The rest of paper is prearranged as follows. Section 2 describes existing frameworks with its advantage and limitation, Section 3 introduces explanation of our proposed method, Section 4 labels the experiment setup with subjective results, and Section 5 concludes our findings.

2. Literature Review

The two most common ways to building TTS systems in the literature are rule-driven and data-driven. According to the rule-based method, linguistic rules were used to generate automatic pronunciation. [20] discusses a rule-based approaches that produce good results.for Tamil, [21] for English and [22] for Urdu. Decision trees [23] and Bayesian networks [24] were the two initial approaches explored for G2P. However, the resulting pronunciation was inferior to that of a phoneme-based decision tree because the syllable boundaries were not correctly detected [25]. Researchers next looked at Pronunciation by Analogy [21] and Pronunciation by Latent Analogy [26], which are only applicable for uncommon context terms.

A joint-grapheme-phoneme ngram method [27], often known as a joint-sequence model, is one of the various ways that have been examined. An enormous data set is required for this joint-sequence model, as are supplementary models [28] such as an Expectation Maximization-based alignment model and a translation model. Impossible to develop additional models, the Long-Short Term Memory (LSTM) neural network model avoids the need to do so, while providing identical results [29]. The use of neural networks and recurrent neural networks [30] to solve TTS difficulties has also been advocated, and they do generate good results. But the LSTM neural network and other neural networks have a dubious trait: how long it takes to analyze the data. Despite the fact that many models have been examined, a comprehensive TTS is still needed. This section discusses some of the work that has been done entirely for a Tamil TTS, as well as some material that has been converted for Tamil.

An article on Thirukural for the Tamil language was presented by Gl.J. Jayavadhan Rama [31]. There are no prosodic features in this, and they have tried to reduce the amount of the speech and increase its quality. To improve the naturalness, a pitch adjustment method is utilized. Synthesized speech was designed by R.Muralishankar and A.G. Ramakrishnan [32] by using prosodic elements. Several emotions are subjected to linear prediction analysis to develop the speech quality.

In the Tamil language, K.G. Aparna [33] has created a machine that reads books for them. In order to do concatenation synthesis, grayscale images are scanned into binary images. This is followed by the text being split into various clusters. This is a complete TTS in Tamil by G.L. Jayavardhana Rama [34]. As a result, the system has been divided into two parts: Afterwards, the rules will be concatenated with the text to form the output. Prosody and pitch marking will be implemented as part of the offline phase. Smoothing is achieved using wave form interpolation.

Web-enabled TTs, using Java, were proposed by Prathiba and Ramakrishnan [35]. Concatenation is based on the syllable as the basic unit. Festival architects SrikanthMajji and Ramakrishnan [36] offered a TTS. If you want to write sentences in ILEAP, you can do so. In this strategy, the units are selected based on clusters of units. Implementation of the Viterbi algorithm In the Tamil language, J.Sangeetha [37] proposed a TTS. There are two ways to synthesize the speech. There is a high degree of naturalness in the word-level synthesis. When an input is not included in the speech corpus, syllable level concatenation is used to synthesize the input.

3. Proposed Methodology

In this section, the dataset description along with the explanation of DQSR are presented.

3.1. Dataset Description

In this proposed system, to evaluate the results a new dataset is created. The data set has been recorded with the help of mike. The overall data set consist of two different kinds of data's such as words and sentences. Audacity was used to record the words and sentences in a closed room with a high-quality microphone. There is a backup of the recorded speech files. Sampling rate used for recording was sixteen thousand hertz (KHz). Table 1 shows some of the words and sentences that were captured. In the words section, our dataset consist of 138 words and 60 sentences. For evaluation purposed all dataset sentence and words are manually labelled. Some of the sample words and sentences of the dataset has been shown in the below table.

Table.1. *Recorded sample words and sentences.*

Words	**Sentences**
அம்மா	எனக்கு உதவி செய்வீர்களா
பூனை	என் சபயர் மலர்
யானை	என்ன செய்தி
ஆனை	காலல வணக்கம்
ஊசி	சீக்கிரம் சகாண்டு வா
உதவி	நல்லா இருக்கிறேன் நீங்க
வணக்கம்	நீங்க என்ன வாசிக்கிறீங்க
நன்றி	நீங்க சராம்ப நல்லவர்

3.2. Deep Quality Speech Recognition

DQSR architecture is described in this section. Rather than using the encoder-attention-decoder architecture employed by the majority of sequence-to-sequence autoregressive and non-autoregressive generation algorithms, we design an unique feed-forward structure. Figure 1 shows

the overall model architecture of DQSR. In the next subsections, we go into further detail on the components.

3.2.1 Feed-Forward Transformer

Self-attention in Transformer and 1D convolution form the basis of the DQSR architecture. As seen in Figure 1, we name this construction a Feed-Forward Transformer (FFT). Phoneme to Mel-Spectrogram FFT uses multiple FFT blocks stacked one on top of the other, with N blocks on the mel-spectrogram side. In order to retrieve cross-positional information, the self-attention network uses a multi-head attention. We utilize a 2-layer 1D convolutional network with ReLU activation instead of Transformer's 2-layer dense network. Due to the fact that character/phoneme and mel-spectrogram sequences are more closely associated to nearby hidden states, this method was developed. In the experimental part, we test the effectiveness of the 1D conv network. Transform is followed by residual connections, layer normalization, and dropout.

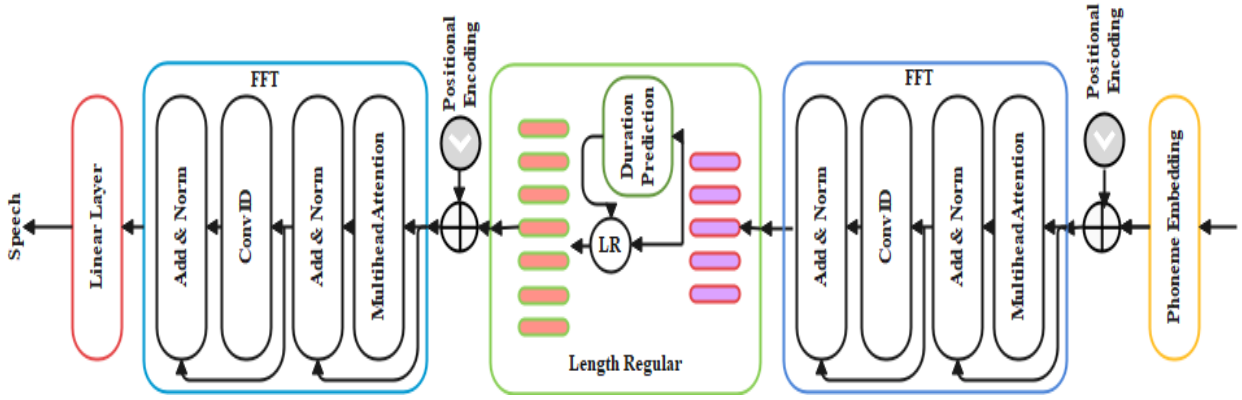

Figure 1: Overall Structure of DQSR

3.2.2 Length Regulator

As well as controlling the voice speed and portion of prosody, the length regulator (Figure 1) is employed to solve the length mismatch problem among the phoneme and spectrogram sequence in the FFT. The phoneme duration is the length of the mel-spectrogram that corresponds to a phoneme (we will describe how to predict phoneme duration in the next subsection). To determine how many hidden states there are in a phoneme sequence, we use a length regulator that multiplies phoneme duration by d. The length of the hidden states is thus equal to the length of the mel-spectrogram. $H_{pho} = [h_1, h_2, \ldots, h_n]$, denotes the hidden states of the phoneme sequence, where n is the length of the sequence. The phoneme duration sequence is $D = [d_1, d_2, \ldots, d_n]$,, where $\sum_{i=1}^{n} d_i = m$ and m represented as the length of the mel-spectrogram sequence (see below for more information). The length regulator LR is referred to as LR.

$$H_{mel} = LR(H_{pho}, D, \alpha) \quad (1)$$

when H_{mel}, is an expanded sequence and is a hyperparameter that controls the voice speed. Using Equation 1, the expanded sequence H mel becomes if $D = [2, 2, 3, 1]$, and H $[h_1, h_1, h_1, h_2, h_2, h_2, h_3, h_3, h_3, h_4]$ if $\alpha = 1$. The duration sequences D=1.3 = [2.6,2.6,3.9,1.3] [3,3,4,1] and $D_{\alpha=0.5} = [1, 1, 1.5, 0.5] \approx [1, 1, 2, 1]$, correspondingly, and the expanded sequences become

$[h_1, h_2, h_3, h_4]$We can also alter the prosody of the synthesized speech by adjusting the length of the space characters in the phrase.

3.2.3 Duration Predictor

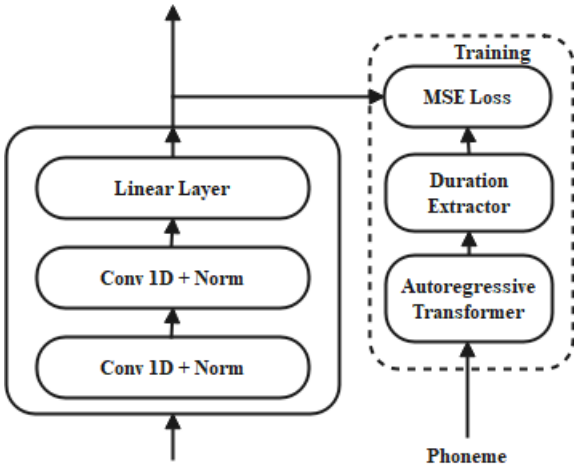

Fig.2. Duration Predictor

Length regulation relies heavily on the ability to forecast the duration of phonemes. It comprises of a 2-layer 1D network with ReLU activation, charted by a layer normalization layer and a dropout layer, as well as an extra linear layer that outputs the anticipated phoneme duration, as illustrated in Figure 2. There is an additional FFT block for each phoneme, and this module is used in conjunction with the DQSR model to predict mel-spectrograms for each phoneme with the MSE. As a result, they are more Gaussian and easier to train. Observe that the learned duration predictor is only needed during TTS inference since the phoneme duration recovered from an autoregressive teacher model can be used directly during TTS training (see following discussions).

The ground-truth phoneme duration is extracted from an autoregressive instructor TTS model, as shown in Figure 2, in order to train the duration predictor. Following is a description of the detailed steps::

- An auto-regressive encoder-attention decoder Transformer TTS model is initially trained using this technique.
- Each training sequence pair's attention alignments are extracted from the trained teacher model. Due to the multihead self-attention, there are different attention alignments, and not all attention heads exhibit the diagonal property. Attention head closeness to diagonal is measured by a focus rate F:

$$F = \frac{1}{S}\sum_{s=1}^{S} max_{1\le t\le T} a_{s,t} \quad (2)$$

where $S\ and\ T$ are the spectrogram and phoneme lengths, a_s, t is the element in the Sth row and Tth column of the attention matrix that has been donated. Every head's focus rate is computed, and then the attention is aligned with the head with the highest F rate.

Let's end with a series of the phoneme durations $D = [d_1, d_2, \ldots, d_n]$ using the duration extractor $d_i = \sum_{s=1}^{S}[\arg max_t a_s, t = 1]$. Meaning that a phoneme's duration depends on how many Mel-spectrograms it has received in the previous phase.

3.3. Model Configuration

DQSR model

Six FFT blocks are used in both the phoneme and mel-spectrogram sides of our DQSR model, a total of 12. Including punctuation, there are 51 phonemes in the vocabulary. Phoneme embeddings, self-attention hidden size, and 1D convolution are all set to 384. According to the settings, the sum of attention heads will be 2. There are two 1D convolution kernel sizes set to 3, with input/output size of 384/1536 for the first layer and 1536/384 for the second layer of the 2-layer convolution network. Three-hundred-and-fourth-dimensional mel-spectrogram is transformed into an 80-dimensional one via a linear output layer. Using a 1D convolution, the kernel sizes are set to 3, with input and output sizes of 384/384.

Autoregressive Transformer

Two reasons are served by the autoregressive transformer TTS model in our research. Phoneme duration will be extracted and used to train the duration predictor; mel-spectrogram will be generated during sequence-level knowledge distillation. There are six layers of encoder and decoder in this model and instead of a position-based FFN, there is 1D convolution network. It has a similar amount of parameters to our DQSR model, but it's not as complex..

4. Results and Discussion

We set up an experimental environment to carry out this research and assess the outcomes of the study. Intel i7 4th generation processor, clocking at 2.3 GHz with four megabytes of memory. The workstation is equipped with 16 gigabytes of DDR3 RAM and a 1 terabyte (TB) SATA hard drive rotating at 7 K RPM. Microsoft Windows 10 Pro is used as the base OS for this project. Version 2018a of MATLAB.

4.1. Evaluation metrics

The assessment of the quality of speech can be done using two measures. They are: subjective quality measures and objective quality tests. The evaluation of Subjective measures includes the comparisons of the original speech signal and the processed speech signals. The assessment of Objective speech quality measure involves the mathematical comparison of the original speech signal and the processed speech signals. The numerical expanse between the original and processed signals is calculated to measure the Objective quality measures. The various parameters considered for the assessment of speech quality measure are Mean Square Error (MSE), Mahalo Nobis distance, Similarity Index, Accuracy and Perceptual Evaluation of speech Quality (PSEQ) measure. MSE is well-defined as the average of the squares of the two errors viz., the difference between the estimated value and the predicted value. For a good quality speech signal the value of MSE should be less than one. The MSE is calculated by the equation:

$$MSE = \frac{1}{r}\sum_{i=1}^{r}[X_i - \hat{X_i}]^2 \qquad (3)$$

Where, r is the Predictions generated from the sample and X_i is the observed values of the variable being predicted $\hat{X_i}$ is the enhanced speech signal. As a result of their mean element vectors μA and μB, as well as the covariance matrix of all samples in the database, the Mahala Nobis distance

calculates how far apart two samples. The distance is given as Mahala Nobis distance is given by the equation,

$$M_D(\mu_A - \mu_B) = (\mu_A - \mu_B)^T \sum(\mu_A - \mu_B)^{-1} \quad (4)$$

The similarity index is calculated by the sum of matching words divided by the sum of all words and gap characters. Then the quotient is multiplied by 100 to give the similarity as a percent. Accuracy is calculated by taking the differences between the synthesized speech and the pronounced speech. Perceptual Evaluation of speech Quality (PSEQ) measure is computed by first equalizing the original and the degraded signals. Then the signals are filtered and processed to produce the loudness. To produce the quality rating, the difference between the original and the degraded signals are calculated. For better quality speech signal, the PSEQ value ranges from 1 to 5. The PSEQ is calculated by the equation,

$$PESQ = a_0 + a_1 D_i + a_2 A_i \quad (5)$$

Where D_i represent as the average disturbance value and A_i signified as an average asymmetrical disturbance value.

4.2. Performance Analysis of DQSR for Words and Sentences

Initially, Table 2 and 3 shows the overall performance of DQSR for words and sentences in terms of various metrics. Here the analysis are taken for more than three times to test the efficiency of proposed DQSR.

Table 2: Overall Performance of DQSR for Words

Word	MSE	MND	SI	PSEQ	Accuracy
அம்மா	0.038412	4.2769	21434574.5	1.0659	70.526
யானை	0.145327	7.0327	16638768.5	1.0987	75.689
பூனை	0.138834	5.6534	21854987.9	1.0846	66.452
ஆனை	0.12421	4.6776	275001574	1.0467	48.017
ஊசி	0.089234	5.3894	32093841	1.096	43.982

Table 3: Overall Performance of DQSR for Sentences

SENTENCES	MSE	MND	PSEQ	SI	Accuracy
எனக்கு உதவி செய்வீர்களா	0.067054	4.9873	1.0705	4844876.319	50.3294
என் சபயர் மலர்	0.06529	4.9877	1.2541	3357986.881	56.8291
என்ன செய்தி	0.06134	2.9694	1.8116	4414118.937	53.2901
காலல வணக்கம்	0.076432	3.8432	1.1088	4438267.284	44.9821
சீக்கிரம் சகாண்டு வா	0.07197	5.7878	1.9675	379246.963	55.29712

நல்லா இருக்கிறேன் நீங்க	0.074567	5.7978	1.8972	3917522.825	63.8267
நீங்க என்ன வாசிக்கிறீங்க	0.08346	4.7653	1.0974	4387986.175	66.5321
நீங்க சராம்ப நல்லவர்	0.09087	4.5987	1.2098	3643639.269	58.4731
சராம்ப நன்றி	0.8635	4.6093	1.04087	4387089.175	65.9047

From that analysis of Table 2 and 3, it is clearly stated that the proposed DQSR achieved better accuracy for some words and sentences. For instance, the word "elephant" has highest accuracy (i.e.75.68%), where the word "needle" has low accuracy (i.e.43.98%) than other words. Likewise, the sentence "What are you reading?" achieved better accuracy (66.53%) and the sentence "Good Morning" achieved low accuracy (44.98%) than other input sentences. The error rate of word "Amma" is very less, while comparing with other words includes elephant, cat, needle, etc. The MND is high for the "elephant", MND of "cat" and "needle" are nearly 5.70 and MND of "Amma" is only 4.27. The PSEQ for all the input words are nearly 1.10. The input sentences achieved less MSE value i.e. nearly 0.06 to 0.09, expect the last input sentence. The MND of "What's news?" achieved only 2.96, where other input sentences achieved nearly 3.8 to 5.0 of MND. Likewise, the PSEQ of all sentences achieved nearly 1.25 to 1.90. Therefore, the proposed DQSR achieved better performance, even the analysis are carried out more number of times. The next section will explain the performance of our proposed DQSR with existing techniques like Hidden Markov Model (HMM) and GMM.

4.3. Comparative analysis of Proposed DQSR with other techniques

Table 4 and 5 shows the comparison of proposed DQSR with existing HMM and GMM model in terms of MSE and accuracy for single word and single sentence.

Table 4: Performance of DQSR with existing techniques for Single Word

Technique	**HMM**		**GMM**		**Proposed DQSR**	
Word	**Accuracy**	**MSE**	**Accuracy**	**MSE**	**Accuracy**	**MSE**
அம்மா	65.255	0.239315	57.171	0.239174	70.526	0.238412
யானை	69.2962	0.22146	57.5472	0.241817	75.689	0.135327
பூனை	62.7034	0.191033	63.6949	0.140054	66.452	0.158834
ஆனை	46.951	0.6041	46.3837	0.639694	58.017	0.19421
ஊசி	39.1327	0.79278	51.1743	0.529215	53.982	0.289234

Table 5: Performance of DQSR with existing techniques for Single Sentence

Techniques	**HMM**		**GMM**		**Proposed DQSR**	
SENTENCES	**MSE**	**Accuracy**	**MSE**	**Accuracy**	**MSE**	**Accuracy**
எனக்கு உதவி செய்வீர்களா	0.079542	48.9407	0.129969	47.4095	0.067054	50.3294

என் சபயர் மலர்	0.17549	52.6695	0.22827	46.769	0.06529	56.8291
என்ன செய்தி	0.168783	49.1924	0.130133	53.0766	0.16134	53.2901
காலல வணக்கம்	0.81253	40.0768	0.81259	40.3113	0.76432	44.9821
சீக்கிரம் சகாண்டு வா	0.277968	52.6962	0.22789	54.0986	0.17197	55.29712
நல்லா இருக்கிறேன் நீங்க	0.077968	51.4609	0.034406	58.0892	0.074567	63.8267
நீங்க என்ன வாசிக்கிறீங்க	0.389637	61.5792	0.22789	54.0986	0.08346	66.5321
நீங்க சராம்ப நல்லவர்	0.192307	51.7018	0.335622	45.6671	0.09087	58.4731
சராம்ப நன்றி	0.189637	61.5792	0.234406	58.0892	0.08635	65.9047

From the table 4, it is clearly proved that proposed DQSR achieved better performance for every single input words. For instance, the proposed DQSR achieved 66.45% of accuracy and 0.13 MSE for the word "cat", where the existing techniques achieved nearly 63% of accuracy and obtained nearly 0.15 to 0.19 MSE for the same word. The proposed DQSR achieved highest accuracy (75.68%) and lowest MSE for the word "elephant". The existing HMM model achieved lowest accuracy (39.13%) and highest MSE for the word "needle". In addition, the experiments for single sentence is carried out to test the efficiency of proposed DQSR. From the results, it is proves that the proposed DQSR achieved highest accuracy (66.53%) for the sentence "What are you reading?", where the existing techniques achieved nearly 55% to 61% of accuracy for the same sentence. Likewise, for the sentence "Good Morning", the proposed DQSR achieved lowest accuracy (44.98%), where the existing techniques achieved only 40% of accuracy for the same sentence. Therefore, our proposed DQSR achieved better performance than HMM and GMM models.

4.4. Comparative Analysis of Proposed DQSR with existing techniques

The implemented work is compared with existing researches [38] for various parameters given in Table 6.

Table 6: Comparison of previous systems with implemented DQSR work

Parameters	Previous system 1	Previous system 2	**Proposed system**
Number of letter and words reserved TTS	110	110	110
Sentence changed correctly	96	98	**98.73**
Accuracy	87%	94%	**95.21**

As per [38], the existing research work is implemented on Punjabi text, however in order to test the efficiency of proposed DQSR, we implemented the existing research work on our input Tamil sentences and words. Those results are presented in the table 6, from that it is proved that the accuracy of our proposed DQSR is 95.21%, where the existing techniques achieved 94% and 87% of accuracy. In addition, the sentences are altered by existing and proposed DQSR techniques, where the proposed technique altered it correctly (98.73). The reason for that is the DQSR uses the various layers for converting the input text to speech signals. Therefore, our proposed DQSR achieved better performance than existing techniques.

5. Conclusion

Existing G2P (phoneme centric) rule-based and machine learning-based techniques in Tamil are still inadequate to handle all of the problems involved. Systems that create speech from text input are known as TTS. Though the creation of speech is rather complex, introducing naturalness to the speaker's expression is a major challenge in TTS. The degree of intelligibility and naturalness of the speech has been reached. This work illustrates a TTS conversion for the Tamil language that has been produced. TTS conversion utilizing DQSR is implemented and validated in this work. The results show that the method is highly accurate. A comparison of standard machine learning approaches with the proposed system's results shows that the suggested system generates speech waveforms with great accuracy. There is no deterioration or mispronunciation with the proposed system. According to the algorithm, its accuracy is 5 percent higher than that of a conventional technique. Using efficient meta-heuristic algorithms, the learning rate of deep learning approach will be optimized in the future.

References

[1] Tomoki Toda, Alan W Black, and Keiichi Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” IEEE Trans. Audio, Speech, Language Process., vol. 15, no. 8, pp. 2222–2235, 2007.

[2] Paul Taylor, Text-to-speech synthesis, Cambridge university press, 2009.

[3] Masanobu Abe, Satoshi Nakamura, Kiyohiro Shikano, and Hisao Kuwabara, “Voice conversion through vector quantization,” J. Acoust. Soc. Jpn. (E), vol. 11, no. 2, pp. 71–76, 1990.

[4] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, “Exemplar-based voice conversion using sparse representation in noisy environments,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 96, no. 10, pp. 1946–1953, 2013.

[5] Ling-Hui Chen, Zhen-Hua Ling, Li-Juan Liu, and LiRong Dai, “Voice conversion using deep neural networks with layer-wise generative training,” IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 22, no. 12, pp. 1859–1872, 2014.

[6] Daniel Erro, Asuncion Moreno, and Antonio Bonafonte, “INCA algorithm for training voice conversion systems from nonparallel corpora,” IEEE Trans. Audio, Speech, Language Process., vol. 18, no. 5, pp. 944–953, 2009.

[7] Yining Chen, Min Chu, Eric Chang, Jia Liu, and Runsheng Liu, “Voice conversion with smoothed gmm and map adaptation,” in Proc. EUROSPEECH, 2003, pp. 2413–2416.

[8] Chung-Han Lee and Chung-Hsien Wu, “Map-based adaptation for speech conversion using adaptation data selection and non-parallel training,” in Proc. INTERSPEECH, 2006, pp. 2254–2257.

[9] Tomoki Toda, Yamato Ohtani, and Kiyohiro Shikano, “Eigenvoice conversion based on gaussian mixture model,” in Proc. INTERSPEECH, 2006, pp. 2446–2449.

[10] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, “Phonetic posteriorgrams for many-to-one voice conversion without parallel data training,” in 2016 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2016, pp. 1–6.

[11] Hiroyuki Miyoshi, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari, “Voice conversion using sequence-to-sequence learning of context posterior probabilities,” Proc. INTERSPEECH, pp. 1268–1272, 2017.

[12] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, “Voice conversion from non-parallel corpora using variational auto-encoder,” in Proc. APSIPA. IEEE, 2016, pp. 1–6.

[13] Takuhiro Kaneko and Hirokazu Kameoka, “Parallelldata-free voice conversion using cycle-consistent adversarial networks,” arXiv preprint arXiv:1711.11293, 2017.

[14] Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba, “High-quality nonparallel voice conversion based on cycle-consistent adversarial network,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5279–5283.

[15] Andros Tjandra, Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura, “Vqvae unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019,” arXiv preprint arXiv:1905.11449, 2019.

[16] Cheng-chieh Yeh, Po-chun Hsu, Ju-chieh Chou, Hungyi Lee, and Lin-shan Lee, “Rhythm-flexible voice conversion without parallel data using cycle-gan over phoneme posteriorgram sequences,” in Proc. SLT, 2018, pp. 274–281.

[17] Kou Tanaka, Hirokazu Kameoka, Takuhiro Kaneko, and Nobukatsu Hojo, “Atts2s-vc: Sequence-to-sequence voice conversion with attention and context preservation mechanisms,” in Proc. ICASSP, 2019, pp. 6805–6809.

[18] Jing-Xuan Zhang, Zhen-Hua Ling, and Li-Rong Dai, “Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations,” arXiv preprint arXiv:1906.10508, 2019.

[19] Hieu-Thi Luong and Junichi Yamagishi, “A unified speaker adaptation method for speech synthesis using transcribed and untranscribed speech with backpropagation,” arXiv preprint arXiv:1906.07414, 2019.

[20] S. Yuvaraja, V. Keri, S. C. Pammi, K. Prahallad, and A. W. Black, “Building a Tamil voice using HMM segmented labels,” in National Conference on Communication, International Institute of Information Technology, Hyderabad, India, Language Technologies Institute, Carnegie Mellon University, USA, 2010.

[21]. R. I. Damper, Y. Marchand, M. J. Adamson, and K. Gustafson, “Comparative evaluation of letter-to-sound conversion techniques for English text-to-speech synthesis,” in Proceedings of the third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis, Jenolan Caves House, Blue Mountains, NSW, Australia, 1998, pp. 53–58.

[22]. S. Hussain, “Letter-to-sound conversion for Urdu TextTo-Speech system,” in Proceedings of the Workshop on Computational Approaches to Arabic Script-Based Languages, Association for Computational Linguistics, Geneva, Switzerland, 2004, pp. 74–9.

[23]. A. K. Kienappel and R. Kneser, “Designing very compact decision trees for grapheme-to-phoneme transcription,” in Proceedings of the Interspeech, Aalborg, Denmark, 2001, pp. 1911–4.

[24]. C. Ma, M. A. Randolph, and J. Drish, “A support vector machines-based rejection technique for speech recognition,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (ICASSP’01), IEEE, 2001, Vol. 1, pp. 381–4.

[25]. L. Jiang, H.-W. Hon, and X. Huang, “Improvements on a trainable letter-to-sound converter,” in Proceedings of the Fifth European Conference on Speech Communication and Technology, Rhodes, Greece

[26] J. R. Bellegarda, “Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy,” Speech Commun., Vol. 46, no. 2, pp. 140–52, 2005.

[27]. M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” 2008a, Speech Commun., Vol. 50, no. 5, pp. 434–51, 2008.

[28]. S. Jiampojamarn and G. Kondrak., “Letter-phoneme alignment: An exploration,” in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 780–8.

[29]. K. Rao, F. Peng, H. Sak, and F. Beaufays, “Grapheme-tophoneme conversion using long short-term memory recurrent neural networks,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Brisbane, QLD, Australia, 2015, pp. 4225–9.

[30]. H. Sak, A. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” in CoRR, Vol. abs/ 1402.1128. Available: http://arxiv.org/abs/1402.1128, eprint. 1402.1128, 2014.

[31] Rama, G.J., Ramakrishnan, A.G., Venkatesh, M.V. and Muralishankar, R., 2001. Thirukkural: a text-to-speech synthesis system. Proc. Tamil Internet, pp.92-97.

[32] Muralishankar, R., and A. G. Ramakrishnan. "Human Touch to Tamil Speech Synthesizer." Tamilnet2001, Kualalumpur, Malaysia (2001): 103-109

[33] Aparna, K.G., Rama, G.J. and Ramakrishnan, A.G., 2001. Machine Reading of Tamil Books. Proceedings of ICBME.

[34] Rama, G.J., Ramakrishnan, A.G., Muralishankar, R. and Prathibha, R., 2002, September. A complete text-to-speech synthesis system in Tamil. In Proceedings of 2002 IEEE Workshop on Speech Synthesis, 2002. (pp. 191-194). IEEE.

[35] Prathibha, P. and Ramakrishnan, A.G., 2002. Web-enabled Speech Synthesizer for Tamil. In Proceedings of Tamil Internet 2002 Conference, Cennai: Asian Printers (pp. 134-140).

[36] Sreekanth Majji, Ramakrisshnan A.G " Festival Based maiden TTS system for Tamil Language", 3rd Language & Technology Conference:Human Language Technologies as a Challenge for Computer Science and Linguistics October 5-7, 2007, Poznań, Poland.

[37] Sangeetha, J., Jothilakshmi, S., Sindhuja, S. and Ramalingam, V., 2013. Text to speech synthesis system for tamil. Int J Emerging Tech Adv En, 3, pp.170-5.

[38] Rashid, M. and Singh, H., 2019. Text to speech conversion in Punjabi language using nourish forwarding algorithm. *International Journal of Information Technology*, pp.1-10.