Text to Speech Synthesis Using Deep Learning

Rabbia Mahum, Aun Irtaza, and Ali Javed

1 Introduction

The best ways to communicate among human beings are speaking and writing. They communicate through natural languages, and the written form of language is called text. The vocal communication is done using speech that is easy to understand for the listeners. Speech or natural language processing techniques aim to develop a system that can analyze, generate, and understand human languages. Before the computer systems were able to understand only machine languages, however, with the advancement in machine learning and deep learning algorithms, the systems have become so powerful that they can process the statement in written or spoken form such as Google Search Engine [1].

Advancements in Internet of Things (IOTs) have made ways for the researchers to work for development of smart systems that aid people suffering from different health issues. Various health-care applications have been developed [2–5] to aid the smart technologies. One of the most important applications is artificial speech synthesis based-on various machine learning [6–8] techniques.

The computational models for human language processing utilize the human cognition in machines i.e. how humans store, and process the text. Due to enhancement in models, computer systems have become able to speak just like humans [9]. The artificial intelligent systems generate waveforms from an input text to synthesize speech. The input can be in any natural language form and the system may comprise of several phases for the conversion from text to speech. There exist various

R. Mahum (✉) · A. Irtaza
Computer Science Department, University of Engineering and Technology, Taxila, Pakistan
e-mail: rabbia.mahum@uettaxila.edu.pk; aun.irtaza@uettaxila.edu.pk

A. Javed
Software Engineering Department, University of Engineering and Technology, Taxila, Pakistan
e-mail: ali.javed@uettaxila.edu.pk

S. A. Parah et al. (eds.), *Intelligent Multimedia Signal Processing for Smart Ecosystems*, https://doi.org/10.1007/978-3-031-34873-0_12

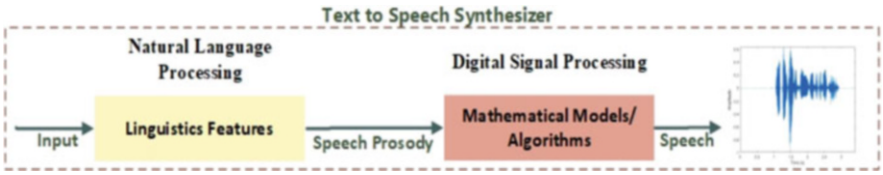

Fig. 1 The basic architecture of TTS synthesizer

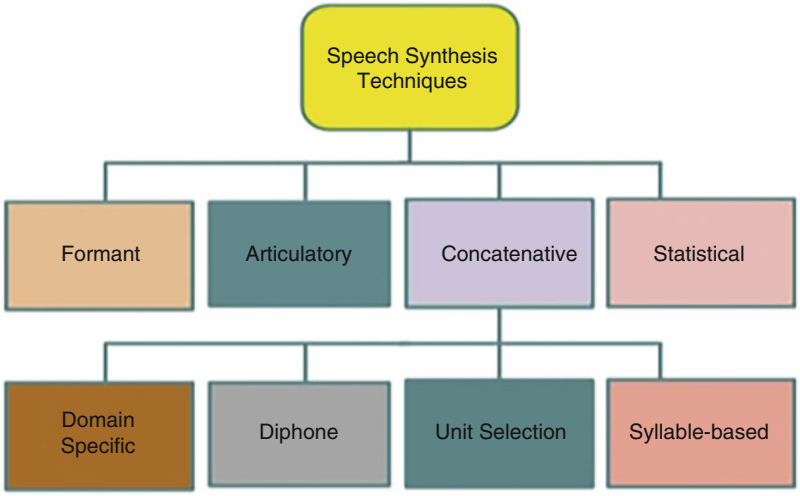

Fig. 2 Types of speech synthesis

applications of TTS synthesizers such as people with speaking difficulties are able to communicate better with those listeners who may not be able to understand sign language. Moreover, in educational background, TTS synthesis based systems can teach 365 days a year without any break. Furthermore, the applications can be programmed for any specific task i.e. for spell correction and pronunciation teaching. TTS synthesizers may be installed in security systems such as in warning or alarm systems, synthesized speech can give accurate information. Besides the positive uses of TTS synthesis systems, one drawback of these systems is to get used by hackers. The hackers may generate spoofed speech to fool the Automatic Spoofing Verification systems (ASV), which create a big threat for the community.

The basic architecture of TTS synthesizer is shown in Fig. 1.

There exist various types of speech synthesis methods as shown in Fig. 2.

The remaining chapter is organized as: Sect. 2 demonstrates some existing techniques for TTS synthesis and detection, Sect. 3 demonstrates the applications of TTS synthesizers, Sect. 4 refers to our proposed TTS synthesizer based on reinforcement block using encoder-vocoder. Furthermore, Sect. 5 demonstrates some experiments and in the end, conclusion of the chapter is presented in Sect. 6.

2 TTS Synthesis and Detection

There exist three types of modalities i.e., replay attack (RA), TTS, and voice cloning (VC) to synthesize speech. TTS and VC comprise of regenerated content and more similar to natural speech than RA. In ASV spoof 2019 competition, logical access (LA) and physical access (PA) tasks were introduced for synthesized speech, and RA for the development of ASV systems respectively. Various researchers have proposed different approaches [10–12] for the spoofing detection including all modalities. Some algorithms exist based on machine learning techniques to discern the audios based on data driven and knowledge focused countermeasures [13]. However, in traditional machine learning algorithms, hand crafted features extraction is performed which is time consuming task, moreover they may ignore the deep features underlying the audios spectrograms [14]. With the improvement in domain of convolutional neural networks (CNNs), some methods have been proposed based on deep layers such as in [15], an end to end algorithm employing raw waveforms as input is developed. Chintha et al. proposed a recurrent CNN structure to detect spoofed (fake) audios [16]. Moreover, a lightweight CNN has been employed by [17], namely LCNN utilizing softmax loss function to detect anti spoofed attacks. Furthermore, various combinations of detecting systems have been tested along with ResNet [18], and explored with other classifiers as well for better performance [19, 20] in spoofed speech detection. In [21], a model was employed based on an end-to-end ensemble method to learn the fusions of various detection systems. Even though, the performance of these proposed algorithms was satisfactory, however there exist an issue of generalization for unseen attacks in the models, and requirement of high computational resources such as time and memory, therefore it is required to introduce an efficient and robust system that can carry out detection of fake audios from any source.

From last two decades text to speech systems have become so powerful that are capable to generate a realistic voice after training of limited audio samples from target speakers [22]. Therefore, it is a huge threat for ASV systems as they may be attacked by the naturalness of the speech generated [23]. The applications that can protect the ASV systems from the spoofed audio attacks are called deepfake speech detectors. Thus, various machine learning and deep learning based works have been proposed in various domains [6–8] including the detection of forged speech. In [24], a SVM based classifier has been utilized as ASV employing Gaussian Mixture Model (GMM). They attained equal error rate (EER) as 4.92%, and 7.78% on the 2006 NIST for speaker identification core test. The authors have proposed the GMM, and a Relative Phase shift with Support Vector Machine (SVM) for the synthetic speech detection to minimize the weaknesses of speaker verification systems. Moreover, a detailed comparison of Hidden Markov Model (HMM), and DNN has been performed for the detection of spoofed speech [25]. In [26], proposed model employs the spectrograms in image form as input to CNN, thus forming a base of audio processing using images. In [27], various features descriptors have been used such as Mel Frequency Cepstral Coefficient (MFCC), spectrogram, etc.

and the effect of GMM-UBM on the accuracy has been analyzed. It is concluded that the combination of different feature descriptors gives better result in terms of EER. Chao et al. [24] utilized SVM to discern the real speeches from the fake recordings of the claimed man. Similarly, in [28] Chao has employed two core methods such as Kernel Fisher Discriminant (KFD) and SVM to verify speakers and attained better results as compared to their previous work based on GBM and UBM method. Moreover, to decline the computational cost of the polynomial kernel SVM by exchanging the dot product among two utterances with two i-vectors [29] was used. Furthermore, authors applied feature selection technique attaining 64% dimensionality reduction in features having EER of 1.7% [29]. Whereas, Loughran et al. [30] overcame the issue of imbalanced data (where the one class samples are greater than the other) utilizing Genetic algorithm (GA) with adjusted cost function. Malik et al. [31], developed a system for audio forgery detection based on acoustic signatures of environment by investigating the integrity of audio. However, these proposed models failed to address synthesized audio content with high precision.

3 Applications of the TTS Synthesis

The intelligent speech synthesizers have widespread domain of usage in development of human-machine interaction systems [32, 33]. Moreover, the systems based on TTS synthesizers are becoming more affordable to common people for daily use [34]. Some important applications of TTS synthesizers are given below.

3.1 Speaker

TTS synthesizers are widely used for the people who have speech difficulty due to disability, therefore they use speech synthesizers as a speaker to communicate. The implication of TTS synthesizers in small devices help them to improve their lifestyle without using sign language. Moreover, the English language is utilized as a medium in the most of these devices by service providers.

3.2 Screen Reader

The TTS synthesizers may be employed as screen readers for the people who have reading difficulties. Moreover, the people with visual impairment are not able to read screens from a specific distance, therefore TTS synthesizers work as an aid for these type of people. These screen readers are developed for various languages such as Urdu, Hindi, and English. They are very helpful for the people having dyslexia.

The TTS based screen reader may be used as listening any document while working, cooking or driving. The old age people can also use them if they have vision issues without any dependency. Various screen readers are available developed by different companies as web portals such as the IBM TTS (https://www.ibm.com/watson/services/text-to-speech/), and VoiceReader (https://www.linguatec.de/en/text-to-speech/voice-reader-home-15/). However, these screen recorders are mostly based on foreign languages, and lack Urdu language as a medium. Therefore, an effort is required to be done for Urdu screen recorder.

3.3 Language Teaching

The TTS synthesizers may be combined with a learning algorithm to develop helpful systems to learn new language by listening the words pronunciations. For example Capti-Voice is a TTS synthesizer system based on cloud platform supporting various foreign languages. However, there is still room for general language learning and teaching that may be able to teach and educate all existing natural languages.

3.4 Multimedia Applications and Telecommunication

The combination of speech synthesis and speech identification provide a significant user interface for mobile devices. The latest applications of TTS synthesizers include multimedia domain and internet search engines where a person is able to interact in his native language. The synthesized speeches are employed in call inquiry systems as well. Moreover, people who are unable to understand English language become able to understand the scenario in their native language using various TTS synthesized applications.

3.5 Entertainment

The synthesized speech is widely used in various games and talking robots or toys. Initially, the voice was not of good quality in talking calculators, however it has been improved in various 3D applications i.e. talking heads. Aiming to improve the learning process in kids, TTS synthesis can be incorporated in kid's toys easily. The IBM TTS [35] also provides the services for interactive toy development for kids. However, those toys mostly understand or speak English language which is a big challenge for illiterate people who are unable to understand English language. Therefore, the work is required for TTS synthesizers which can generate audios of different languages in toy industry.

3.6 Human-Machine Interaction

Various computer-machine interactive systems such as kiosks and automated tellers employ TTS synthesizers. Moreover, TTS systems may be used in alarms due to more accurate information delivery through TTS synthesis. The response due to warning speech is faster instead of warning light or signal from different room or place. Additionally, the activity log of printer devices can be heard on connected computer. However, all these applications mostly employ English language only.

3.7 Other Applications

Other than above-mentioned applications, the TTS synthesizers have been used in browsing, as reader in mails and SMS, dictionary, pronunciation, and PDF readers. Although, all of these applications use English language, however the work is yet to be done for other natural languages as well. The browser plug-ins and SMS reader application on android exist for Indian languages as well. During the last decade, the communication methods have been formed for 3 dimensional audio-visuals techniques. The vast applications of speech synthesis in different domains bring more funds and ideas. TTS synthesizers may also be employed in language interpreters, or talking mobile devices.

4 The Proposed Solution

Deep learning architectures are composed of various layers such as input, hidden, and classification layer. These hidden layers have various types i.e. convolutional, batch normalization, pooling, activation etc. The deep learning models extracts features utilizing various filters convolving over the input images. Moreover, when the filters are convolved over all the data then feature map is formed. These feature maps are reduced in dimensions employing pooling layers minimizing the computational power of the system. These feature maps can be fed again to convolutional layers repeating the above steps again. Numerous applications exist for various purposes such as facial feature recognition [36], speech identification [37], and emotion detection [38]. The basic architecture of deep network is shown in Fig. 3.

There exist various possibilities for flow of data in TTS synthesizers such as shown in Fig. 4.

Our proposed TTS synthesizer, consists of four different trained CNNs such as: (1) An encoder that generates a feature vector of speech from dataset [39], (2) A synthesizer that predicts a mel-spectrogram from grapheme sequences, conditioned on the vector from speaker embedding [34], (3) A Vocoder based on autoregressive

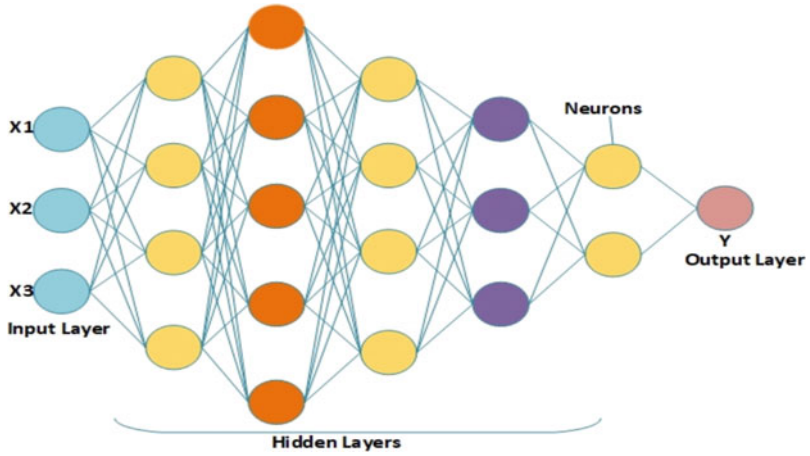

Fig. 3 The architecture of Deep network

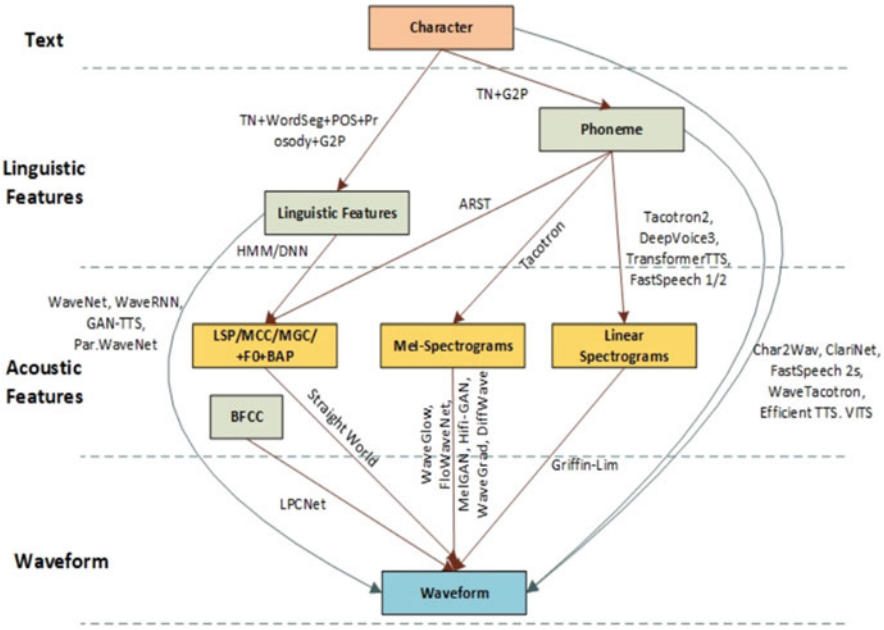

Fig. 4 A taxonomy of data flow from character to waveforms

WaveNet that generates the waveforms from spectrograms [40], (4) an additional RL block. The RL block consists of a neural network and make our proposed TTS model an agent which learns on the basis of reward. The accuracy of the network is computed and considered as a reward to give feedback to the main model comprising of Encoder, Synthesizer, and Vocoder.

4.1 Dataset

In the first phase of our proposed idea, we gathered data for English language from [41] i.e. LibriSpeech. The dataset comprises of 1000 hours having 16,000 Hz read English speech. The data was gathered from the LibriVox project's audio books. For French language based synthesis, we employed the SIWIS French speech synthesis database [42] having speech recordings of high quality along with text files. The database has been designed mainly for TTS synthesis systems. There exist 9750 utterances attained from numerous sources i.e. parliament speeches and speech talent by French professionals. The database is available freely, and comprised of 10 hours of speech in total. Moreover, for Chinese TTS synthesis, we employed Mandarin Chinese Speech Corpus [43]. It is comprised of 30 hours of annotated speeches. The speeches were collected using a single carbon microphone, and mostly participants were young speaking Mandarin with fluency.

4.2 The Proposed Encoder

The encoder is employed to form a condition for synthesizer block on a speech input by dataset. For the good generalization of the model, feature representations from various speakers are essential to capture without considering the background noise and phonetic content. These requirements are attained utilizing a speaker-discriminative technique that is trained employing a speaker verification task independent of text. We utilized [39], a very reliable neural network for the speaker verification. This network matches log mel-spectrograms sequentially attained from the speeches having an arbitrary length to an embedding vector of fixed dimensions, known as *d-vector* [44, 45]. The network's training is optimized based on generalized end-to-end loss for speaker verifications, so the same speaker's embedding exhibit the high similarity between cosine, whereas varying speaker utterances are far apart in embedding space. The training data utilized for the network comprises of speech that are segmented as 1.6 seconds audios having corresponding speaker identity labels. The mel-spectrograms of 40-channel are fed to the network comprising of a 3 LSTM layers with 768 cells, each individual is followed through a projection to 256 dimensions. The last embedding is produced using L_2-normalization of the output at top layer in the final frame. Moreover, through the inference, an utterance of arbitrary length is divided into 800 ms windows, overlapping by half (50%).

The network get trains for each window, and the outputs are combined to compute average, and normalized for final utterance embedding formation. However, the encoder network was not directly optimized during training that could extract features relevant to speaker identification.

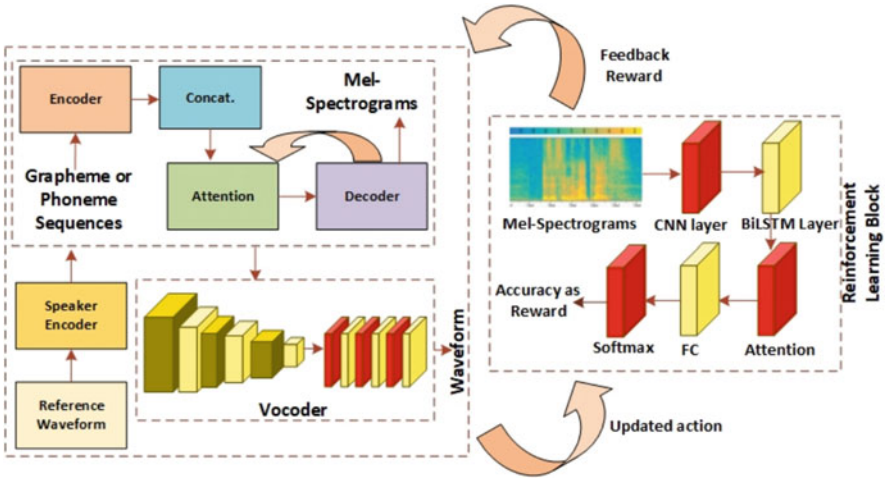

Fig. 5 The proposed model's architecture

4.3 The Proposed Synthesizer

In this section, we discuss the synthesizer's network which is based on Tacotron 2 architecture [34] using recurrent sequence to sequence model for supporting multiple speakers as described in [46]. At each time step, the synthesizer encoder's output is combined with an embedding vector. We analyzed that passing embedding to an attention layer as shown in Fig. 5, is better and network converges significantly across multiple speakers. We trained synthesizer over target audios and pairs of text transcripts. We mapped the text to phonemes at the input, which lead to improved pronunciation of difficult words and fast convergence. The proposed network is trained in a transfer learning manner, utilizing a pre-trained encoder (frozen parameters) to extract embedding from the target audios such as the target speech should be same to the speaker's reference signal. The features from target spectrograms are extracted from 50 ms windows along with 12.5 ms step, passed through 80 channel mel-scale filter bank followed by logs compression of dynamic range. We improved [34] by applying augmentation of the L_2 loss on the computed spectrogram with an additional L_1 loss. We concluded that the combined loss is more robust towards noisy training data. Moreover, we didn't employ an additional loss term for speaker embedding as in [47].

4.4 The Proposed Vocoder

We employed autoregressive WaveNet [40] vocoder sample-by-sample to transform the mel-spectrograms generated by synthesized network into time-domain wave-

forms. The architecture applied for the proposed vocoder is same as [34] comprising 30 dilated convolutional layers. The proposed network is not directly conditioned on the encoder's output. The mel-spectrograms generated via synthesizer network extract all of the details for high quality synthesis of various voices, which allow a vocoder to act as multi-speaker voice generator based on training data.

4.5 The Proposed RL Block

The RL block and our proposed model works together as an agent. The parameters of the proposed model are known as policy. Furthermore, the purpose of the policy is to predict the audio features at each instant. The features of speech describe the actions of agent. After the features prediction, the reward value is computed which gives feedback on accuracy. We employed policy gradient method for back-propagation that optimizes the model for maximum reward gain. As shown in Fig. 5, left block is TTS synthesizer based on Tacotron. Moreover, we utilized Griffin-Lim [48] technique for rebuilding the speech signals. Our proposed text editor is based on same structure as Tacortron2 that take input of text embedding and form output for encoder embedding. Whereas, the decoder based on attention module takes the output from encoder and embedding as input. Then, decoder predicts frame by frame speech features.

Reward The reward function is utilized for the computation of more natural speech that is based on a network as shown in Fig. 5. The network consists of a CNN layer, an attention layer, BiLSTM layer, a fully connected layer (FC) and softmax layer. The extracted features proved better efficiency for the natural speech synthesis using a neural network. The network takes the mel-spectrogram first as input, and then give output in the form of constant size latent output. Then, BiLSTM takes the temporal information and transform it into modified form. Whereas, an attention layer gets training for each frame's weight.

Let us suppose an input is x_i and the speech is $\grave{y}_i$ having speaker type as t_i, and the goal is to form a natural speech y_i having desired emotion l_{yi}. To update the model, we employed accuracy as the reward value. To form the end of sequence (EOS) token, the proposed RL network is employed to evaluate how accurate a mel-spectrogram feature y_i matches with the original speech label. The probability p_i for target speech with emotion l_{yi} of y_i is computed as below:

$$p_i = \mathrm{RL}\,(l_{yi} \mid \mathrm{yi}\,;\, \mathfrak{F}), \tag{1}$$

Here, ϑ represents the parameters of RL network, and the value of probability varies from 0 to 1. The RL reward is computed as:

$$R = \frac{N}{K} = \frac{\sum_{i=1}^{K} 1\,(p_i > \Omega)}{K} \tag{2}$$

Here, R is reward of spectrum features y_i. The reward value also varies from 0 to 1. K refers the size of sample, and Ω is a threshold number that is set to 0.5. N represents the total samples.

5 Experimental Evaluation

The generated speeches from our proposed TTS synthesizer are evaluated using Mean Opinion Score (MOS) for similarity with real speech and naturalness, based on subjective listening tests. The MOS score varies according to Absolute Category Rating scale [49] from 1 to 5 such as Excellent, Good, Fair, Poor, and Bad respectively. In total, 100 participants were included to assign MOS for naturalness and similarity, separately from 1 to 5 including 80 males and 20 females. 100 audios for each language, ranging from 20 to 35 seconds were generated and reviewed by participants. Then, the average score was computed to evaluate the quality of the synthesized audios. We attained 4.67 average MOS for similarity and 4.70 for naturalness for synthesized speech through our proposed model that is a significant number. More particularly, we achieved best MOS for naturalness as 4.75 and MOS for similarity as 4.71 for Chinese language as shown in Table 2.

5.1 Environmental Setup

We performed the experiments using a GPU NVIDIA card i.e. GEFORCE GTX with 4 GB memory. The details of employed hardware are shown in Table 1. The operating system was Windows 10 having a RAM of 16 GB. The experiment was performed on the Matlab 2021a.

Table 1 System specifications for the employed model

Hardware	Specifications
Computer	GPU Server
CPU	Intel Core i5
RAM	16 GB
GPU	NVIDIA GEFORCE GTX × 4

Table 2 Results over French, Chinese, and English datasets

Dataset	MOS Similarity	MOS Naturalness
French	4.6	4.64
Chinese	4.71	4.75
English	4.7	4.72
Average	4.67	4.70

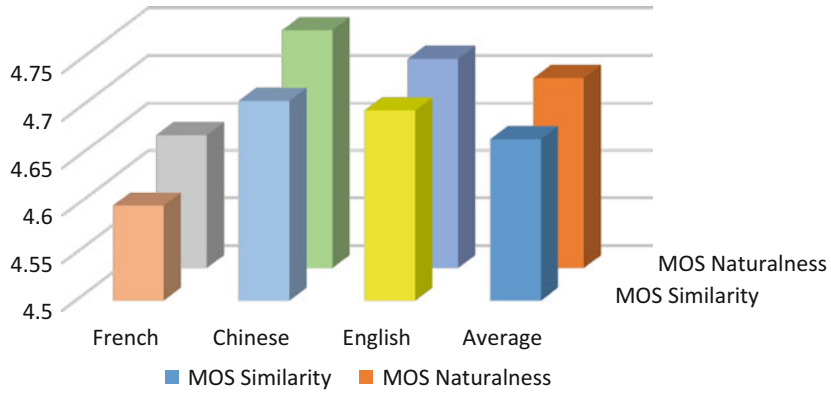

Fig. 6 The comparison plot for French, Chinese, and English Dataset

5.2 Assessment Over All Datasets

As mentioned in section of datasets, we have described the details of three datasets such as for English language: LibriSpeech, French language: SIWIS French speech synthesis database, and Chinese language: Mandarin Chinese Speech Corpus. In this section, we assess the individual results for our proposed TTS synthesizer as shown in Table 2. We employed the similar protocol for all three datasets i.e. average MOS for naturalness and similarity from group of 100 people. We achieved MOS similarity as 4.7 for English, 4.54 for Chinese, and 4.6 for French language dataset and average MOS similarity is 4.67. Whereas, MOS naturalness for French dataset is 4.64, for Chinese 4.75, and for English 4.72. Moreover, average MOS naturalness is 4.70. The comparison plot for all three datasets is shown in Fig. 6.

5.3 Time Complexity of Existing Techniques

In this section, we analyse various existing TTS synthesizers in terms of time complexity for training and inference time. In Table 3, T refers to the number of steps or iterations in the algorithm. Whereas, N represents the sequence length. Our proposed model attained $O(N)$ computational time as it is based on Tacotron 2 and inference time is also $O(N)$. Although, some algorithms take constant time such

Table 3 Comparative analysis of existing TTS models in context of time complexity

Model	Training	Inference
Tacotron 2	O(N)	O(N)
Deep Voice 3	O(1)	O(N)
FastSpeech 2	O(1)	O(1)
WaveGlow	O(T)	O(T)
WaveNet	O(1)	O(1)
Flowtron	O(1)	O(1)
Our proposed TTS	O(N)	O(N)

Table 4 Comparison with existing TTS synthesizers

Model	MOS	Inference Time(ms)
Tacotron 2	4.46	538
Fast Speech 2	3.83	374
VARA-TTS	3.88	33
BVAE-TTS	4.1	19.1
Our proposed TTS	4.67	13.5

as $O(1)$, however the resulting MOS is not considerable for these techniques as reported in Table 4. The details of complexities for training and inference time are reported in Table 3.

5.4 Comparison with Existing Techniques

In this section, we perform an experiment to compare our proposed TTS synthesizers with some existing models such as Tacotron2 [50], FastSpeech 2 [51], VARA-TTS [52], and BVAE-TTS [53] based on MOS and Inference Time. We utilized LibriSpeech dataset for assessment of all the mentioned systems in Table 4. It is exhibited from the table that MOS for Tacotron 2 is 4.46 that is maximum after our proposed TTS synthesizer. However, inference time for Tacotron 2 is maximum than other existing models i.e. 538 ms. Furthermore, Fast Speech 2 achieved 3.83 MOS, VARA-TTS attained 3.88 MOS, and BVAE-TTS obtained 4.1 MOS that is significant than former models. Moreover, the minimum inference time is attained by our proposed TTS synthesizer i.e. 13.5 ms. The Fast Speech 2 utilized 374 ms, VARA-TTS used 33 ms, and BVAE-TTS produced speech in 19.1 ms. Therefore, we believe that our proposed TTS synthesizer, achieved maximum MOS than existing models exhibiting significant naturalness. A comparison plot with existing TTS synthesizers is shown in Fig. 7.

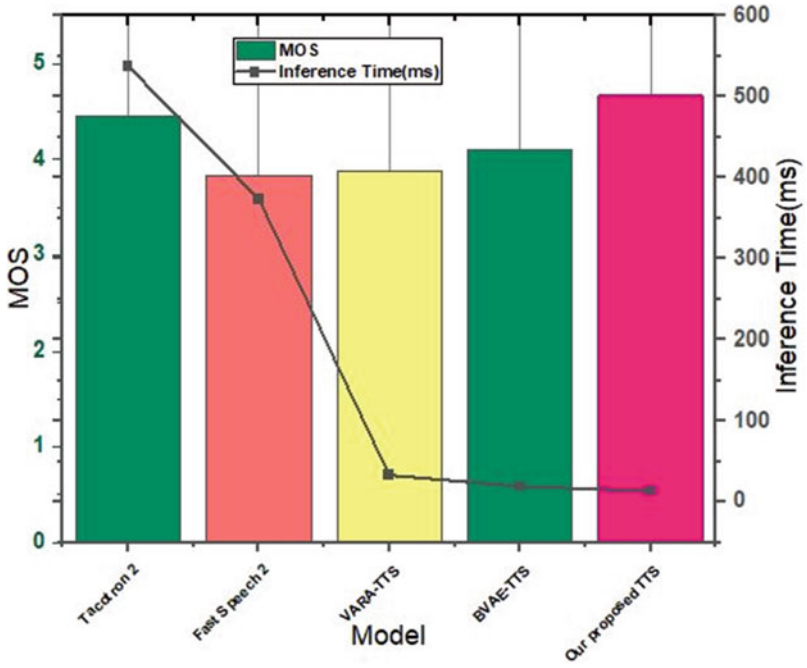

Fig. 7 A comparison plot of proposed model with existing TTS synthesizers

6 Conclusion

In this chapter, we have presented an efficient TTS synthesizer employing an encoder, synthesizer, decoder, and an additional reinforcement learning-based network. We utilized an existing structure of Tacotron2 for the synthesis of speech from text. More specifically, our proposed TTS synthesizer is comprised of three main stages such as (1) An encoder that generates a feature vector of speech from the dataset, (2) A synthesizer that predicts a Mel-spectrogram from grapheme sequences, conditioned on the vector from speaker embedding, and (3) A Vocoder based on autoregressive WaveNet that generates the waveforms from spectrograms. Furthermore, the RL block is attached consisting of CNN, BiLSTM, attention, FC, and softmax layer. Then, the accuracy of the RL network is considered as a reward value for the training. The proposed TTS synthesizer is based on a reference audio signal being fed to the encoder. Additionally, the quality of the synthesized speech varies according to the length, therefore short utterances provide better naturalness. We assessed our proposed algorithm using MOS, which vary from 1 to 5 such as Excellent, Good, Fair, Poor, and Bad respectively. In total, 100 participants were assigned the task to listen the audio and assign MOS to a generated speech by our proposed synthesizer. We attained 4.67 MOS overall which is a significant number than achieved from existing TTS synthesizers.

Although, our proposed synthesizer is significant for three languages i.e. English, Chinese, and French, however, we aim in the future to make it more generalized TTS synthesizer which can generate maximum natural languages exhibiting naturalness and similarity to human speech.

Acknowledgement The researchers want to thank University of Engineering and Technology Taxila to provide research environment.

References

1. Mishra R, Tripathi SP (2021) Deep learning based search engine for biomedical images using convolutional neural networks. Multimed Tools Appl 80(10):15057–15065
2. Parah SA, Sheikh JA, Ahad F, Bhat GM (2018) High capacity and secure electronic patient record (EPR) embedding in color images for IoT driven healthcare systems. In: Internet of things and big data analytics toward next-generation intelligence. Springer, Cham, pp 409–437
3. Hurrah NN, Parah SA, Sheikh JA (2020) Embedding in medical images: an efficient scheme for authentication and tamper localization. Multimed Tools Appl 79:21441–21470
4. Sarosh P, Heidari AA, Muhammad K (2021) Secret sharing-based personal health records management for the internet of health things. Sustain Cities Soc 74:103129
5. Ahad F, Bhat GM (2015) On the realization of robust watermarking system for medical images. In: 2015 Annual IEEE India conference (INDICON), New Delhi, pp 1–5. https://doi.org/10.1109/INDICON.2015.7443363
6. Mahum R et al (2022) A novel framework for potato leaf disease detection using an efficient deep learning model. Hum Ecol Risk Assess: Int J 29:1–24
7. Mahum R et al (2021) A novel hybrid approach based on deep CNN features to detect knee osteoarthritis. Sensors 21(18):6189
8. Mahum R et al (2021) A novel hybrid approach based on deep CNN to detect glaucoma using fundus imaging. Electronics 11(1):26
9. Korzekwa D et al (2022) Computer-assisted pronunciation training—speech synthesis is almost all you need. Speech Comm 142:22–33
10. Korshunov P et al (2016) Overview of BTAS 2016 speaker anti-spoofing competition. In: 2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS). IEEE, New York
11. Wu H et al (2020) Defense against adversarial attacks on spoofing countermeasures of ASV. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
12. Wu D (2019) An audio classification approach based on machine learning. In: 2019 International conference on intelligent transportation, big data & smart city (ICITBS). IEEE, Los Alamitos
13. Todisco M et al (2019) ASVspoof 2019: future horizons in spoofed and fake audio detection. arXiv preprint arXiv:1904.05441
14. Dinkel H, Qian Y, Yu K (2018) Investigating raw wave deep neural networks for end-to-end speaker spoofing detection. IEEE/ACM Trans Audio Speech Lang Process 26(11):2002–2014
15. Chintha A et al (2020) Recurrent convolutional structures for audio spoof and video deepfake detection. IEEE J Sel Top Signal Process 14(5):1024–1037
16. Lavrentyeva G et al (2019) STC antispoofing systems for the ASVspoof2019 challenge. arXiv preprint arXiv:1904.05576
17. He K et al (2016) Identity mappings in deep residual networks. In: European conference on computer vision. Springer, Berlin

18. Alzantot M, Wang Z, Srivastava MB (2019) Deep residual neural networks for audio spoofing detection. arXiv preprint arXiv:1907.00501
19. Lai C-I et al (2019) ASSERT: anti-spoofing with squeeze-excitation and residual networks. arXiv preprint arXiv:1904.01120
20. Monteiro J, Alam J, Falk TH (2020) An ensemble based approach for generalized detection of spoofing attacks to automatic speaker recognizers. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
21. Verma NK et al (2015) Intelligent condition based monitoring using acoustic signals for air compressors. IEEE Trans Reliab 65(1):291–309
22. Wu Z et al (2015) Spoofing and countermeasures for speaker verification: a survey. Speech Commun 66:130–153
23. Wu Z et al (2016) Anti-spoofing for text-independent speaker verification: an initial database, comparison of countermeasures, and human performance. IEEE/ACM Trans Audio Speech Lang Process 24(4):768–783
24. Chao Y-H et al (2008) Using kernel discriminant analysis to improve the characterization of the alternative hypothesis for speaker verification. IEEE Trans Audio Speech Lang Process 16(8):1675–1684
25. Ze H, Senior A, Schuster M (2013) Statistical parametric speech synthesis using deep neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, Piscataway
26. Dörfler M, Bammer R, Grill T (2017) Inside the spectrogram: convolutional neural networks in audio processing. In: 2017 international conference on sampling theory and applications (SampTA). IEEE, Piscataway
27. Balamurali B et al (2019) Toward robust audio spoofing detection: a detailed comparison of traditional and learned features. IEEE Access 7:84229–84241
28. Chao Y-H (2014) Using LR-based discriminant kernel methods with applications to speaker verification. Speech Comm 57:76–86
29. Yaman S, Pelecanos J (2013) Using polynomial kernel support vector machines for speaker verification. IEEE Signal Processing Lett 20(9):901–904
30. Loughran R et al (2017) Feature selection for speaker verification using genetic programming. Evol Intel 10(1):1–21
31. Zhao H, Malik H (2013) Audio recording location identification using acoustic environment signature. IEEE Trans Inf Forensics Secur 8(11):1746–1759
32. Handley Z (2009) Is text-to-speech synthesis ready for use in computer-assisted language learning? Speech Comm 51(10):906–919
33. McCoy KF et al (2013) Speech and language processing as assistive technologies. Comput Speech Lang 27(6):1143–1146
34. Shen J et al (2018) Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
35. Alghoul A et al (2018) Email classification using artificial neural network. Int J Acad Dev 2(11):8–14
36. Yang S et al (2015) From facial parts responses to face detection: a deep learning approach. In: Proceedings of the IEEE international conference on computer vision. IEEE
37. Dhamyal H et al (2021) Fake audio detection in resource-constrained settings using microfeatures. Proc Interspeech 2021:4149–4153
38. Ng H-W et al (2015) Deep learning for emotion recognition on small datasets using transfer learning. In: Proceedings of the 2015 ACM on international conference on multimodal interaction. ACM
39. Wan L et al (2018) Generalized end-to-end loss for speaker verification. In: 2018 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
40. Oord AVD et al (2016) Wavenet: a generative model for raw audio. arXiv preprint arXiv:1609.03499

41. Panayotov V et al (2015) Librispeech: an ASR corpus based on public domain audio books. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
42. Honnet P-E et al (2017) The SIWIS French speech synthesis database? Design and recording of a high quality French database for speech synthesis. Idiap
43. Wang D, Zhang X (2015) Thchs-30: a free chinese speech corpus. arXiv preprint arXiv:1512.01882
44. Variani E et al (2014) Deep neural networks for small footprint text-dependent speaker verification. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
45. Heigold G et al (2016) End-to-end text-dependent speaker verification. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Piscataway
46. Arık SÖ et al (2017) Deep voice 2: multi-speaker neural text-to-speech. In: Proceedings of the 31st international conference on neural information processing systems. Curran Associates Inc., Long Beach, California, pp 2966–2974
47. Wang X et al (2020) ASVspoof 2019: a large-scale public database of synthesized, converted and replayed speech. Comput Speech Lang 64:101114
48. Griffin D, Lim J (1984) Signal estimation from modified short-time Fourier transform. IEEE Trans Acoust Speech Signal Process 32(2):236–243
49. Rec I (1996) P. 800: methods for subjective determination of transmission quality. International Telecommunication Union, Geneva, p 22
50. Elias I et al (2021) Parallel tacotron 2: a non-autoregressive neural TTS model with differentiable duration modeling. arXiv preprint arXiv:2103.14574
51. Ren Y et al (2020) Fastspeech 2: fast and high-quality end-to-end text to speech. arXiv preprint arXiv:2006.04558
52. Liu P et al (2021) VARA-TTS: non-autoregressive text-to-speech synthesis based on very deep vae with residual attention. arXiv preprint arXiv:2102.06431
53. Lee Y, Shin J, Jung K (2020) Bidirectional variational inference for non-autoregressive text-to-speech. In: International conference on learning representations