iKSP Journal of Computer Science and Engineering
http://iksp.org/journals/index.php/ijcse/index
iKSP Publisher

A Deep Learning Framework for Efficient High-Fidelity Speech Synthesis: StyleTTS

Ather Fawaz*, Mubariz Barkat Ali, Muhammad Adan, Malik Mujtaba, Aamir Wali
Department of Computer Science, FAST - NUCES, Lahore Campus

Abstract - As we transition into the age of Artificial Intelligence (AI), one of the most incredible feats that it has achieved is the ability to talk and engage with human beings. An integral part of this task, referred to as speech synthesis, is to make the computer sound more human. Currently, the generative adversarial networks (GANs) have emerged as effective generative models and have more or less dominated the image generation domain. However, there is still untapped potential in what they can offer in the audio domain. Due to their highly parallelizable structure, GANs can produce hours of audio within seconds. Moreover, their inherent nature of modelling the latent space can afford some artistic control as well. In this paper, we propose Style text-to-speech (StyleTTS) model that uses image-based StyleGAN to efficiently generate high-fidelity speech. The model takes a character string and generates the corresponding speech. In this paper, we present results for digits from zero to nine. We also compare our results with older TTS approaches and GAN models and report gains by using the newer architecture. We intend to provide our pre-trained GAN to the open-source community in the form of a library. Upon release, this would be trained on audio samples of spoken English statements by various speakers. The library will be designed in such a way that it can be easily extended by researchers on more data. It will be simple for practitioners, and fast and robust in industrial deployments.

Keywords: Deep Learning, Artificial Intelligence, GAN, Open-source, Audio-synthesis, Text-to-Speech

INTRODUCTION

Speech is a vital part of human life. It is the primary method humans use in order to effectively communicate with each other on a daily basis. In order to understand how we can generate speech; we must first understand what are the constituent parts that make up speech. Speech can be thought of as soundwaves of varying wavelength, amplitude, timbre, and phase. To produce comprehensible speech, we concatenate distinct units of sounds together in order to form words. These units are known as phonemes. Later in this paper, we will see how we can use both these ways of thinking in generating speech.

The two major approaches for speech generation within deep learning are autoregressive models (AM) and generative adversarial networks (GANs). AMs are based on time series prediction, these models use previous input information to predict the next outputs linearly for each input unit (Oord & Dieleman, 2016). The output quality is at par with natural speech but due to the nature of this process it takes an extensive amount of time to produce complete output. To overcome this issue a parallel WaveNet was developed. In the parallel WaveNet authors parallelized the original WaveNet, enabling it to produce speech at 20x the speed of real-time (Driessche & Lockhart, 2017). Parallel WaveNet achieved this speed by introducing Probability Density Distillation, a new method for training the parallel feed-forward network from the original trained WaveNet. Importantly, the changes in the feed-forward network led no significant difference in quality and given its impressive speech, it has been deployed online for general public use on Google Assistant with support for English and Japanese voices.

*Corresponding author:
Email: L174004@lhr.nu.edu.pk (A. Fawaz),

On the other hand, GANs are one of the bleeding-edge techniques in deep learning today. With their ability to generate high quality data efficiently, they have come to dominate image generation. The advent of GANs has introduced new domains of research like deepfakes and synthetic content creation. Researchers have developed sophisticated GAN architectures that can create realistic paintings, photographs, music, etc. Likewise, they can augment specific areas of interest and realistically morph faces in videos. This has specifically spurred controversy and debate in the community about the potential threats of misinformation and disinformation that can result from the mainstream usage of GANs, and, by extension, deepfakes. Additionally, NVIDIA has also launched an GAN based product for improved video quality with low bandwidth connection: an important need for today's time. This shows the potential for GANs to be used in real world applications.

However, there is still untapped potential in what they can offer in the audio domain. Much of the research and applications utilizing GANs have been restricted to the image and video generation, leaving substantial room for exploration in the audio domain. Moreover, given their inherent, highly parallelizable internal structure, the inference or forward-pass in GANs is quick. To illustrate, some approaches that we shall see in subsequent chapters, have reported that GANs can produce hours of audio within seconds. To add to this, their inherent nature of modelling the latent space can afford some artistic control as well. Additionally, GANs are capable of learning significantly complex data distributions (provided with enough data), which enables them to generalize better.

For clarity, the approach for producing audio using GANs can be further subdivided into acoustic and duration model, and spectrogram model. The acoustic and duration model uses phonemes and sound duration as features. The input word is decomposed into its constituent phonemes. Each phoneme will have its accompanying time duration with it. The network then trains itself using these features and produces output by concatenating different phonemes together. Meanwhile, spectrogram GAN or SpecGAN for short, uses an image generating model deep convolutional GAN (DCGAN) to generate spectrogram images that can further be turned into audio. The main advantages for this approach are that we do not have to rely on phonemes in the textual input in order to get our features

Recently, a new image generating technique has been proposed called Style GAN or StyleGAN (Karras et al., 2018, 2020). This can be considered as the next generation of DCGAN. StyleGAN improves the generator model. Earlier works mainly focused on the discriminator, and the generator model was used as a black box. In this paper, we propose StyleTTS that uses StyleGAN as a spectrogram-image generating model to generate speech. Usually, GANs would take a random noise and generate any data instance. In this paper, we also incorporate a context or conditional information to force the GAN to produce a specific word. Our model displays ability to generate controllable, high fidelity, and fast speech commands.

The rest of the paper is organized as follows. Section 2 presents a detailed literature review on GANs. Our proposed model is given in section 3 followed by results and discussion in section 4. Conclusion and future directions are presented in section 5.

LITERATURE REVIEW

This section will include an introduction generative adversarial network. Subsequent subsections will review domain-specific research work that ties in with the domain of our project.

Generative Adversarial Networks (GAN)

A typical GAN is made up of two networks, a discriminator and a generator as shown in figure 1 (Rahul, 2020). The discriminator is a convolutional classifier while the generator is a deconvolutional network that translates noise to a data instance.

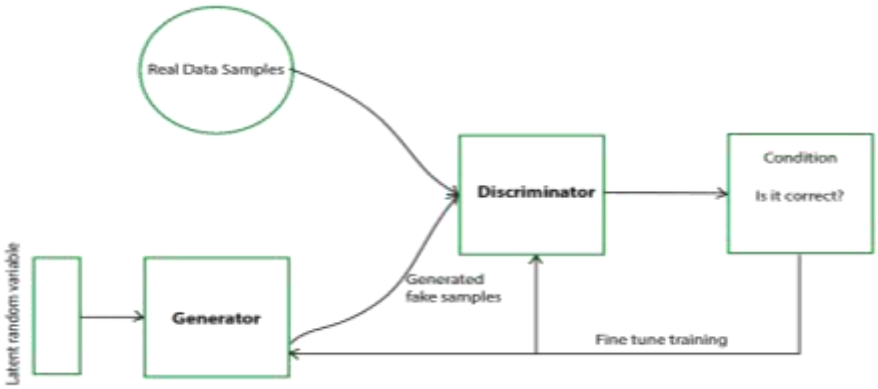


Figure 1: A typical GAN consists of two neural networks competing with each other (Rahul, 2020)

Both networks are trained simultaneously in the form of a minmax game. The generator aims to generate images which would maximize the binary cross entropy loss (BCE) for the discriminator and the discriminator tries to minimize this loss. Figure 2 (Goodfellow et. al., 2014) shows how a generator tries to map a random noise from one distribution to another. In the figure the blue dots represent discriminative distribution, the generated distribution (from mapping of z noise vector) is shown using green lines and the black dots show the actual data distribution. Optimal value is achieved when the generated distribution maps perfectly on the data distribution resulting in loss of improvement in discriminative distribution as the discriminator can only guess for the correct answer. Using Jensen–Shannon divergence (JS-divergence) it is proven that the optimal value of this min max game exists when the distribution of generated data and input data is the same.

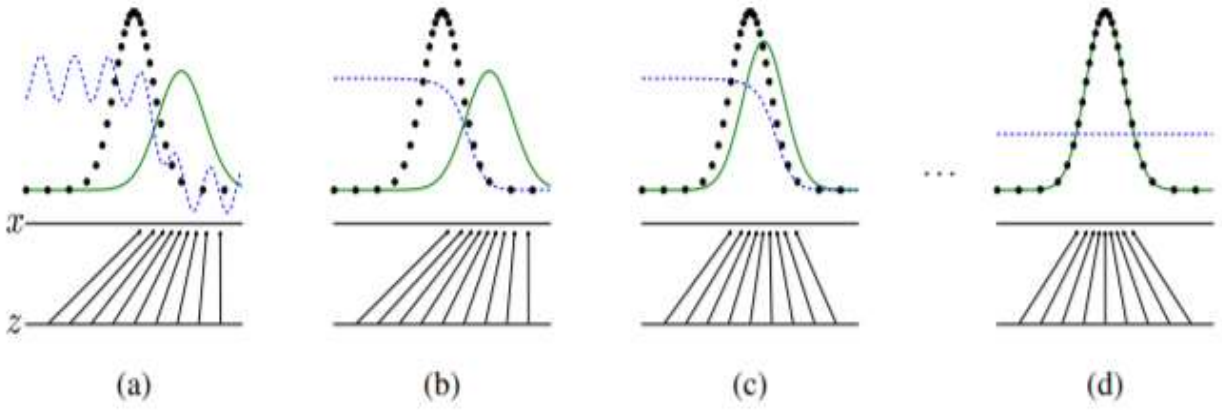


Figure 2: Mapping one distribution onto another. The generator tries to mimic the data distribution by continuously morphing its own (Goodfellow et. al., 2014).

The results produced by these networks have been remarkable, they had SOTA results at that time and resultantly sparked a whole generation of neural networks for generative modeling. The greatest advantage of these neural networks is their ability to produce accurate representation quickly, maintaining most of the information from the original data distribution. Though it should be noted GANs are difficult to train due to their Min Max nature and they are prone to mode collapse (and issue when Generator learns a trick to fool Discriminator and optimizes that trick instead of mapping to the original data distribution.

Audio Synthesis using Generative Adversarial Networks

Generative Adversarial Networks (GANs) have demonstrated immense potential in working with images but have seen limited usage in the audio-visual domain until recently. In an industry-first, Donahue, McAuley, et al. from UC San Diego applied the venerable DCGAN (Deep Convolutional GAN) for high-fidelity audio synthesis. Their results rivaled the state-of-the-art autoregressive models like WaveNet in accuracy and eclipsed them in generation speeds (Donahue et al., 2019); (Oord & Dieleman, 2016).

Concretely, the researchers proposed two separate architectures – SpecGAN and WaveGAN. SpecGAN took the time-frequency representations of sounds, called *Spectrograms*, and fed them into a DCGAN. The intuition behind this effort was to treat these spectrograms as traditional images and to train a generator to learn their latent space. Notwithstanding, the authors had to circumvent certain invertibility issues associated with spectrograms, so they first had to design a spectrogram representation with approximate inversion (pseudo-lossless) and then had to bootstrap it into the DCGAN for training. By learning to produce sounds from birds, pianos, drums, etc., SpecGAN demonstrated impressive results with an inception score of 3.71. The novel semi-invertibility technique also proved its efficacy.

WaveGAN's technique took a slight departure from the image domain and flattened DCGAN's images into a one-dimensional sequence of values to represent audio signals. One important point to note here is that while audio signals, especially instruments and music, demonstrate repeating patterns, the challenge here was to keep the generator from learning repeating patterns in sounds and subsequently fixate on those examples, a problem known as mode collapse. The authors devised a clever technique – phase shuffle – that essentially threw off the generator by *intelligently* shuffling the sequence of values. With this setup in place, the trick was then to carry out convolution on the 1-D array of values using a sliding window with a certain *stride* as before. Once training was completed, on paper, WaveGAN proved to be even superior to SpecGAN, boasting an inception score of 4.12.

The researchers also trained both architectures on TIMIT, which is a large dataset containing words from the English language (Garofolo, 1993). Both architectures learned the latent space well and humans correctly labeled the synthesized words 58-66% of the time, depending on the choice of model.

As mentioned before, due to the extensively parallelizable nature of GANs, both WaveGAN and SpecGAN are extremely fast, capable of producing hours of sounds within a few seconds. This is a massive gain over the state-of-the-art sequential autoregressive models like WaveNet, which are more accurate but much slower.

Recreating Audio from Spectrograms

Spectrograms are a handy conversion between audio and images. Hence, the creation of SpecGAN was one of the key developments in the field of generative modelling. SpecGAN relies on images of spectrograms to be able to train and produce equivalent audios. These spectrograms contain information such as amplitude and frequencies at different time intervals.

Spectrograms are created by dividing a longer time signal down to equal-length shorter segments or windows. Fourier transform is then computed on each of these segments, which basically means that we are decomposing the sound signal down to its constituting frequencies called the Fourier spectrum. The magnitude of this spectrum is taken which makes each segment correspond with a vertical line. We then take the magnitudes of all the segments and plot them onto a frequency-time graph, which is called a spectrogram. The implementation corresponds with computing the squared magnitude of a Short-Time Fourier Transform, STFT, of our audio signal.

However, taking a squared magnitude causes the spectrogram to lose phase information of the audio, which further prevents us from recreating the original audio accurately and without loss of information. SpecGAN tries to circumvent this limitation by using the algorithm proposed by Griffin & Lim in 1984 (W & S, 1984, 236-242). The algorithm uses a Modified Short Time Fourier Transform, MSTFT, to estimate an audio signal. Its main objective is to shorten the Mean Squared Error, MSE, between the STFT of the estimated signal and the MSTFT.

The algorithm achieves this objective by iteratively minimizing the MSE between magnitudes of the STFT of the estimated audio signal and the MSTFT. This produces high quality modified time-scale sounds which means that sounds are sped up or slowed down without affecting their frequency content such as perceived pitch.

Although this algorithm is a step in the right direction, it still has its pitfalls. Its error surface is plagued by local minima, which makes it harder to guarantee that it will always provide a high-quality reconstruction of the audio signal. The other limitation is that it is slow since it is iterative in nature and must make iterations over forward and inverse STFTs along with needing the entire audio duration to reconstruct each point in time (Wyse, 2017). These limitations hamper SpecGAN when it comes to audio reconstruction and real-time performance. In later sections, we will see how to bypass such limitations.

Analyzing and Improving the Image Quality of StyleGAN

StyleGAN introduced the style transfer techniques used previously in Deep Learning to the GANs, resulting in greater control over latent variables and consequently the features of the output.

For this network NVIDIA created a new Full HD dataset with greater variation in data distribution. There are exactly 70,000 images at 1024 resolution with a much greater variation in age, race, color, clothes, and background. This network discards the old style of generating images from a random latent space, instead it provides a new method which provides learned mapping of latent space to the generator. The authors use learned affine transformations to stylize input vectors. Finally, the input is normalized and combined with noise at the pixel level and sent to the Progressive Generator. The progressive generator starts by learning features in lower first and incrementally increasing dimensions until a certain number of steps resulting in efficient and better learning of complex data distribution.

The AdaIN operations ensure that the weights are normalized before applying each style ensuring the style control remains within convolution operation. The authors applied a technique of style mixing. They generated two latent vectors from different distributions and alternated both to ensure style localization.

Furthermore, addition of noise at pixel level ensures that the generator does not prioritize relative pattern. Resulting in the greater variety of Images generated. This noise improves the quality of image background and increases its detail as well. In conclusion, style-based generators are far superior to traditional GAN architects, in terms of metrics image quality and Image control.

High Fidelity Speech Synthesis with Adversarial Networks

Text-to-Speech GAN(TTS-GAN), as the name suggests, is specifically attuned to human speech unlike its predecessors which were geared towards more general audio. The dataset is specifically prepared for this task. Instead of operating on the raw audio waveforms or spectrogram representation, it takes an approach that, on paper, should be ideal for

speech. The linguistic features, which encode phonetic and duration information, and pitch are calculated. These features are calculated using a separate model.

The generator network is a feed-forward convolutional neural network (CNN). It is fed linguistic features and pitch at 200 Hz and it outputs a raw audio waveform at 24,000 Hz (24k Hz). It uses dilated convolutions to capture long term dependencies and the dilation factor increases the deeper the model goes (Fischer & Koltun, 2015).

Instead of a single discriminator, an ensemble of discriminators is used which operate by selecting random windows from the original sample and evaluating them instead of the whole sample. There are two types of discriminators, conditional and unconditional. The conditional discriminators are fitted with the linguistic and pitch features so they can evaluate the sample with respect to the input conditioning. The unconditional discriminators are only concerned with the realism of the sample regardless of whether it corresponds to the input conditioning or not. In the paper, ten discriminators were employed with varying window sizes determined experimentally. This ensemble configuration of discriminators with random window sizes is one of the key features of this architecture.

All models that used a collection of discriminators produced better results than a single conditional discriminator. A full discriminator that evaluated a sample in its entirety performed the worst of all configurations. Utilizing only fixed sized windows was detrimental to the results, a mixture of both unconditional and conditional discriminators with varying window sizes proved to be beneficial. The authors conjecture the reason as to why small random window sizes work better than considering the whole sample is because of the relative simplicity of the underlying distribution. The architecture affords very stable training as it is noted that not a single mode collapse occurred during as many as one million steps and the efficiency of the model increased gradually.

The results produced by the best model were worse but comparable to SOTA, WaveNet and Parallel WaveNet results. Mean Opinion Score and Fréchet distance were used as evaluation metrics along with some other novel metrics like the DeepSpeech distance that the paper proposed.

SUMMARY

A concise summary of prominent GAN architectures is given in table 1.

Table 1: A summary of prominent GAN architectures is presented here.

No	Name	Year	Generates	Description
1.	Generative Adversarial Network (GAN)	2014	Data, usually in the spatial domain	The first demonstration and proof of concept for data generation by modelling an implicit data distribution using a pair of adversarial networks which essentially play a minimax game
2.	Conditional GAN (CGAN)	2014	Data, usually in the spatial domain	Proof of concept that the generator and the discriminator of the GAN can be conditioned on an input variable to produce results from a conditional probability distribution
3.	Deep Convolutional GAN (DCGAN)	2015	Data, usually in the spatial domain	Using a Convolution based architecture for GANs
4.	Wasserstein GAN (WGAN)	2017	Data, usually in the spatial domain	Replaces the Evaluation function like Jensen-Shannon Divergence with Wasserstein distance.
5.	SpecGAN	2018	Data in the audio domain	Convert audio samples into spectrograms and training on DCGAN. Proof of concept that the distribution of audio data can be modelled by training GANs optimized for spatial data on the spectrograms of the audio
6.	WaveGAN	2018	Data in the audio domain	Essentially a flattened version of DCGAN, configured to work on raw audio waveforms directly in the time-frequency domain
7.	TTS-GAN	2019	Human speech	A GAN that is fed an encoding of linguistic features and pitch. It is evaluated by an ensemble of discriminators and produces results comparable to SOTA models
8.	Style-GAN	2018	Images	A GAN architecture that employs a novel technique of progressively increasing the image resolution during training. This results in a high-quality image. This model currently is the state of the art

PROPOSED MODEL

Our proposed StyleTTS model draws inspiration from the SpecGAN (Donahue et al., 2019) model and is a direct successor to it. The method proposed in SpecGAN consists of employing the image generation capabilities of DCGAN (Radford et al., 2016). The dataset consisting of audio was preprocessed into 128x128 spectrograms. DCGAN was trained on these spectrograms. Finally, once the model was trained it was sampled and the generated spectrograms were inverted to get an audio waveform. Our proposed model builds on this and replaces the DCGAN with the improved and efficient StyleGANv2 (Karras et al., 2020) which is the state-of-the-art model in image generation. This led to generating more accurate spectrograms which in turn led to higher fidelity audio. However, SpecGAN was unconditionally trained and samples were randomly produced. To condition our model to produce samples relating to specific classes, the technique proposed in the CGAN is utilized for conditional generation.

Data Preprocessing

Looking closely at this process and taking it sequentially, the training data provided to this GAN was labelled and consisted of two parts – a Mel spectrogram and its transcription. This training data was preprocessed via a data loader class that encoded and embedded this Mel spectrogram as well as the characters in a tensor which was fed into the generator of the GAN. Formally, this generator consists of two parts, an encoder, and a decoder. The encoder handled the preprocessing and generated the embeddings in a tensor.

Synthesizing

Once encapsulated, these embeddings were passed on to a decoder. The decoder took these embeddings and tried to learn the relationship between a sample audio (or more specifically, its Mel spectrogram) and its equivalent transcription. Using this learned relationship, the decoder synthesized the Mel spectrogram for a given sequence of characters that were passed as a tensor to it. The synthesis was done via a spectrogram-inversion algorithm. This was done via the Griffin-Lim inversion algorithm for an arbitrary spectrogram.

Minimax Game

This synthesized output was given as input to the discriminator or *critic* (depending upon the choice of the loss function used in the GAN training) that passed a verdict on whether it is a synthesized output or one from the real training data distribution itself. Feedback from the discriminator and its success at telling fake data apart from the real one updated each neural network, that is, the generator and the discriminator itself. As stated before, eventually, a Nash Equilibrium was reached, and the generator became so adept at producing high-fidelity audio for a given sequence of characters that the discriminator's accuracy plateaued at 50%, at which point it was randomly guessing the difference between real and fake data.

Apropos the loss function, compared to the vanilla GAN, we are using the improved Wasserstein distance to calculate the loss. The Wasserstein distance will calculate the distance, or the degree of dissimilarity, between two probability distributions, and therefore, help the GAN learn the latent distribution of the training data. It has demonstrated various improvements over the vanilla loss function, which was based on the traditional binary cross entropy loss. Most importantly, the Wasserstein distance gives us meaningful feedback, in terms of loss, to track how well our GAN is training. This was not always possible with the traditional loss function. In addition to this, this metric also improved the stability of the optimization process during training. So, in nomenclature, whenever we use Wasserstein distance, it is common practice to call the discriminator a critic.

Model Pipelines

After discussing the individual components in the previous subsections, the model pipeline is given in figure

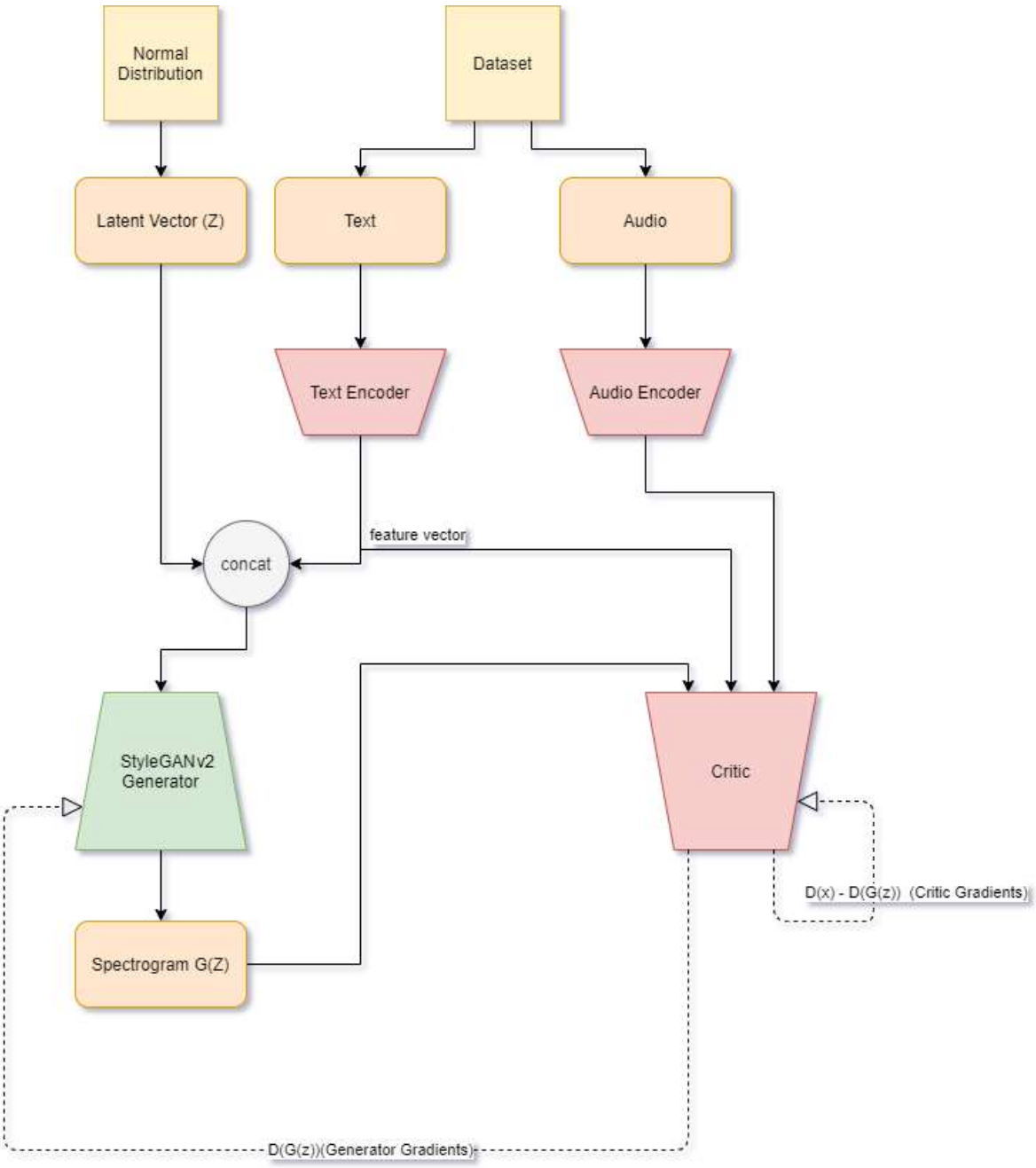


Figure 3: Pipeline of the proposed StyleTTS model. Latent vector is initialized from 100 dimensional random normal distribution and concatenated with encoded text vector. Finally this information is sent to our StyleGANv2 generator which generates and spectrogram. This spectrogram along with feature vector and ground truth spectrogram(generated from audio encoder) is sent to Critic to evaluate loss. Finally this loss is used to optimize both Generators and Critic.

RESULTS

In the last section we presented our proposed StyleTTS model. In this section a demonstration of our model is given. We first begin with the experimental setup.

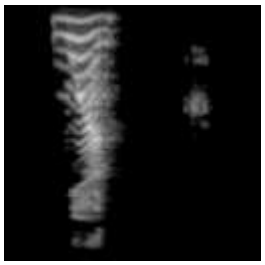
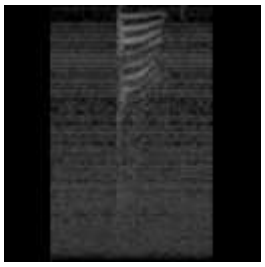
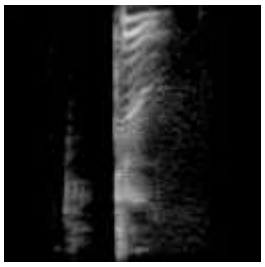
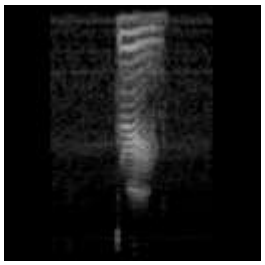
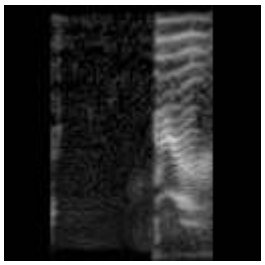
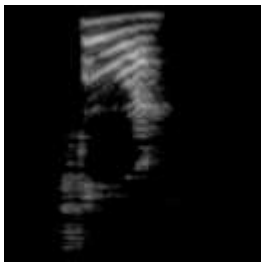
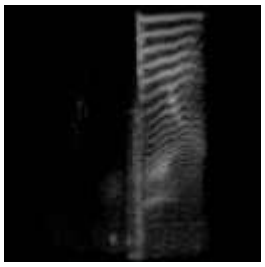
Experimental Setup

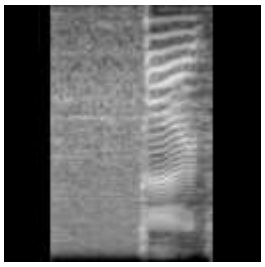
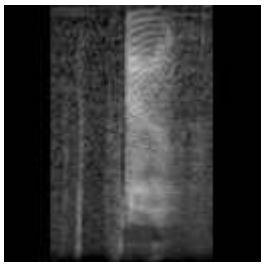
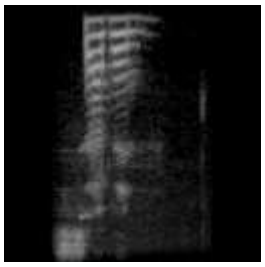
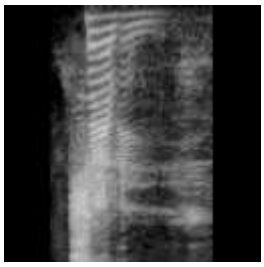
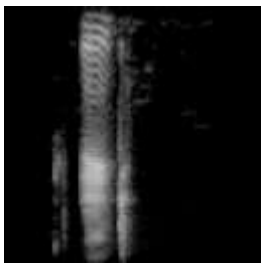
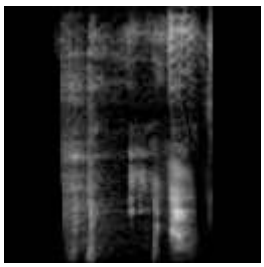
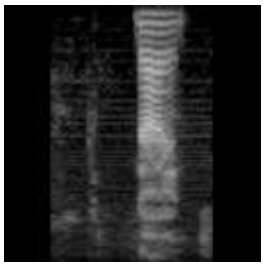
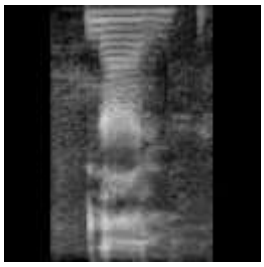
We used the speech command dataset (Pete, 2018) to train our StyleTTS model. The dataset has 35 classes. In this paper, we mainly used the trained model to generate words from zero to nine. The data set contains audio clips of utterances which are approximately one second long. These raw audio files were preprocessed into low resolution spectrogram images of size 128x128 with 128 mel bins and the rest of image time padded to reach 128 dimensions. Thus, the size of resulting spectrogram is also 128x128.

Spectrogram Generation

The spectrograms generated using StyleTTS for words zero to nine are given in table 2. The table also shows the real spectrogram. In all generated spectrograms besides zero, it can be seen that the vowels significantly match the vowels in the original spectrogram. Also, fricative like \f\ has high fidelity in the word five.

Table 2**:** Real Mel-Spectrograms vs. Generated

Words/labels	**Original spectrogram**	**Generated spectrogram**
Zero		
One		
Two		
Three		
Four		

Five		
Six		
Seven		
Eight		
Nine		

Speech Quality Testing

We compare our generated audio files with those from SpecGAN. Our model demonstrated better results owing to the use of the StyleGANv2, instead of its predecessor DCGAN.

For this purpose, we crowd-sourced mean opinion score (MOS) tests via an online form. On the form, a set of samples were played back to the people without telling them which model the sound files originated from. The participant over the age of 20, also assessed the speech quality. From the data collected from initially, 20 people, the results are quite promising and in most of the cases our model's speech output was judged having higher quality. The metric of quality is audible, clear and perception.

CONCLUSION AND FUTURE WORK

In this research, we present an end-to-end text-to-speech synthesis that is capable of generating spoken digits from zero to nine. Compared to previous works (Donahue et al., 2019), our results demonstrate the high-fidelity of our GAN. As

stated before, this is because we have swapped out the original SpecGAN's DCGAN with StyleGAN-2 from NVIDIA. Since StyleGAN-2 is better at image synthesis than DCGAN, our spectrogram generation and subsequent inversion is better too, leading to improved results in speech generation. This provides proof of concept that improving the GAN architecture equates to higher fidelity speech.

However, our work is limited to only digit generation. That is majorly due to the chosen one-hot vector encoding scheme that is input to the GAN. By replacing this one-hot vector with a robust model like the Tacotron2 encoder, we will be able to extract more features and pass them on to the GAN to learn. We have already started working on this, and moving forward, the full model will take any sequence of characters and convert it into speech, not just numbers from 0 to 9. We plan to package our complete work with the Tacotron2 encoder and a pre-trained model in the form of an end-to-end TTS library founded upon generative adversarial networks. The library will be designed in such a way that it can be easily extended by researchers on more data. It will be simple for practitioners, and fast and robust in industrial deployments.

Furthermore, our framework of choice for all work (present and future) is PyTorch Lightning. It was recently introduced in the latter half of 2020, and aims to abstract much of the existing PyTorch library to make development and deployment of machine learning models easier. Since PyTorch Lightning is still new, there are not many implementations of GANs in the framework. Our library, which we plan on completing and releasing soon, could prove to be an important development for PyTorch Lightning and GANs in general.

REFERENCES

Donahue, C., McAuley, J., Puckette, M.: Adversarial Audio Synthesis. arXiv preprint. https://arxiv.org/pdf/1802.04208.pdf (2019)

Driessche, G. V., Lockhart, E.: Parallel WaveNet: Fast High-Fidelity Speech Synthesis. International conference on machine learning, pp. 3918-3926 (2017)

Fischer, Y., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint. https://arxiv.org/pdf/1511.07122.pdf (2015) Garofolo, J. S.: TIMIT acoustic-phonetic continuous speech corpus. Linguistic data consortium (1993).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., WardeFarley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. Advances in Neural Information Processing Systems, 2672–2680 (2014)

Griffin, D., Lim, J.: Signal estimation from modified short-time Fourier transform. IEEE Transactions on Acoustics Speech and Signal Processing, 32, 236–242 (1984).

Karras, T., Laine, S., Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks. arxiv. https://arxiv.org/abs/1812.04948 (2018)

Karras, T., Laine, S., Aittala, M.: Analyzing and Improving the Image Quality of StyleGAN. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8110-8119 (2020)

Kominek, J., Black, A. W.: CMU ARTIC databases for speech synthesis. CMU-LTI-03-177. https://en.wikipedia.org/wiki/Speech_synthesis#cite_note-28 (2003)

Oord, A. v. d., Dieleman, S.: WaveNet: A Generative Model for Raw Audio. https://arxiv.org/abs/1609.03499 (2016)

Pete, W.: Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:1804.03209 (2018)

Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arxiv.org. https://arxiv.org/pdf/1511.06434.pdf (2016)

Rahul, R.: Generative Adversarial Network (GAN). https://www.geeksforgeeks.org/generative-adversarial-network-gan (2019). Accessed September 2020

Schwarz, D.: Current Research in Concatenative Sound Synthesis. Proceedings of the International Computer Music Conference (2005)

Wyse, W. L.: Audio Spectrogram Representations for Processing with Convolutional Neural Networks. Proceedings of the First International Conference on Deep Learning and Music, 37-41 (2017)