AttnGAN: Audio to Image Synthesis with Multistage Attentional Fine-Grained Text Representation*

Changsheng Yan
cya96@sfu.ca

Yifan Li
yla570@sfu.ca

Andong Ma
ama151@sfu.ca

Risheng Wang
rishengw@sfu.ca

Yi Xiao
yxa95@sfu.ca

Abstract

In this project, we try to improve the original AttnGAN[1] architecture by adding one more stage (one attention model, discriminator and generator), which is able to generate higher quality images with larger size (512x512) and more details. We apply our re-fined model on CUB dataset and compare with the visual and quantitative results from the original AttnGAN. Attempting to optimizing our model, we quantitatively evaluate its variants by changing multiple learning rates, weight of Deep Attentional Multimodal Similarity Model(DAMSM) loss and training step of generators in each mini-batch iteration. An inception score of 4.34 is obtained on CUB dataset, which is at a similar level with the original state-of-the-art AttnGAN[1]. Based on previous related work[1] [2] [3], we build an interactive application by adding a user interface and speech recognition component so that images can be generated easily from users' audio inputs. All our code is available on GitHub[4].

1 Introduction

1.1 Problem

Our idea is inspired by crime films in which police sketch criminal faces based on the witness's descriptions. In machine learning's world, it is not a dream that this could be done faster, more accurate and more easily if you had a powerful generative model. Generative models such as Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) are getting more and more powerful in recent years and various useful applications could be developed based on these generative models.

With an interesting and practical idea which can be used in multiple industries and demonstrating our project with an attractive interaction effect, we start our initial research of previous work. We tried to figure out a direct way of generating images from audio without text representation. However, the problem is that we need to record audio by ourselves, which could be tremendous work and may not able to have a good result. The alternative way through intermediate text representations allows us to make full use of the existing work, which are speech recognition and text-to-image generation. Text to image synthesis has been attempted and excavated by many people and it is still a potential research area currently. Therefore, our main focus is on figuring out a re-fined model based on

*Poster Name: Speak Out and Get Your Own Bird

[1]https://pypi.org/project/SpeechRecognition/

[2]https://gist.github.com/sloria/5693955

[3]https://github.com/taoxugit/AttnGAN

[4]https://github.com/scmadmad/SpeakOutAndGetYourBird

existing networks to generate images from text in this project. Audio to text is added here with the purpose of creating a dynamic and interactive effect for demo.

1.2 Basic Network Architecture Determination (Related Work)

We research different Generative Adversarial Networks(DCGAN[2], StackGAN[3], StackGAN++[4] and AttnGAN[1]) on achieving generating high resolution images. S. Reed et al. [2] proposed using deep symmetric structured joint embedding for visually-discriminative vector representation of text descriptions[5] as a condition for zero-shot generation of images, however with only one stage in the networks(one generator and one discriminator), the generated images are low-resolution and lack of necessary details. Later in H. Zhang et al.'s work, StackGAN [3] was proposed to generate larger size (256x256) and higher resolution images, based on multiple stacked generators and discriminators and Conditioning Augmentation technique. By modifying the structure of StackGAN, [4] proposed a tree-like structure GAN (StackGAN v2), which further improves quality of generated samples and stabilizes training process. However, these proposed models still lack the ability of correcting or generating details in different stages, because the condition of generating images solely relies on sentence embedding.

The Attentional Generative Adversarial Network(AttnGAN)[1] is a network proposed by incorporating an attention mechanism. For one thing, it refines problems raised from previous GANs by introducing attention-driven multi-stages networks, which enables the generator to generate images based on sentence embedding, and refine sub-regions of the image in the later stages, through a word-level attentional mechanism. Additionally, the DAMSM loss is designed to provide extra supervision, in order to help stabilize the training process. Also, the structure of AttnGAN is more comprehensive so there might be more different aspects to explore. Therefore, we try to improve the original AttnGAN and obtain our new model to generate larger size (512x512) of images in a more detailed manner.

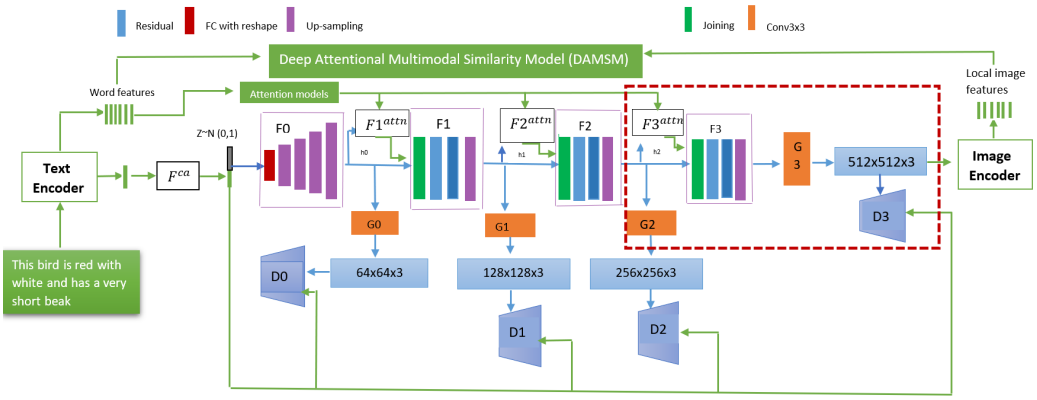

Figure 1: The architecture of AttnGAN (Components added are in the red rectangle)

2 Approach

2.1 Understanding the Attentional Generative Adversarial Network

The current Attentional Generative Adversarial Networks(AttnGAN)(see Figure 1) has two attention models stacked with three generators in the attention generative networks along with Deep Attentional Multimodal Similarities Model(DAMSM). It is able to generate distinct and realistic images within multiple stages. The detailed structure and work flow of AttnGAN are described as following.

In the feed-forward process, the text encoder, a bi-directional Long Short-Term Memory, takes the text description in and converts it into word features and a sentence feature by concatenating two different hidden states in each direction in the LSTM. The Conditioning Augmentation Network F^{CA}, inspired by VAE[5], then transforms the sentence feature to learn the mean and standard deviation

[5]https://github.com/pytorch/examples/blob/master/vae/main.py

vector, and gives a sampled global conditional vector, which is to improve the diversity of generated samples. Together with the noise vector, they are passed to the first generator for reshaping and upsampling (same functionality as DCGAN), with output of the first image feature (h_0) and first generated image (64*64). After the first stage, the attention model then will take image feature as query for the word embedding and convert them into a word-context matrix representation using an attentional mechanism. The word-context matrix represents dynamic word-context vector for each sub-region of the image. The more relevant the word to each sub-region is, the larger weights to emphasis of the word will be given. Then the word-context matrix is concatenated with the image feature(h_i) to generate larger size, as well as correct and improve details of the image in the following stages.

For training the AttnGAN, two parts of objective loss need to be introduced. The first part is the common min-max function[6] defined as previous GANs. The second part is the DAMSM loss[7], which measures match losses of text descriptions and generated samples. Therefore the objective function is defined as:

$$L = L_G + \lambda L_{DAMSM} \tag{1}$$

The DAMSM is a pre-trained component that has the same intention as [5], which is to learn a deep representation function in a multimodal semantic space, based on training images and their ground-truth descriptions. The main difference between [5] is that an attention mechanism is also used in DAMSM. With a text encoder as described above, and an CNN image encoder built based on Inception-v3 model, the DAMSM loss as an objective function for training DAMSM is used to measure similarity and correlation (called attention-driven image-text matching score[8]) between sentence embeddings and global image vectors, as well as word embeddings and sub-region image vectors. By pre-training DAMSM, the text encoder and image encoder are used for training the generators and discriminators.

Figure 2: Example generation processes and results of our model. The left shows the images from low-to-high resolution generated by different generators. The right shows the top-5 most attended feature words of the attention models respectively.

2.2 Improving by adding to its architecture

The current structure of AttnGAN is already hard to train primarily because the need of large GPU memory (11 GB for 20 batch-size). So in order to handle the new model using existing computing power we have (16GB on Google Cloud Platform's Tesla V100), we thought about different

[6]Eq. 4&5 in [1]
[7]Eq.7-Eq.14 in [1]
[8]Eq.10 in [1]

approaches to add minimal blocks of networks and see if 512x512 images can be generated. For instance, we could add only an upsampling block after F_2. We also thought about cutting the residual blocks in the current F_3, or cutting the discriminators in early stages. The training would be time-consuming so we would like to be cautious about our chosen approach.

We consult with TAs about the feasibility of some of our thoughts and finally decide to append a whole stage after the original AttnGAN model. After adding a third attentional model $F3^{attn}$, we join the previous stage's image features h_2 with the attentional word-context vector and send it to two residual blocks which could compensate details loss. And we add an additional upsampling function to set the output to be 512 pixels. When we modify the corresponding discriminator, we add an additional downsampling block to get the 4x4 image unit. After that, we use one convolution layer and then a sigmoid function to get the discriminator score. Also, we modify the connection of the DAMSM to our current last stage of generated images.

An example of image generation flow of our model is shown in Figure 2, we can see that the attentional mechanism is able to locate sub-region of the image according to different attention weights of the words(white area). Then refined and corrected images are generated for better quality in later stages. It is noticeable that after stacking another stage, the attention model $F3^{attn}$ can locate the sub-regions more accurately, compared with Figure 4 in [1], which is able to focus more precisely on the subsequent image generation.

3 Experiments

Our model is trained on CUB dataset[9](8,855 training and 2,933 testing samples, with 10 captions for each image), with hyperparameters set to the same as [1] (e.g. $\lambda = 5$, DISCRIMINATOR_LR=0.0002, GENERATOR_LR=0.0002), as a baseline performance for our experiments. The training process is extremely slow due to small batch-size of 7 (15 min for 1 epoch). After training 400 epochs for baseline model which performs worse and worse (see the 'baseline' line in Figure 3). We try to adjust different part of the hyperparameters. However, with limited number of virtual machines and amount of free credit on Google Cloud Platform, we only carry out several experiments on some most relevant hyperparameters. And due to the slow training, we only obtain experiment data for the first 200 epochs (see Figure 3 for full results).

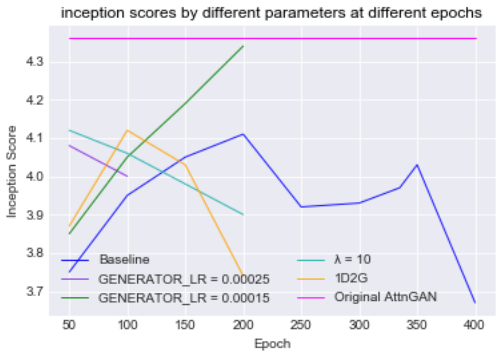

Figure 3: Inception scores for tuning different parameters at different epochs

3.1 Adjust generator learning rate(GENERATOR_LR)

The default generator learning rate in the baseline model is 0.00020. When training the baseline model, we found the training process is not stable. The discriminator loss is always low and generator loss gets higher and higher over epochs. In order to guarantee that our training process is purposive and meaningful but in an acceptable training duration, we decided to decrease the generator learning rate to 0.00015. The inception score reaches to 4.34±0.24 at only 200 epochs. This is the highest inception score we get so far, which is at similar level of mean with the origi-

[9]http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

nal AttnGAN(4.36±0.03)(See the 'Original AttnGAN' horizontal line in Figure 3). But with this upward trend, higher inception score and stability could be obtained in later epochs.

3.2 Re-weight L_{DAMSM}

λ (in Eq 1) is another hyper-parameter that controls the weight of the DAMSM loss in the entire objective function. Our initial perspective is to generate images with higher resolution and quality so we attempted to catch more details in each stage. We try to give more weights to the detailed features from the description at the word level for training the generator. Also, as shown from the experiments of testing λ in the paper of AttnGAN[1], a higher lambda leads to a higher inception score in some scope. Therefore, we choose λ to be 10 and the inception score under this value is 3.90±0.10 at 200 epochs.

3.3 Slacken discriminator training process

As mentioned in [6], the most basic idea of GAN is known as the two-player game between the generator and discriminator, in other words, an ideal start for both generator and discriminator is to be well-matched and powerful. However, in our baseline model, as mentioned the initial loss for the discriminator is quite small and generator loss becomes larger and larger. Some of the images generated by later-epoch models are in lower quality. Therefore, we train an additional model with different training steps, 1-step update for discriminators and 2-step update for generators in each mini-batch iteration. However, the result is not ideal. The inception score is 3.74±0.11 at 200 epochs(the '1D2G' line in Figure 3).

3.4 Generalization ability

Figure 4 shows the generalization ability of our model by changing the most attentional words in a sentence. Higher resolution of the images can be obtained compared with Figure 5 in [1].

4 Application

4.1 User Interface

Our application mainly contains a main interface and two sub interfaces.

The main panel(see Figure 5) provides users with an entry to get their inputs in and it simply contains two buttons that route users to the two sub interfaces, which allow them to either speak to the microphone or type in their descriptions of a bird. The sub interface for speech input looks like (Figure 6). After clicking on Speaking button, the user is routed to the speech input interface. It contains a recording session, by which user's speech is recorded and passed to Google Speech Recognition unit. Users could control their start and end of speech recording by clicking on Start button and Stop button. They are routed to text input sub interface by clicking on right most Show button. Sub interface for keyboard input is just like(Figure 7) and can be accessed by directly clicking on the Keyboard button, or after speech recording. It is used to manually type in or adjust user's description of birds and send the final description to our model to produce an image. Figure 8 shows an example of a set of generated images in our application.

4.2 Methodology

We adopt Tkinter[10], Speech Recognition[11] and Pyaudio Recording[12] to build our application. Implementaion details can be found in on GitHub[13].

[10]https://docs.python.org/2/library/tkinter.html
[11]https://pypi.org/project/SpeechRecognition/
[12]https://gist.github.com/sloria/5693955
[13]https://github.com/scmadmad/SpeakOutAndGetYourBird#methodology

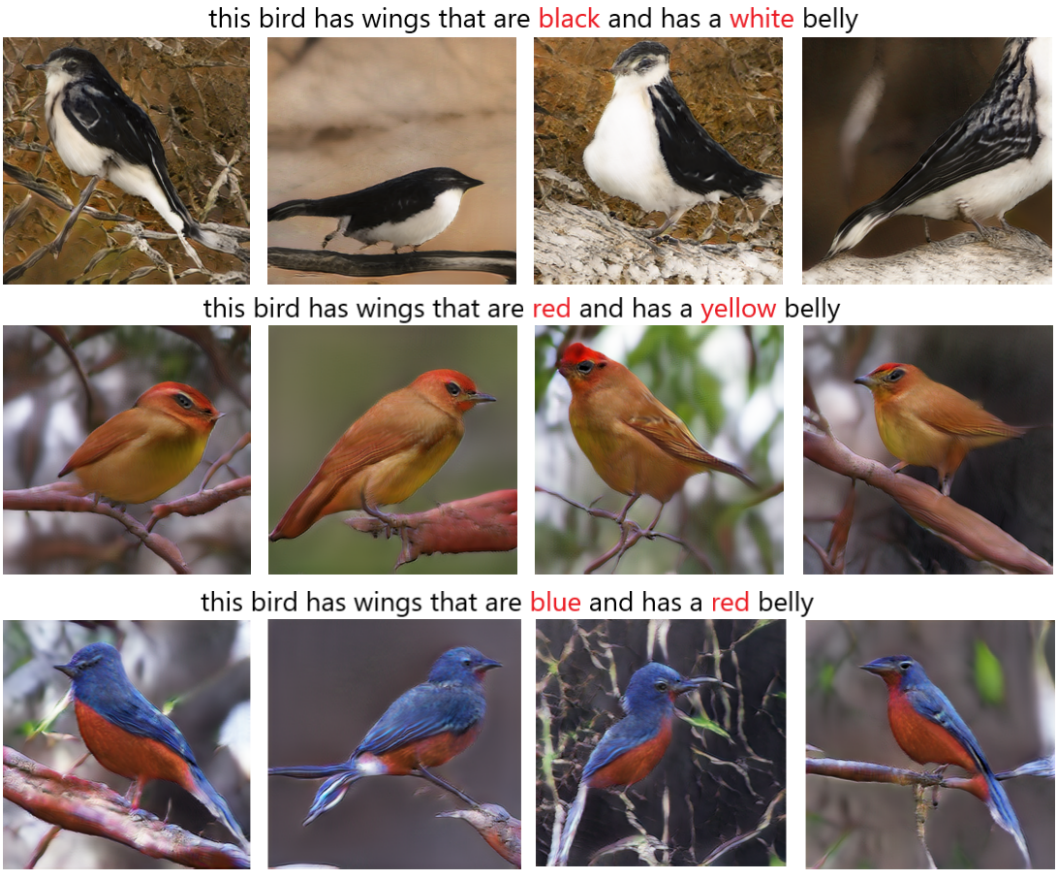

Figure 4: Example images of our model trained on CUB dataset, which shows the generalization ability of our model

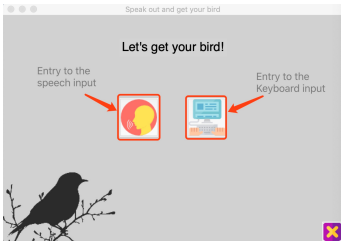

Figure 5: Main panel

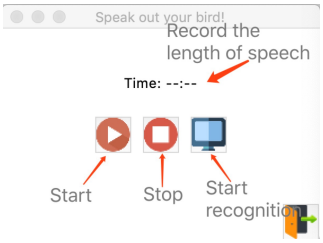

Figure 6: Sub interface for speech input

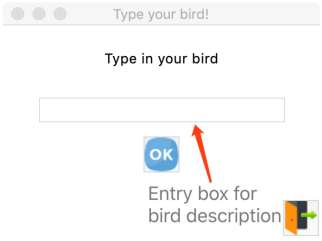

Figure 7: Sub interface for keyboard input

5 Conclusion

5.1 Contribution

All the members in our group had different emphasis on different aspects of work in our project, in order to work more efficiently and effectively. In the initial research stage, A. Ma and R. Wang researched DCGAN[2], Y. Li and Y. Xiao researched StackGAN[3,4], C. Yan researched AttnGAN[1]. In this stage our primary work was to read related paper and discuss about the feasibility of implementation our idea.

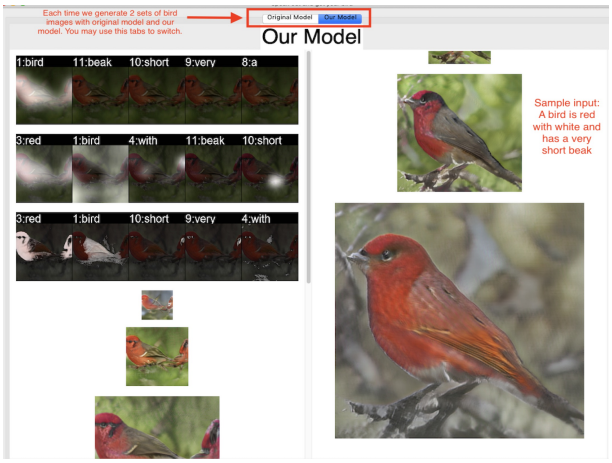

Figure 8: Image generating process interface by our application

After deciding on basis structure of AttnGAN, Li, Yan and Xiao worked on rebuilding and improving AttnGAN, while Ma and Wang worked on implementation of the application. We did not have a model with inception score as high as the original one with baseline setting, so we trained the model on different hyperparameters concurrently on five Google Cloud Platform accounts to see whether better score can be achieved. Yan and Li worked on testing the model. Poster was drafted by Xiao. Ma and Wang worked on improving the application and putting it to GitHub. Report was primarily drafted by Xiao and Yan, with other members' supplements.

5.2 Result

In our project, a four-stage Attentional Generative Adversarial Network is built for generating higher resolution and quality images with size 512x512. Our refined network is based upon the previous AttnGAN by appending one more stage. Different hyperparameters are tuned for the purpose of reaching better outcomes. Specifically, up till now, our model trained with generator learning rate at 0.00015 reaches highest inception score 4.34 ± 0.24 within 200 epochs on CUB dataset. This number is only 0.02 less than that of the original AttnGAN model and as our training process is still ongoing, we are expecting the final outcome to be higher. From the perspective of visual sense, we are able to generate images of size 512x512 with more details by a more precise attentional model. The generation ability is also tested in our experiments.

References

[1] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang and X. He. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. *arXiv:1711.10485*, 2017.

[2] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text to Image Synthesis. *arXiv:1605.05396v2*, 2016.

[3] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Wang and D. Metaxas. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. *arXiv:1612.03242v2*, 2017.

[4] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D. Metaxas. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *arXiv:1710.10916v3*, 2017.

[5] S. Reed, Z. Akata, B. Schiele, H. Lee. Learning Deep Representations of Fine-grained Visual Descriptions. *arXiv:1605.05395*, 2016.

[6] Ian J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661*, 2014.