Faculty of Informatics and Computer Science

Artificial Intelligence

Image Synthesis from Text using Generative Adversarial Network

By: Ahmed Tammaa

Supervised by: Associate Professor. Nahla Barakat

June 2022

Contents

1 Abstract 5
2 Introduction 6
2.1 Overview 6
2.2 Problem Statement 6
2.3 Scope and Objectives 7
2.4 Work Methodology 7
3 Preliminaries of Adversarial Text-To-Image 8
3.1 Background 8
3.2 Pure GAN Architecture 8
3.3 Conditional GAN (cGAN) 9
3.4 Text Encoding 10
4 Related Work (State-of-Art) 11
4.1 First Adversarial Text-To-Image Work 11
4.2 Stacked Generative Adversarial Network 12
4.3 Attention Architectures 15
4.4 Brief Cover of Interactive Text-to-image Approach 20
4.5 The Evaluation Challenge 21
5 Gannt Chart 24
6 Design 25
6.1 GAN-INT-CLS 25
6.1.1 Generator 25
6.1.2 Discriminator 26
6.1.3 Drawbacks 26
6.2 Attention Gan (AttnGAN) 28
6.2.1 DAMSM 28
6.2.2 Generator 29
6.2.3 Discriminator 29
6.2.4 Drawbacks 30
7 Implementation 32
7.1 Dockerization 32
7.2 Code Organization 32

7.3 Tensorflow 32
7.3.1 Preprocessing 32
7.3.2 Usage of BERT and Inception 33
7.3.3 Speeding up execution time 33
7.4 Gensim 33
7.5 PyTorch 33
7.6 Tensorflow and Pytorch Mix 33
8 Trials, Results, and discussion 34
8.1 Initial Thoughts, Results and Generic Problem Discussion 34
8.1.1 Discussion: Are the Available Datasets good? 34
8.1.2 Result: GAN-INT-CLS 34
8.2 Instance Selection Thinking (Think of Generation as Classification) 37
8.2.1 Discussion: Generative Datasets and Model Size 37
8.2.2 Trial: Instance Selection on INT-CLS-GAN 38
8.2.3 Trial: Increase Model Complexity (Instance Selection Power) 42
8.3 BERT (Text-Classification for Better Generator Latent Space) 44
8.3.1 Discussion: Bigger Embedding vector and using BERT 44
8.3.2 Trial: BERT Pooled Output as Embedding For GAN-INT-CLS With Encoding 45
8.3.3 Trial: BERT Pooled Output as Embedding For GAN-INT-CLS Without Encoding 47
8.4 Create Customized Embeddings from Scratch 50
8.4.1 Discussion: Manually Train a Word-To-Vec 50
8.4.2 Trial: Continuous Back of Words Manually trained 51
8.4.3 Trial: Skip Gram Manually Trained 53
8.5 Divide the Problem into Stages and Use Attention (Reverse Captioning Problem) 56
8.5.1 Discussion: Why Stage Generators Are Better and What the Drawbacks? 56
8.5.2 Discussion: How the Attention Models Improve the Task and How Is It Related to Image Captioning? 57
8.5.3 Trial: Using Attention GAN for improving the image quality 58
8.6 Connecting Results, Comparative Analysis and Future Work 63
8.6.1 Results: Final Best Results 63
9 Future Work 66
9 Conclusions & Recommendations 68
9.2 Project Contributions 68

Reference 70
Appendix I 73

List of Figures

Figure 1 shows pure GAN architecture 9
Figure 2 shows pure cGAN architecture 9
Figure 3 Shows pure ACGAN with extra Auxiliary loss 10
Figure 4 Text encoder simple structure 10
Figure 5 Shows comparison between GAN, GAN-INT- and GAN-INT-CLS [7] 12
Figure 6 shows a comparison between GAN-INT-CLS, Stack GAN and Stack GAN++ [11] 13
Figure 7 stackGAN Architecture 13
Figure 8 stackGAN++ Architecture 14
Figure 9 architecture of the HD-GAN [12] 15
Figure 10 shows a comparison between Stack GAN and HD-GAN [12] 15
Figure 11 shows attnGAN architecture [17] 16
Figure 12 shows the importance of the channel-wise attention to the model (ControlGAN) by showing the same model with it and without it. [19] 17
Figure 13 shows the importance of the channel-wise attention to the model (ControlGAN) by showing the same model with it and without it. [19] 18
Figure 14 showing a comparison between StackGAN++, AttnGAN and Control GAN [19] 18
Figure 15 shows the architecture of the MirrorGAN [24] 19
Figure 16 showing a comparison between mirrorGAN and its older counterpart attnGAN 20
Figure 17 example of dialogue-based text-to-image 21
Figure 18 Gantt Chart 24
Figure 19 shows how the generator image moves in latent space 26
Figure 20 results of GAN-INT-CLS 28
Figure 21 Output image from the AttnGAN issues in the structure of the bird 30
Figure 22 Output image from the AttnGAN 31
Figure 23 shows the attention plot for the image generation 31
Figure 24 This bird has red head and blue body 35
Figure 25 This bird has white head and blue body 36
Figure 26 this bird is yellow 37
Figure 27 This bird has red head and blue body 39
Figure 28 This bird has white head and blue body 40
Figure 29 This bird is yellow 41
Figure 30 This bird has red head and blue body 42
Figure 31 This bird has white head and blue body 43
Figure 32 This bird is yellow 44
Figure 33 This bird has red head and blue body 45
Figure 34 This bird has white head and blue body 46
Figure 35 This bird is yellow 47
Figure 36 This bird has red head and blue body 48
Figure 37 This bird has white head and blue body 49

Figure 38 This bird is yellow 50
Figure 39 This bird has red head and blue body 51
Figure 40 This bird has white head and blue body 52
Figure 41 This bird is yellow 53
Figure 42 This bird has red head and blue body 54
Figure 43 This bird has white head and blue body 55
Figure 44 This bird is yellow 56
Figure 45 Example of AttnGAN output 58
Figure 46 This bird has red head and blue body 64× 64 58
Figure 47 This bird has red head and blue body 128×128 59
Figure 48 This bird has red head and blue body 256×256 59
Figure 49 This bird has white head and blue body64 × 64 60
Figure 50 This bird has white head and blue body 128×128 60
Figure 51 This bird has white head and blue body 256×256 61
Figure 52 This bird is yellow 64 × 64 61
Figure 53 This bird is yellow128 × 128 62
Figure 54 This bird is yellow 256 × 256 62
Figure 55 Evolution of bad samples 63
Figure 56 Evolution of Good Samples 63
Figure 57 Visualized Comparison of Inception scores with literature 64
Figure 58 FID scores Comparison of our different trials 65
Figure 59 Comparison between Imagen and DALL-E on writting text on the image 67

List of Tables

Table 1 compare different metrics [34] 23
Table 2 shows the output of the discriminator 26
Table 3 shows quantitative comparison for different trails and the official paper results 63

1 Abstract

The Text-to-Image generation is an open research field aims to generate a photo-realistic image from a text description. The text input can vary from a single statement, the generic approach, and/or an interactive dialogue which makes the human and computer interact to modify and generate the image. The current results are far from perfect since the research has started in 2016 and the task is complex with various challenges, such as, a lack of a good metric. Furthermore, there is no discovered metric can give a single score to measure it. However, the current metrics can give higher scores to a synthetic image more than the ground truth image even if the synthetic image quality is remarkably low. Moreover, in the early stages of the research the generator was sacrificing the relevance and the quality to deceive the discriminator. Fortunately, the steps of the research solved this problem by enforcing the generator to generate a text-relevant image. The semantic and colours of the image are improved by introducing the attention-based GAN architectures. The project aimed on trying various variants of the GAN-INT-CLS for its small and maintainable size relative to other methods such as stackGAN and AttnGAN. We showed that using bigger embedding vector can improve the image quality in the expenses of the diversity. Moreover, this project shows experiment of instance selection which made the model trains faster, however, the performance has degraded which resulted in a trade-off between speed and quality. We have achieved best FID of 258 and Inception score of 2.43 with few numbers of epochs compared to the original. We have showed that better implementation can raise the performance by 54%. Thus, the implementation of the new ideas should be carefully written with respect to the performance.

2 Introduction

Humans are naturally visual; furthermore, listening to phrases, stories, or reading a text immediately an image is visualized in our head. This ability is proven to be key factor on humans' cognitive abilities [1]. On contrary, the computers are fast calculators. However, with the modern advances in artificial intelligence specifically the Generative Adversarial Network (GAN), the computers can form a form of visualization from a given text. This project aims to generate photo-realistic images that are consistent with a given text input. This process is the reverse of image captioning, moreover, the image captioning describes a given image while this project generates an image from text. This project is helpful in image editing and contributes to the early stages of creativity and imagination. Furthermore, an artist can describe what is in the mind and get a close image to describe what is in the mind or the early design stages. Besides, it can help designers to make quick modifications and facilitate communication with their customers. Finally, the text-to-image is used in image retrieval application.

2.1 Overview

The image generation is using Generative Adversarial Network (GAN). The GANs architecture is comprised of two deep learning models. The First one is the discriminator which is implicitly a classifier to differentiate between the real and fake image. The second is the generator which tries to generate a realistic output to deceive the discriminator. Thus, the images generated must be as realistic as possible. However, this architecture generally has some undesired features, such as, there is no agreed metric for evaluation of GANs. In 2019, Barua et al [2] proposed cross local intrinsic dimensionality (CrossLID) as metric that is based on the manifold degree of coincidence between two data distributions. The CrossLID is proposed as sensitive to mode collapse image transformation, and robust to small-scale noise. However, this new evaluation metric is critiqued for scalability problems and clarity of the paper is questioned by the researchers. Hence, this measure still under research and it is not certain if it should be used or not. From other challenges of GANs is it takes very long time to train and at some case the GANs do not converges.

2.2 Problem Statement

Given a textual description for a scene generate a photo-realistic image that aligns with the given description with diversity of the output.

2.3 Scope and Objectives

Improve the quality of the generated image by improving preprocessing approach and/or by improving a network architecture.

2.4 Work Methodology

This paper attempts to generate a higher quality image by modifying the GANs architecture and/or improving the pre-processing.

3 Preliminaries of Adversarial Text-To-Image

3.1 Background

A lot of research has been conducted on the text-to-image synthesis using GANs. As its motivation is to synthesis a realistic input that can deceive a classifier to predict it as real. At some cases, the GAN networks can deceive humans as it has various applications like image super resolution, human face synthesis and many other applications. On the text-to-image context, the first research conducted on text-to-image was by Reed et al [3] by developing a deep convolutional GAN (DC-GAN) where the encoded text is the conditional factor. Furthermore, both deep learning models, generator, and discriminator, are trained feed-forward with inference conditioned on the given text description [3].

3.2 Pure GAN Architecture

The GAN on the pure form is a deep learning model consisting of two neural networks which are the Generator and Discriminator. Where the discriminator is a binary classifier to differentiate the real vs fake inputs, while the generator tries to generate image that deceive the classifier by generating a realistic input. Formally, both neural networks can be represented by a zero-sum game. Moreover, each neural network tries to maximize its chance of winning the game and minimize their counterpart [4]. More formally, this architecture is represented mathematically by this equation.

$$\min_G \max_D V(D,G) = E_{x p_{\text{data}}(x)}\left[\log(D(x)\right] + E_{z p_{z(z)}}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

The ‘D’ and ‘G’ Represent the Discriminator and Generator respectively. ‘E’ represents the mathematical expectation of the real data distribution $p_{\text{data}}(x)$ of the variation of binary cross entropy of ground truth 1,0 (real - fake) $p_z(z)$ are the noise variable. G (Z; θ_g) is denoting the differential function of the generator that is represented by Multi-Layer Perceptron (MLP) D(X) is the discrimination function that denotes the probability of X were from the original dataset not from the generated dataset. Moreover, the generator tries to minimize $log(1 - D(G(z))$ and the discriminator tries maximizing it. Hence, the mathematical representation shows the idea of GAN

clearly, moreover, other GAN architectures will be discussed is using a variation of this basic idea. Figure 1 shows the architecture of the pure generative adversarial network.

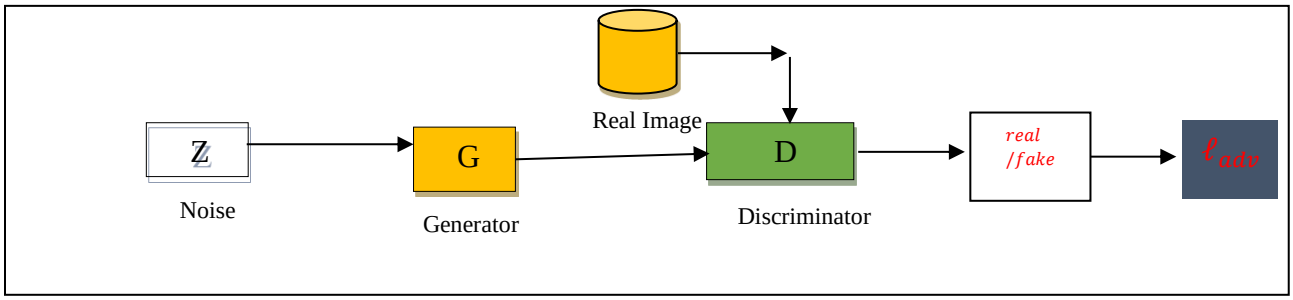

Figure 1 shows pure GAN architecture

3.3 Conditional GAN (cGAN)

The GAN generation of image is very powerful approach, however, adding a control of the generated image is a great enhancer for the quality of the image. Mirza et al [5] introduced the conditional GAN (cGAN) which added new variable (y) that is class labels to have the generated image be conditioned on it. Formally, it can be represented in this formula.

$$\min_G \max_D V(D,G) = E_{xp_{\text{data}}(x)}\left[\log(D(x|y)\right] + E_{zp_{z(z)}}\left[\log\left(1 - D\left(G(z|y)\right)\right)\right] \quad (2)$$

This Architecture is later extended with Auxiliary Classifier that is changes the discriminator to predict the class label of the classifier to check it instead of being given as an input [6]. This improvement has improved the stability and enhanced the quality of the generated image. This idea will enhance the relation between the generated image to the text description. Figure 2 shows cGAN and Figure 3 shows ACGAN.

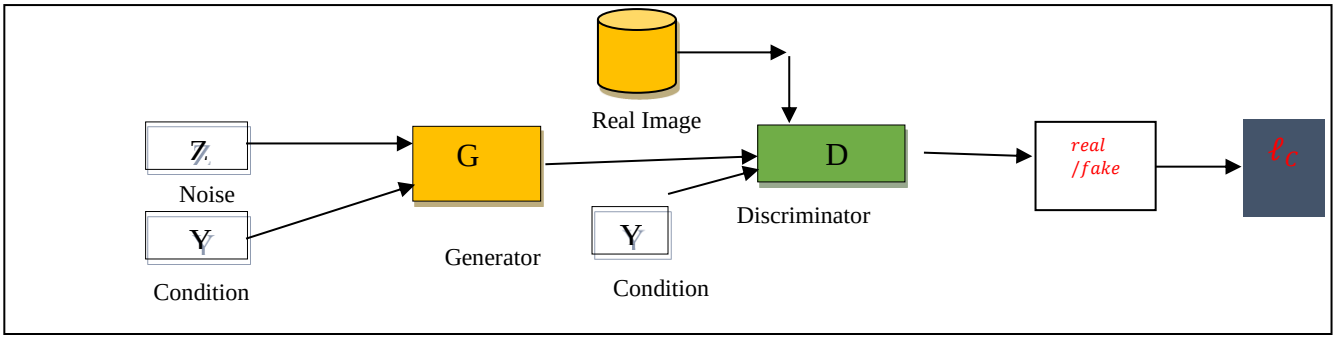

Figure 2 shows pure cGAN architecture

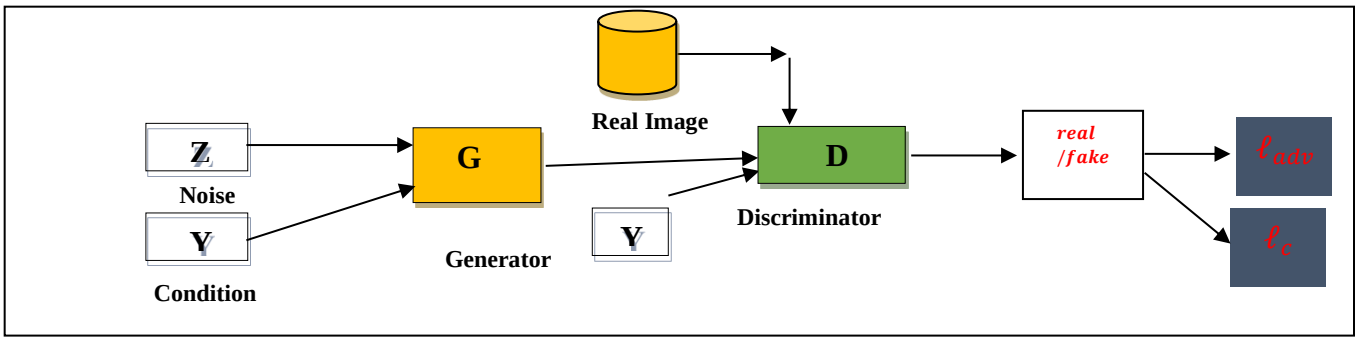

Figure 3 Shows pure ACGAN with extra Auxiliary loss

3.4 Text Encoding

The first step in the text encoding of the text to image was proposed by Reed et al [3] on the first research by using character-level Convolutional Recurrent Neural Network (Char-CNN-RNN). Furthermore, it is trained neural network that can learn the relation between a text and image by a given set of labels. Moreover, the Reed et al [3] experimented other text encoding techniques such as Bag-of-Words and Word2Vec their experiment proved that Char-CNN-RNN was more effective in the text-to-image context. Zhang et al [7] proposed a better modeling way conditioning augmentation uses the covariance matrix and text embedding as functions for text embedding and samples latent variable from Gaussian distribution. This approach is regularized by the Kullback-Leibler Divergence (KL-Divergence) term among the training. This technique became used frequently for the text to image translation. Moreover, another approach was introduced Souza [8] by proposing the Sentence Interpolation similar to the conditioning augmentation it smoothen the text embedding throughout the training phase. The advances continue by proposing the bi-directional Long Short-Term Memory (BiLSTM) for forming feature matrix for each word using hidden states [8]. Recent research [9,10] is starting to use pre-trained transformers such as BERT to perform the text embedding task [11]. Figure 4 shows the basic architecture.

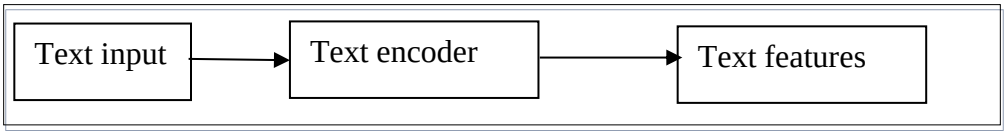

Figure 4 Text encoder simple structure

4 Related Work (State-of-Art)

This section will review the related work and state-of-art of the text-to-image with different neural network architecture

4.1 First Adversarial Text-To-Image Work

As mentioned in the background section the first text-to-image translation was introduced by Reed et al [3]. The architecture proposed in Reed et al was a variation of cGAN by replacing the class label y by the text embedding φ. Their training approach forces both components of GAN to focus on text-alignment to the given textual description; Furthermore, the input to the discriminator is real image with mismatched text and generated image with a mismatched text. This approach is named "Matching-aware discriminator" GAN-CLS. The research added another architecture that learns based on manifold interpolation which is named GAN-INT. Furthermore, the idea is motivated by the property of deep learning models to learn from interpolation between embedding when it nears to the data manifold. The interpolation does not have to correspond to the input text; thus, it does not add labelling cost. Formally, it is an additional generator objective to minimize

$$E_{t_1,t_2 p_{data}}[\log(1 - D(G(z, \beta \mathrm{t}_1 + (1-\beta)\mathrm{t}_2)))] \quad (3)$$

Where z is noise distribution and β interpolate the $t_1$and $t_2$embedding. Experiments showed that $\beta = 0.5$ showed good results [3]. Lastly, the final architecture is combination between GAN-INT and GAN-CLS to result into GAN-INT-CLS. Figure 5 shows a comparison of the three architectures.

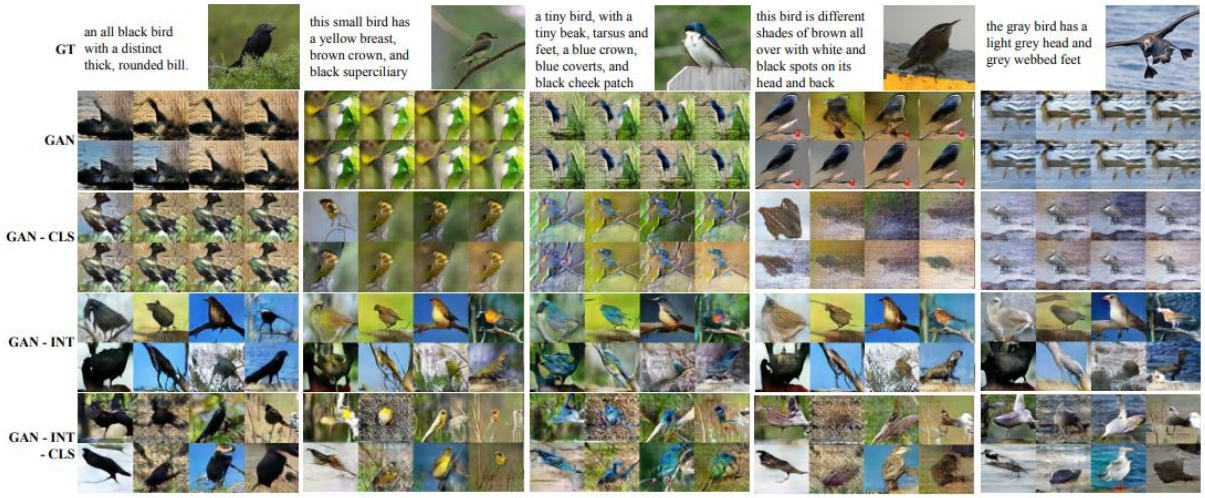

Figure 5 Shows comparison between GAN, GAN-INT- and GAN-INT-CLS [7]

Qualitatively, the results of GAN-INT-CLS outperform its counterparts. Notice that all generated images are 64x64.

4.2 Stacked Generative Adversarial Network

The stacked GAN approaches came to solve the problem of building higher resolution images that the GAN-INT-CLS which is limited to 64 by 64. Its mechanism works by first generate an initial 64x64 image from the given input text and random noise. The generated image is inserted into another generator with text embedding to generate 256 x 256. In the two-staged iterations the is two discriminators that check matching to non-matching text [7]. Zhang et al [11] proposed stackGAN++ that improved their previous architecture by training the three generator and the three discriminators jointly trained to simultaneously figure out the image distribution for multi-scale and conditional training. Besides, they introduced the dynamic text embedding to smoothen the conditional manifold from the gaussian distribution replacing the fixed embeddings. Figure 6 shows comparison between GAN-INT-CLS, StackedGAN and StackedGAN++.

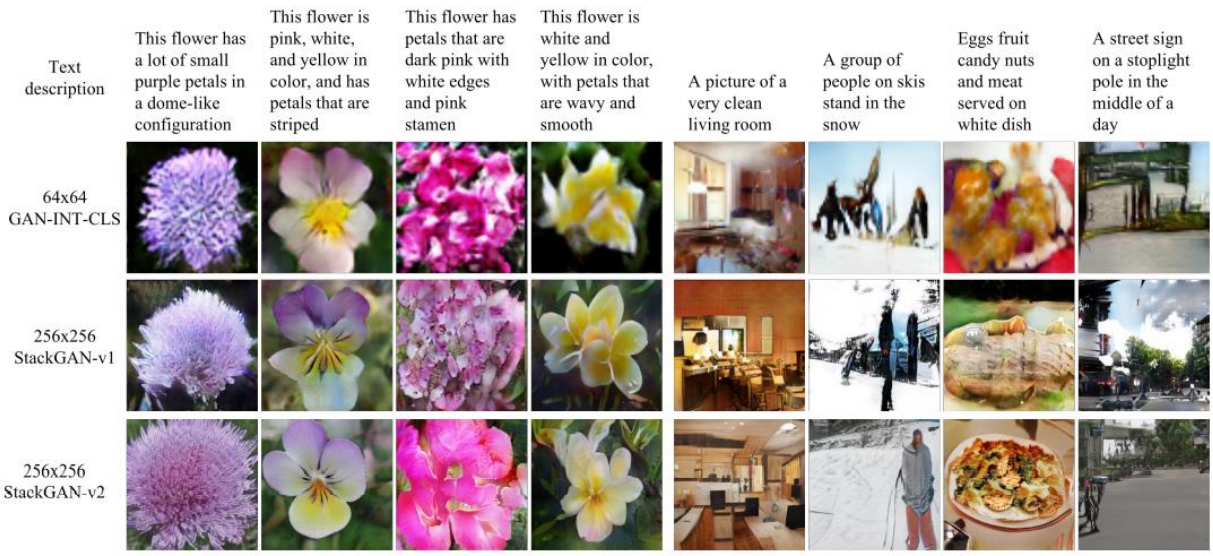

Figure 6 shows a comparison between GAN-INT-CLS, Stack GAN and Stack GAN++ [11]

The stackGAN has better color consistency and shows greater details than the other two previous architectures. Figure 7 shows the architecture of stackGAN and Figure 8 shows StackGAN++

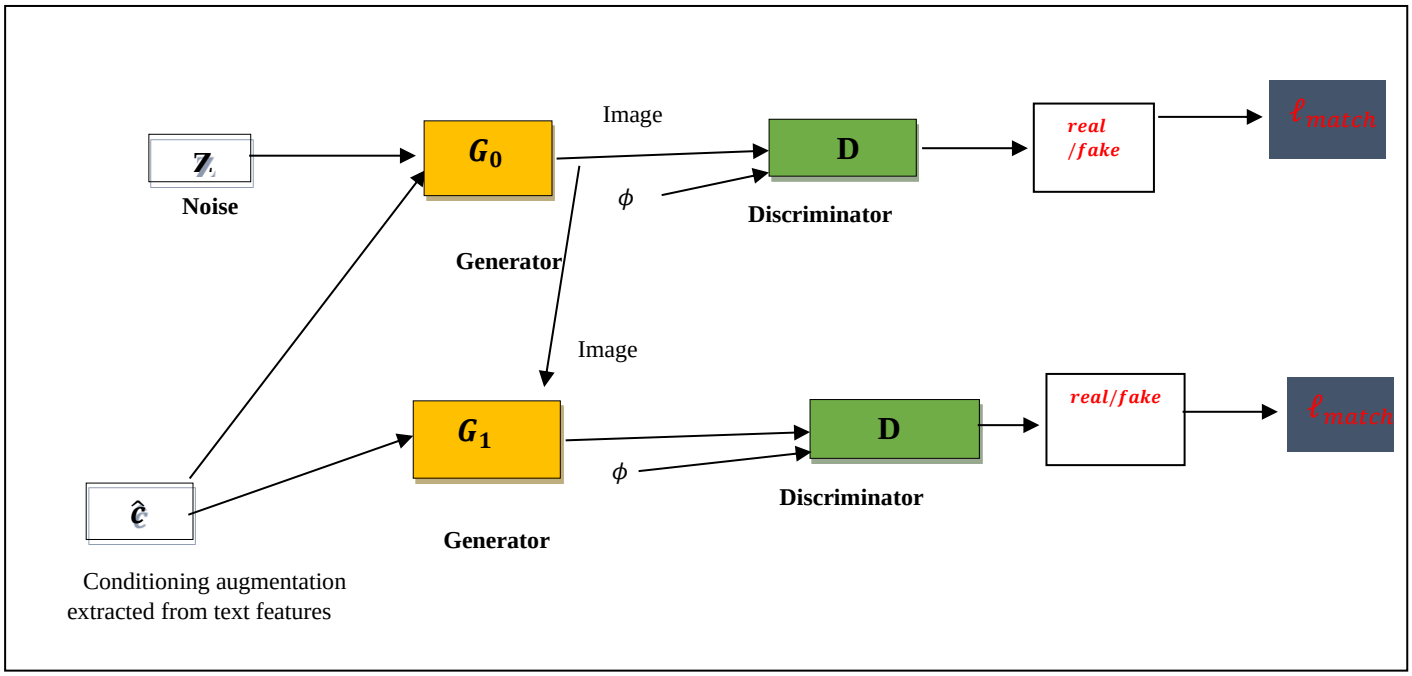

Figure 7 stackGAN Architecture

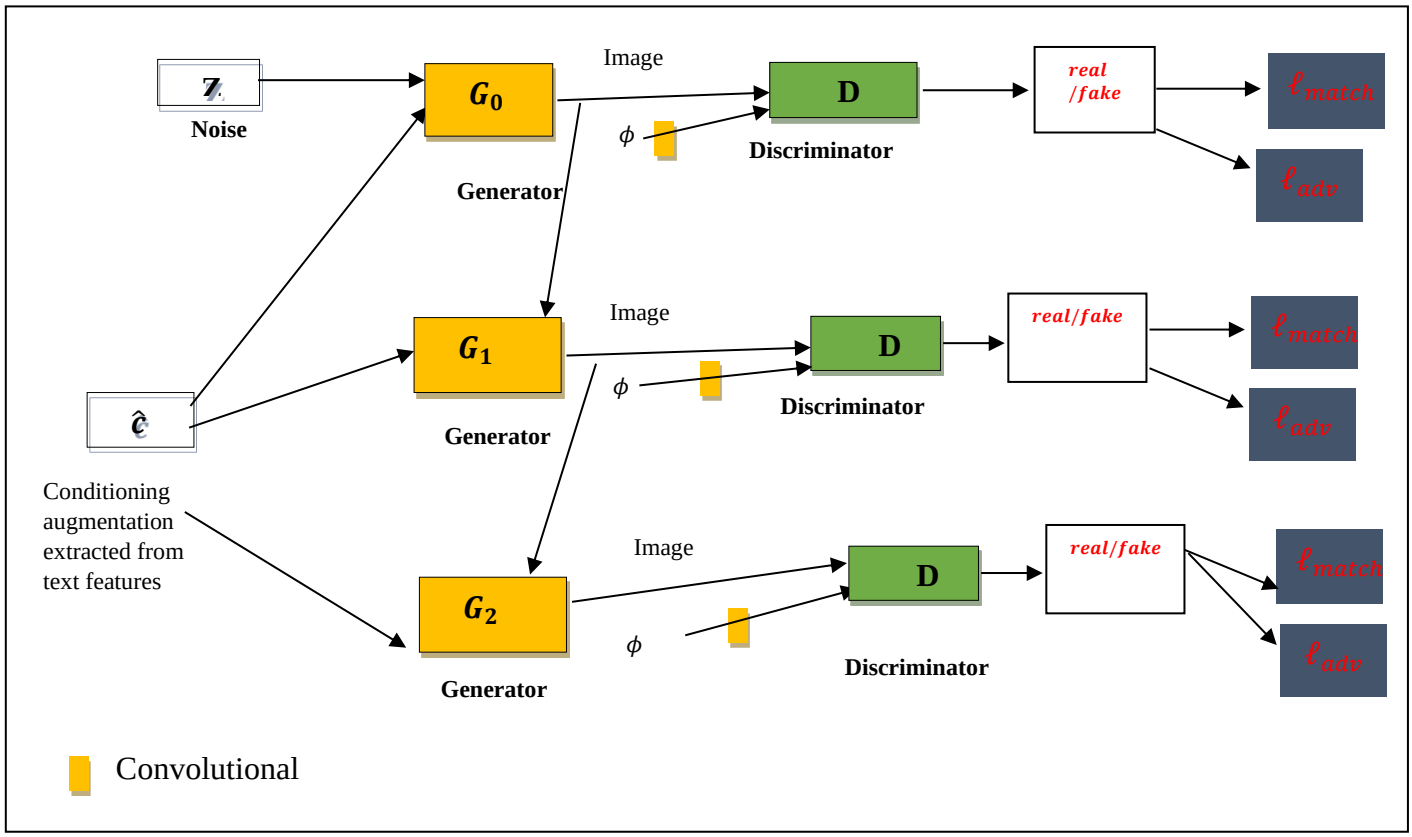

Figure 8 stackGAN++ Architecture

When the layers jointly trained for simultaneous detecting the image distribution, it enhances the quality of the generated image by regularizing the colour consistency. There is challenge introduced in this architecture which is multi-level generators. This challenge led to a new stack-based architecture which works with one generator and three discriminators. The new architecture uses hierarchical-nested network which has adversarial objectives [12]. Besides, having multi-level purpose as it has three discriminators. Figure 9 shows the architecture of the HD-GAN. Figure 10 shows a comparison between the stackGAN and the HD-GAN. From the comparison it can be shown that HD-GAN is finer detailed and has a significantly better scaling.

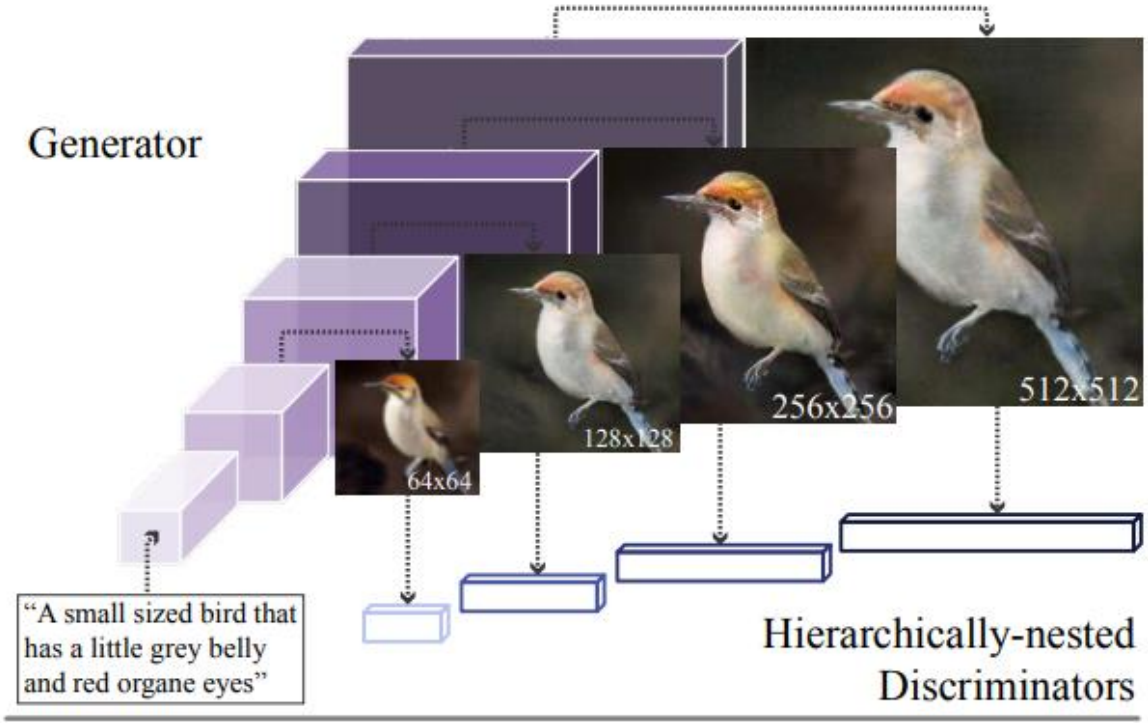

Figure 9 architecture of the HD-GAN [12]

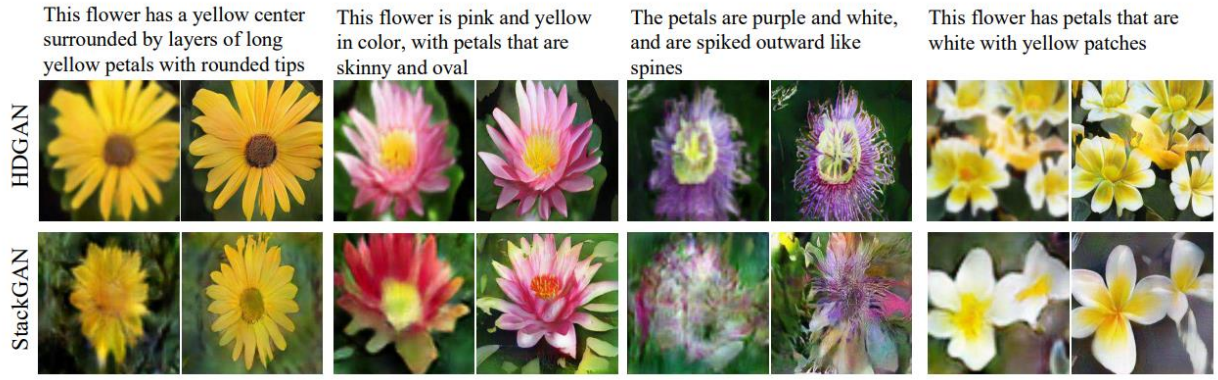

Figure 10 shows a comparison between Stack GAN and HD-GAN [12]

4.3 Attention Architectures

In deep learning context, the attention refers to the focus of specific part of the input by weighting its input correspondingly to its importance. The attention mechanisms have proven to be very effective in applications of visual computing and language processing [13,14,15,16].

Which highly relevant to text-to-speech. The attention mechanism in text-to-image architecture is based on Stack-GAN++ [15] they build-in the attention mechanism upon the pipelines of the layers of Stack-GAN++. The AttnGAN achieve that by Deep Attentional Multimodal similarity model (DAMSM) that targets to compute the similarity index between on word-level [17]. Figure 11 shows the architecture of attention GAN.

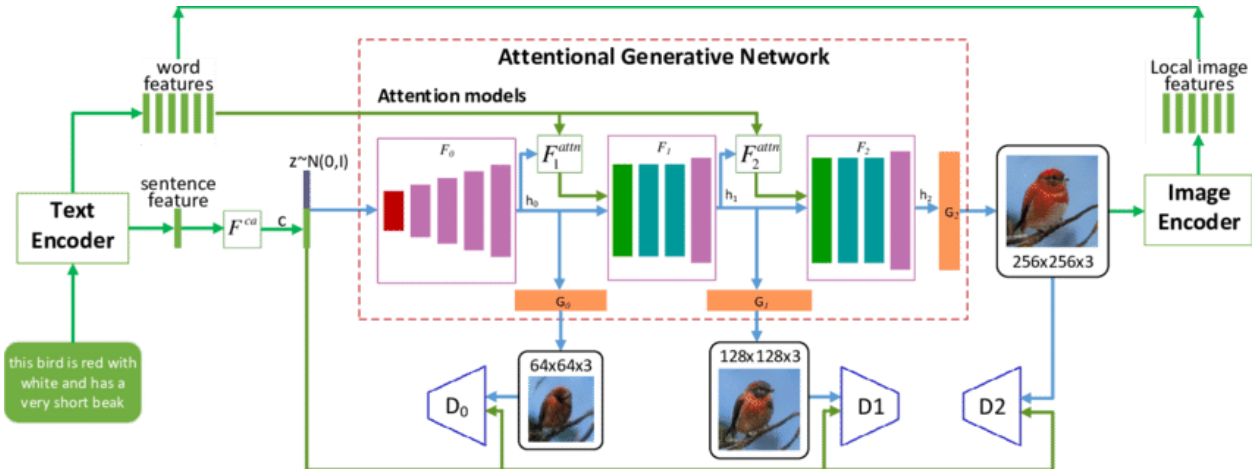

Figure 11 shows attnGAN architecture [17]

Tan et al [18] proposed a new approach SEGAN in text-to-image. Refining the idea of AttnGAN by exclusively focusing on visually heavy keywords instead of assigning a weight for each word in the textual description. To know the keywords only the authors used attention regularization term.

Lastly, in this section, Li et al introduced the controlGAN architecture which is capable of achieving the text-to-image synthesis and editing the textures and visual attributes without affecting the background of the image and other content [19]. This is achieved by creating a word-level spatial; besides, it is supplied with channel-wise attention to have a more accurate region-based synthesis from the attention mechanism. Furthermore, a 'bird' is very likely to be the central region also specifying the head, chest, and wing of the bird as subregions. Moreover, this attention approach can differentiate between regions and their colors which adds up to a better-quality image. Moreover, spatial domain attention (Word-level) focuses mainly on colours while channel-wise attention focuses on the semantic parts of the images. To visually illustrate the word-level discriminator, Li et al. have provided the controlGAN model trained without the word-level

discriminator as shown in Figure 12. It can be shown that the model failed to correctly correlate the colors with the regions given, such as the last image that states the head must be white, but it outputs a brown-headed bird. In figure 13, the model has been trained without the channel-wise attention the small changes in the regions of the image such as the belly and the head made the generator create a very different-looking image in terms of background and the pose. While channel-wise attention allows the model to have better control over the image and easily create a semantic change to the image while keeping other details correctly aligned to the text. Figure 14 shows an output comparison between StackGAN++, AttnGAN, and ControlGAN.

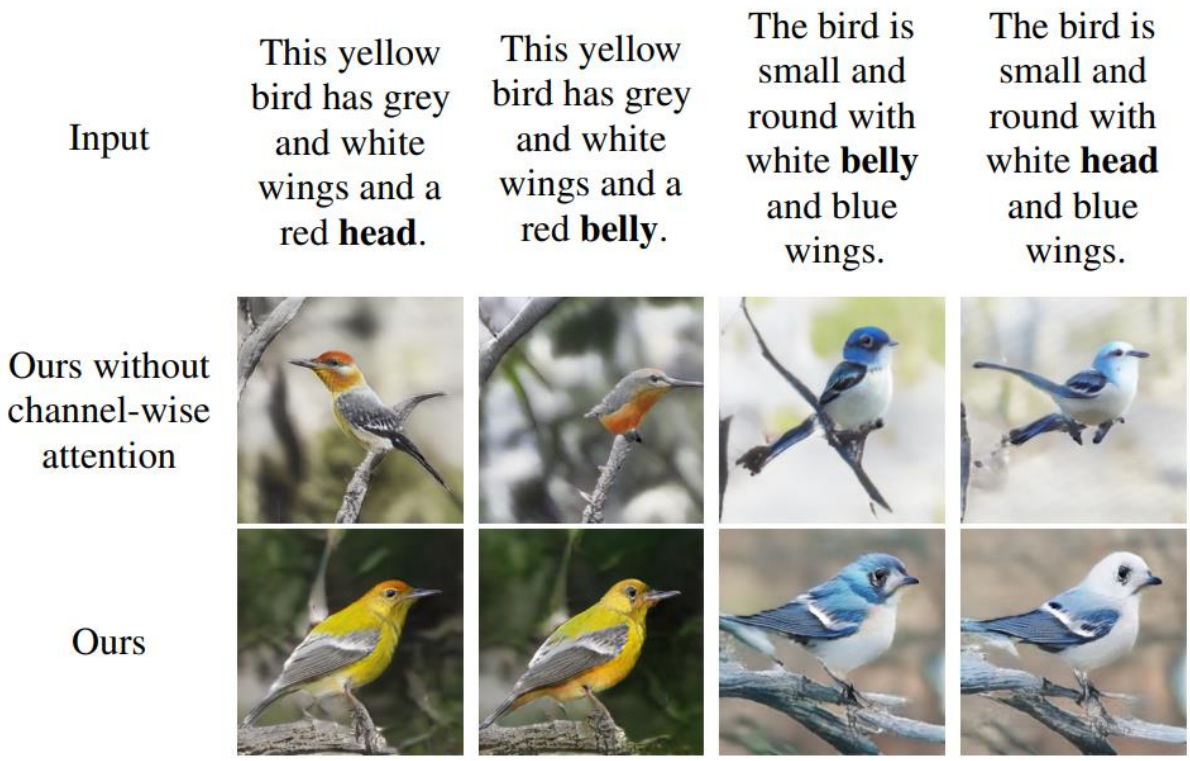

Figure 12 shows the importance of the channel-wise attention to the model (ControlGAN) by showing the same model with it and without it. [19]

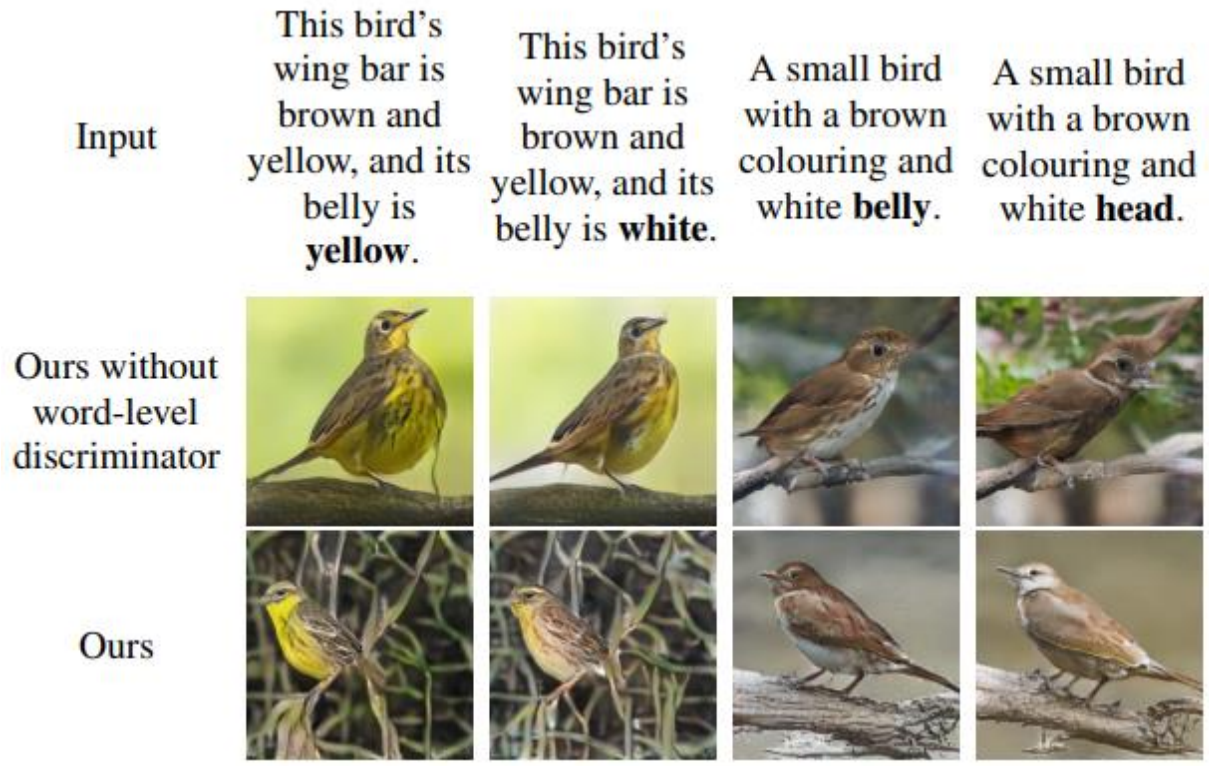

Figure 13 shows the importance of the channel-wise attention to the model (ControlGAN) by showing the same model with it and without it. [19]

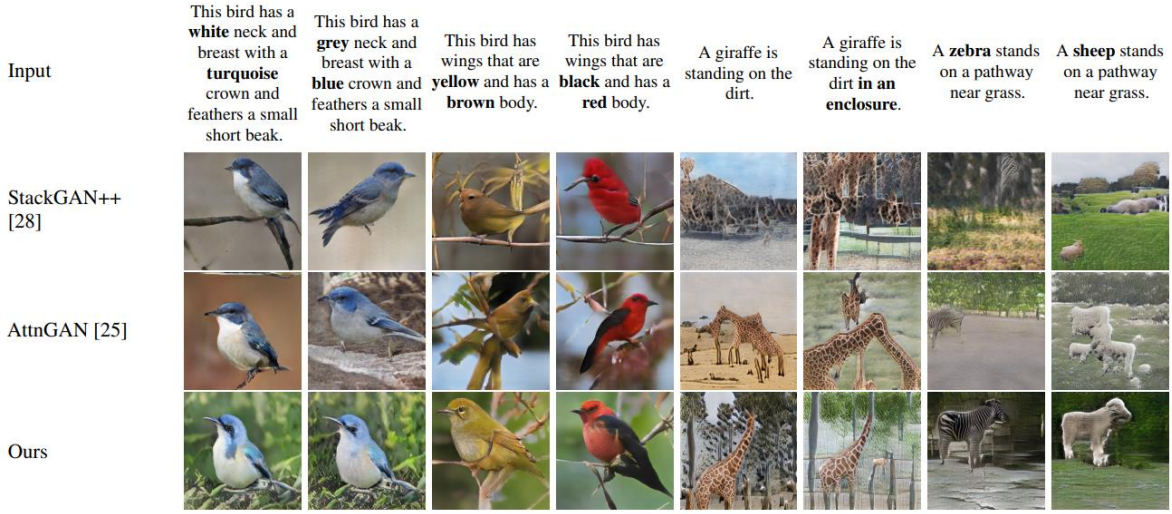

Figure 14 showing a comparison between StackGAN++, AttnGAN and Control GAN [19]

4.3 Cycle Consistency Architecture

In the context of unsupervised learning the cycle consistency is the process which allows the model to reconstruct an image x from the latent variable z. The cycle consistency has been integrated with various GAN architectures such as cycleGAN [21], DiscoGAN [22] and DualGAN [23]. Qiao et al [24] proposed a new architecture that takes the advantage of encoders and decoders networks that are used for image captioning to check the loss entropy of the text reconstruction of the image to see how much the synthetic image correlate with the given text [25,26] to guide the generators from the given text embeddings.

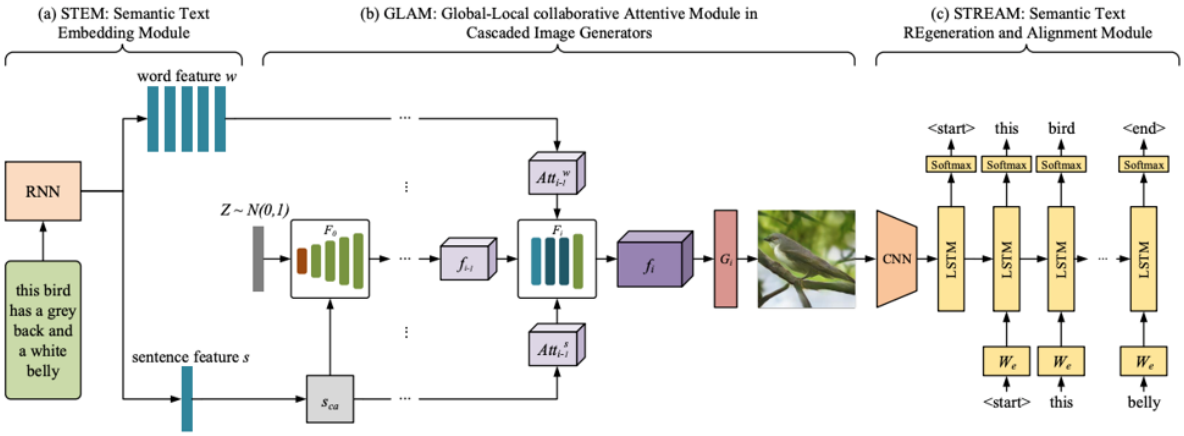

Figure 15 shows the architecture of the MirrorGAN [24]

Figure 16 shows a comparison for MirrorGAN and AttnGAN. Qualitiatively, the MirrorGAN outperforms the older AttnGAN.

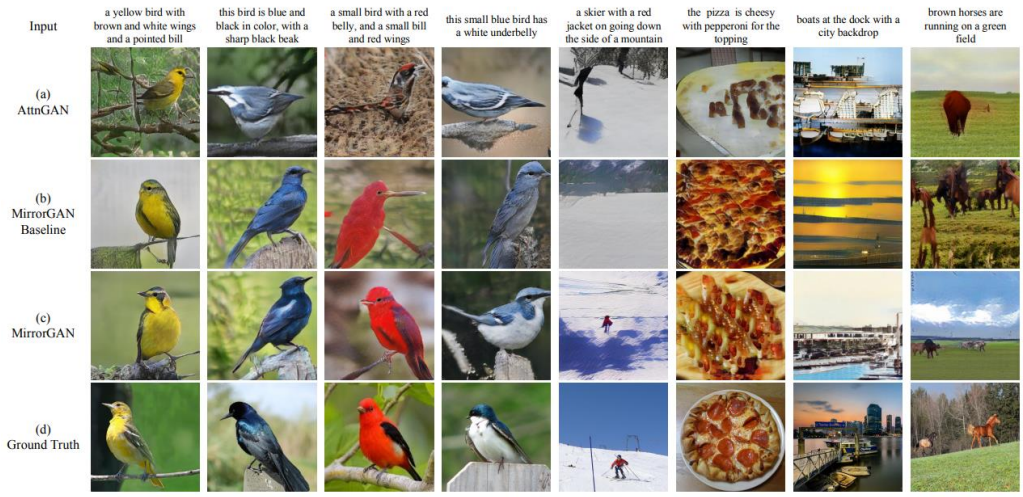

Figure 16 showing a comparison between mirrorGAN and its older counterpart attnGAN

Lao et al [27] proposed a new approach which infer two latent variables (style and content) from a real image by the cycle consistency approach. These latent variables are correspondingly inserted into the generator to synthesis image too close to the inferred variables.

4.4 Brief Cover of Interactive Text-to-image Approach

The approaches and architecture discussed was mainly unsupervised approaches and relies on only one given caption or description. Sequential Attention GAN sqnAttnGAN was proposed by Cheng et al [28] a dialogue-based text-to-image approach which generate image based on user interactivity as shown in Figure 17. The approach is based on interactively gaining information by asking questions. This approach enhances image editing by automating the editing by given a base image.

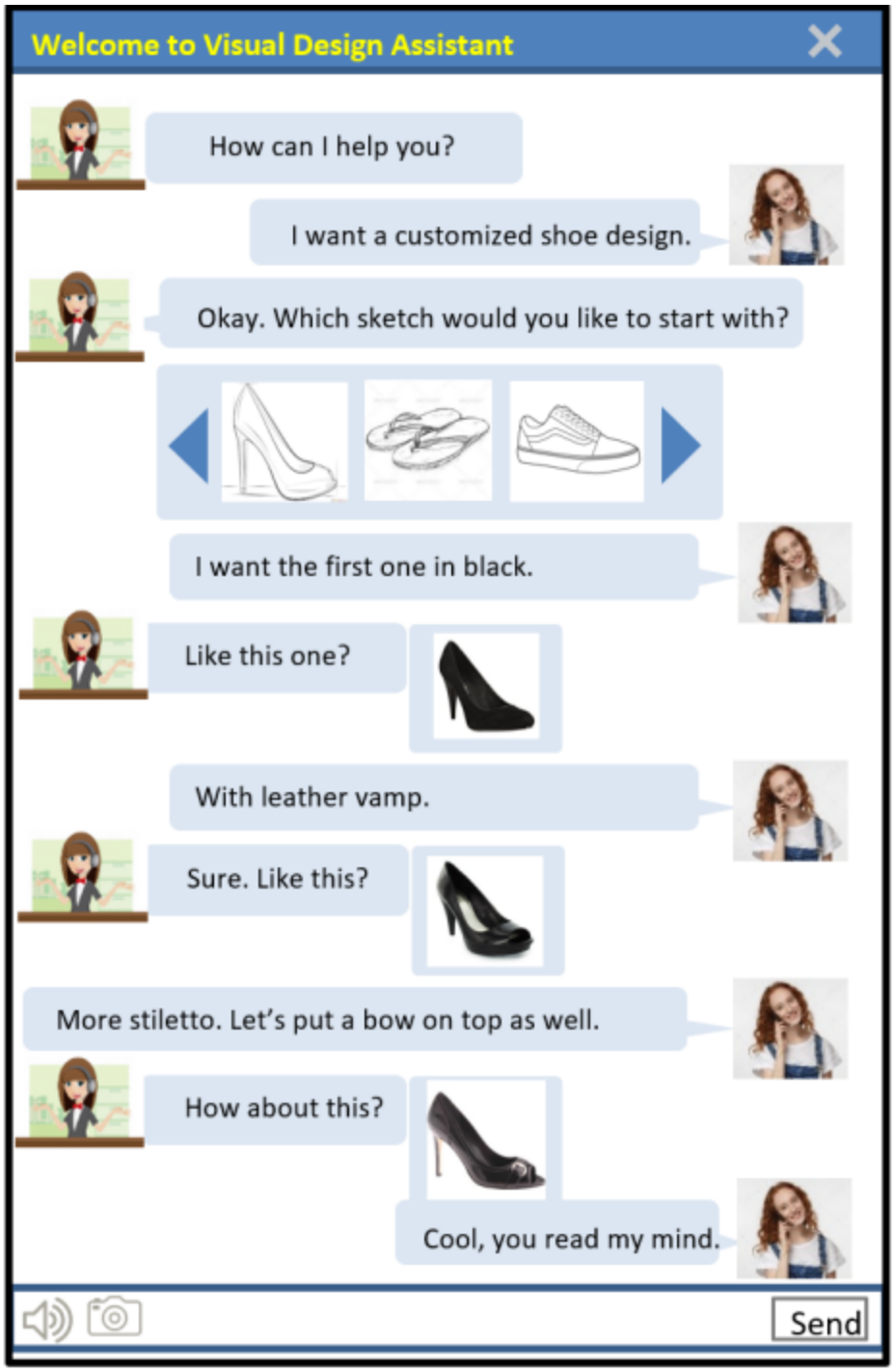

Figure 17 example of dialogue-based text-to-image

4.5 The Evaluation Challenge

The aforementioned approaches enhanced the output of the text-to-image all the way from the first approach down the line to the most recent architectures. However, there is an underlaying

problem of measuring the performance. The current measures, at some cases, give a higher result value for a clearly looking synthetic image over the real image which is very deceiving. Hence, if the Generator training was objectively trying to optimize the scores of the metrics it can greedily optimize the benchmark score with sacrificing the result quality of the image. It can be inferred that the architecture mentioned before tries to constrain the generator to keep the semantic alignment and reflect the image to the text otherwise the generator tries to form any image that can deceive the generator. There is no single metric can evaluate the two aspects of product which are the generated image and the relevance of text. Thus, there are two types of generating the image. The first category are the image evaluators which are the Inception Score (IS) [29] and Fréchet Inception Distance (FID) [30]. The IS takes two factors into consideration the first is distinctively of the image and the diversity of the generated images. The IS takes the conditional probability distribution $p(y|x)$ the meaningfulness of the image is inversely proportional with the entropy of the distribution. Furthermore, a more meaningful image will result in a low entropy. The diversity of the image is measured with integral margin of the probability $\int (p(y|x = G(z))dz$ a more diverse images results in high entropy values. The two objective requirements are measured by the KL-Divergence of $p(y|x)||p(y)$. Formally, the Inception score is defined by

$$IS = \exp(E_x(KL(p|x)\ ||p(y))\ \ (4)$$

The FID metric is more stable and consistent than the inception score metric it is based on the Euclidian distance of the probability distribution of the real image and the synthetic image [31]. However, this metric assumes that the features follow a gaussian distribution. This assumption is not always true; thus, a Kernel Inception Distance (KID) introduced as a better metric and unbiased as in FID [32].

As an example, for text relevance, Hinz et al [33] introduced a semantic Object Accuracy, the idea of this metric is based on image captioning. Furthermore, if the caption was “A man standing on a chair in an office” then there must be a man, a chair, and an office as recognizable objects. This metric detects only the explicit mentions of the objects. The inferred details of images and other objects that was not mentioned in the text are not metered; thus, the meaningfulness of the image is not assessed in this metric. There are other approaches that measures the text relevance such as R-precision such in [17] and image captioning. The different metrics are represented in table 1

Table 1 compare different metrics [34]

Metric	Image Quality	Image Diversity	Text relevance	Explainable
IS	Yes	No	No	No
FID	Yes	Yes	No	No
R-Precision	No	No	Yes	No
SOA	No	No	Yes	No
Captioning	No	No	Yes	No

5 Gannt Chart

Figure 18 shows the working plan for the project.

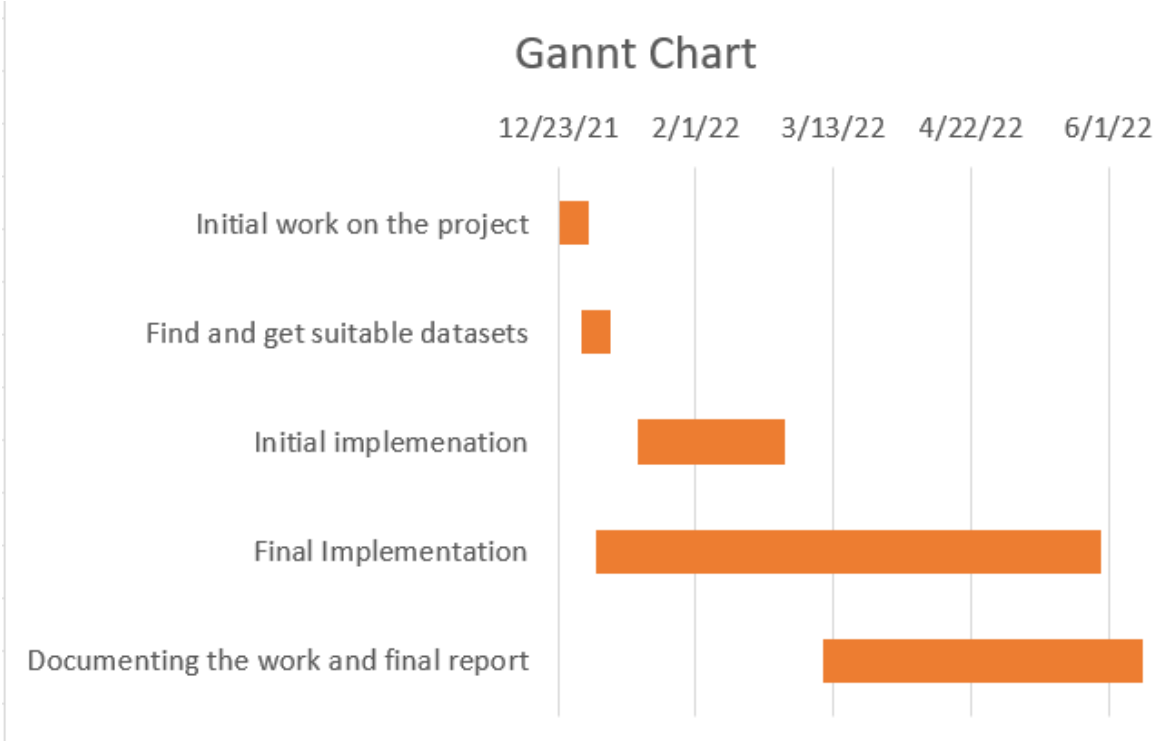

Figure 18 Gantt Chart

6 Design

The design of the neural network divides into two main architectures which are INT-CLS-GAN and AttnGAN which has previously discussed. There were limitations on the trails as the simplest text-to-image GAN takes at least 2 days of training to give initial results. Besides, large architecture such as AttnGAN has to have changes from the original architecture to match the available computer hardware.

6.1 GAN-INT-CLS

GAN-INT-CLS was designed as the first trial. The output image is 64*64 image. It is the first adversarial text-to-image model.

6.1.1 Generator

The generator network has two inputs the noise vector z and the word embedding vector of size 300. Both are concatenated together to generate an image. The embedding vector was observed to contribute to the image quality. As the generator network is conditioned upon it. Our experiments show that changing the vector to have tight differences by normalization resulted in less sensitivity to the words in the input sentences. By increasing the dimension and using wider embedding range such as the output layer from BERT the sensitivity to the word increases. The generator must learn the weights and the directions of the embedding vector. The caption “this bird has red feather” must move into the dimension in the latent space where it has red birds. This process of learning is adversarial with the discrimination that is caption aware that will refuse the image if it looks like areal bird but has a blue feather. This process enforces the generator to generate conditionally on the word embeddings as described abstractly in [5] more extensive discussion in [3]. The idea is visually described in figure 19 that the latent space can affect the output. The generator matches the probability manifolds that is called “Generator interpolation”.

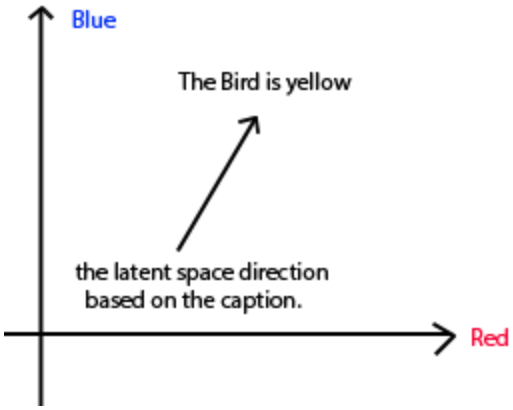

Figure 19 shows how the generator image moves in latent space

6.1.2 Discriminator

The discriminator is designed to have two tasks. The first is "is this image real?" the second "does this image matches the input caption". The generator is feedback with the results real or fake based on the answer of these two questions. It can only be real if the first and second is true. Moreover, generating a realistic image that does not follow the text description will be regarded as fake. This discriminator is called "matching aware discriminator". The discriminator loss is defined by the binary cross entropy for the results as it outputs only real and fake.

Table 2 shows the output of the discriminator

Image statues	Caption Statues	Correct Prediction
Fake Image	Real caption	Fake
Real Image	Fake Caption	Fake
Fake Image	Fake Caption	Fake
Real Image	Real Caption	Real

As it is shown here the discriminator roughly acts as and an AND gate. The generator job is having input of real caption and it should output an image that look real to deceive the discriminator. As both networks are training both are getting better with time the quality of the images are improved as well and this is called adversarial training.

6.1.3 Drawbacks

- It outputs a very small image 64*64 which make it very difficult to have good scoring.

The known metrics for the generative adversarial networks are the Inception score and the Fréchet distance. These metrics are based on the inception network pretrained models. The inception network has minimum size of 299*299*3 which means the images should be up sampled by 367% which obviously will not yield in any good quantitively means. The ways it could up sampled by is either using bicubic interpolation which will not yield in any better solution in this case as the up sampling required is very higher or using super resolution network which will add in some information that was not there. Such up sampling is very challenging as the information in quality principle assures that no extra information can be added unless it has already been there. The super resolution model will forcibly add in hallucinated details that was not generated by the model. Thus, this hints the images should be generated for a size close to the 299*299 to be effectively measured. This was the research starter for the stackGAN.

- There is no attention in the details.

  Despite the generated images moves freely in the latent space there is no multimodal attention. Furthermore, adding description of "red head" or "red body" will not make noticeable change in the generator. The word red and head will have a meaning in the latent space of the colors these changes does not refer to any location. Besides, the small size of the output image hides this point as it will not be very observable in smaller images. However, scaling them will show this point. The changes can happen to work if the generator infer the location from the dataset. However, this inference is not robust to generalization. Thus, small changes form the "typical" dataset captions will result in different images that the text description may not be perceptually accurate in the generated image.

Sample results:

Input: "this bird is blue"

Figure 20 results of GAN-INT-CLS

6.2 Attention Gan (AttnGAN)

The attention gan was one of the greatest contributions in the field of the adversarial text-to-image. It deployed attention model in the generation process. They proposed a new architecture "Deep Attentional Multimodal Similarity Model" (DMSM) which target to measure similarity of the sentence to the image. This DAMSM model is used as a loss function for the generator so that it adapts the attention to improve the semantic of the image and gives care for the other details. This process makes it more sophisticated than simply moves in a latent space for generation.

6.2.1 DAMSM

The DAMSM is composed of two deep learning models the text encoder and image encoder. The text-encoder network is a simple bi-LSTM that extracts the meaning and semantics of the input text and give it to different two hidden states in order. The last hidden state of the bi-

LSTM is used in the global sentence attention. The image encoder follows the classical usage of convolution neural networks which is feature extraction. The efficient yet most effective way to do that is to use transfer learning. Specifically, using inception model in order to get the global features and local features of the image. The local features are extracted from the first layer and the global layers are extracted from the deep layers.

Then the DAMSM model starts with matching every word in the sentence with every subregion in the image and retrieve the dot product similarity as shown in formula five. Then the s , similarity matrix is normalized. Finally, the cosine similarity is computed after matching the text to the regions of the image.

$$s = e^t * v \ (5)$$

A closer look at DAMSM it can be found that it learns in a "semi-supervised" way. The supervision is that it is granted that this image has this caption. The DAMSM has to figure out which word refers to which region. This is the unsupervised part as the network has to train to learn the weights for each word and location.

6.2.2 Generator

As shown in the previous section the generator can ideally generate a 64×64 images. The upscaling is done by "stacking gans". Thus, there are three generators that outputs $64 \times 64, 128 \times 128, 256 \times 256$ respectively. Moreover, there is no limit on how big the image can be generated. However, the paper stopped at 256×256 which is ideal to test and train the model. Increasing the model size severely increase the trainable parameters in the network. The size of 256×256 is good size as it can be easily interpolated to 299×299 which is the minimum input for the inception network. Such interpolation will not give big damage to the image quality as it is very small. The generator network gets the noise vector z along with conditional augmentation vector c, except for the first generator, supplied with attention vector f_{attn}. The attention model requires an input image with a sentence more details are discussed in Appendix I.

6.2.3 Discriminator

The discriminators here can work in parallel as each scale can be trained independently since they are not dependent on each other. This property makes the attnGAN training a very fast

process. The discriminators are having two questions to answer, “is this real?” this formulated as “unconditional loss”. “Does every region match the sentence?” this is the conditional loss. The discriminator should catch the generator if generated a realistic image with mismatched **regions**. This emphasis the difference between AttnGAN and any other GAN developed before AttnGAN. As the previous state-of-the-art was checking if sentence matches the sentence as GAN-CLS in “matching aware discriminator”. However, the mismatched text and images are still used in AttnGAN to train the discriminator on the region not the entire image at once.

6.2.4 Drawbacks

- No Backwards checking

  Caring for image based on the text regional instead of global is great idea. However, regional synthesis can introduce the challenge that the image can violate the global spectrum. This issue can be improved if there is a loss function for re-captioning the generated image. This was already a published as MirrorGAN as it recaption the image and check if it still has the same caption as intended or not. As shown in Figure 22, the bird has two heads

Figure 21 Output image from the AttnGAN issues in the structure of the bird

Example, figure 23 and 24, of AttnGAN “this bird has a blue head and red body”. Shows the plots of attention and how each word matched a region in the image.

Figure 22 Output image from the AttnGAN

Figure 23 shows the attention plot for the image generation

7 Implementation

The project is written in python 3.7 on a docker environment.

7.1 Dockerization

One of the main challenges in the development of any AI system is the environment configuration. The libraries are having dependencies tangles and challenges to get many environments to work together. Besides, it allows easier collaboration as the docker image is roughly defined as a lightweight virtual machine. Hence, having the same docker image will grantee same exact output everywhere. Moreover, it gives faster model deployment. Thus, the docker image used also allowed easier deployment of the GPU without wasting time was traditional method as the used image has all the configuration ready. Otherwise, the entire environment has to be fitted with the correct CUDA version with the correct library dependencies. The docker allowed easy mix of Tensorflow code and PyTorch code in the same project easily without separation in the environment.

7.2 Code Organization

The code is organized in files which can be easily generalized and managed easily. Along with using configuration files that allow the changes to occur globally without caring to adapt the code. This was beneficial for downsizing the models to match the available hardware. It was mainly limited to the low memory size of the GTX 1660TI of 6GB. The GAN models require high memory as there are three models (generator, discriminator, adversarial) to be trained each has separable trainable parameters.

7.3 Tensorflow

Tensorflow was used for preprocessing and training GAN-INT-CLS. Thus, all trials were built on the top of Tensorflow and Keras API.

7.3.1 Preprocessing

The preprocessing was based on the data augmentation by creating different orientations for the images. The images were zoomed in to focus on the bird inside it and reduce the effect the data bias of the background as the important part is to generate the bird especially on the size of 64×64. Furthermore, the image is too small to draw a context for the image.

7.3.2 Usage of BERT and Inception

As it is clearly known that TensorFlow and Inception are built by Google. This is makes Google's model is easily integrable with the TensorFlow code. Thus, the model was loaded easily and used.

7.3.3 Speeding up execution time

The execution time was accelerating by using the idea of *dynamic programming "store results are faster than calculating them"*. The repeatedly used data and information were stored in files that are loaded in the memory which contributed to a highly accelerated training. This was the storage-processing tradeoff. The computer used for training had 32 GB RAM which made it easier for taking advantage of the available memory without any crashes.

7.4 Gensim

The genism library was used in order to create the skip-gram and CBOW from scratch on the training data as it has a good class built in with the functions needed to work quickly for the module as their classes has their own structure stored and easily modifiable.

7.5 PyTorch

PyTorch was used for preprocessing in the instance selection case. The AttnGAN model is entirely written in PyTorch as prove of capability to use to two different frameworks in the development.

7.6 Tensorflow and Pytorch Mix

This project *has uniqueness of combining the two main Frameworks in the AI which are Tensorflow and PyTorch*. However, it is quite rare to use multi-framework on the same project. For this case was because the official implementation of instance selection for GANs were written in PyTorch. *The contribution made was the mixing of both ideas in the code as it is known that Integration of TensorFlow and PyTorch is a non-trivial task*. To Achieve that serialization of data so that a medium of communication is established between the two Frameworks which made the project more powerful.

8 Trials, Results, and discussion

8.1 Initial Thoughts, Results and Generic Problem Discussion

8.1.1 Discussion: Are the Available Datasets good?

There are three datasets used on the text-to-image tasks for a fair comparative study of the models which are CUB 2011 birds dataset [35], Oxford-102 flowers dataset [36] and the MS-Coco dataset [37]. Most commonly, the CUB is used for showing image quality and the MS-COCO is used for proving of generalization as MS-COCO is a huge dataset. In this project, only CUB birds were used as MS-COCO is very huge and generative adversarial networks require a very high processing power. The CUB bird dataset is consisting of 11788 images of a different birds of 200 categories/breeds. All the images are captioned with multiple captions.

8.1.2 Result: GAN-INT-CLS

As it has been shown in the design section how the text contributes to the image by roughly moves the latent space direction. This section will focus mainly on the image fidelity. This model was trained for 1000 epochs took 3 days on the entire dataset. Here are three different captions for the image.

Input: “This bird has red head and blue body”

Output:

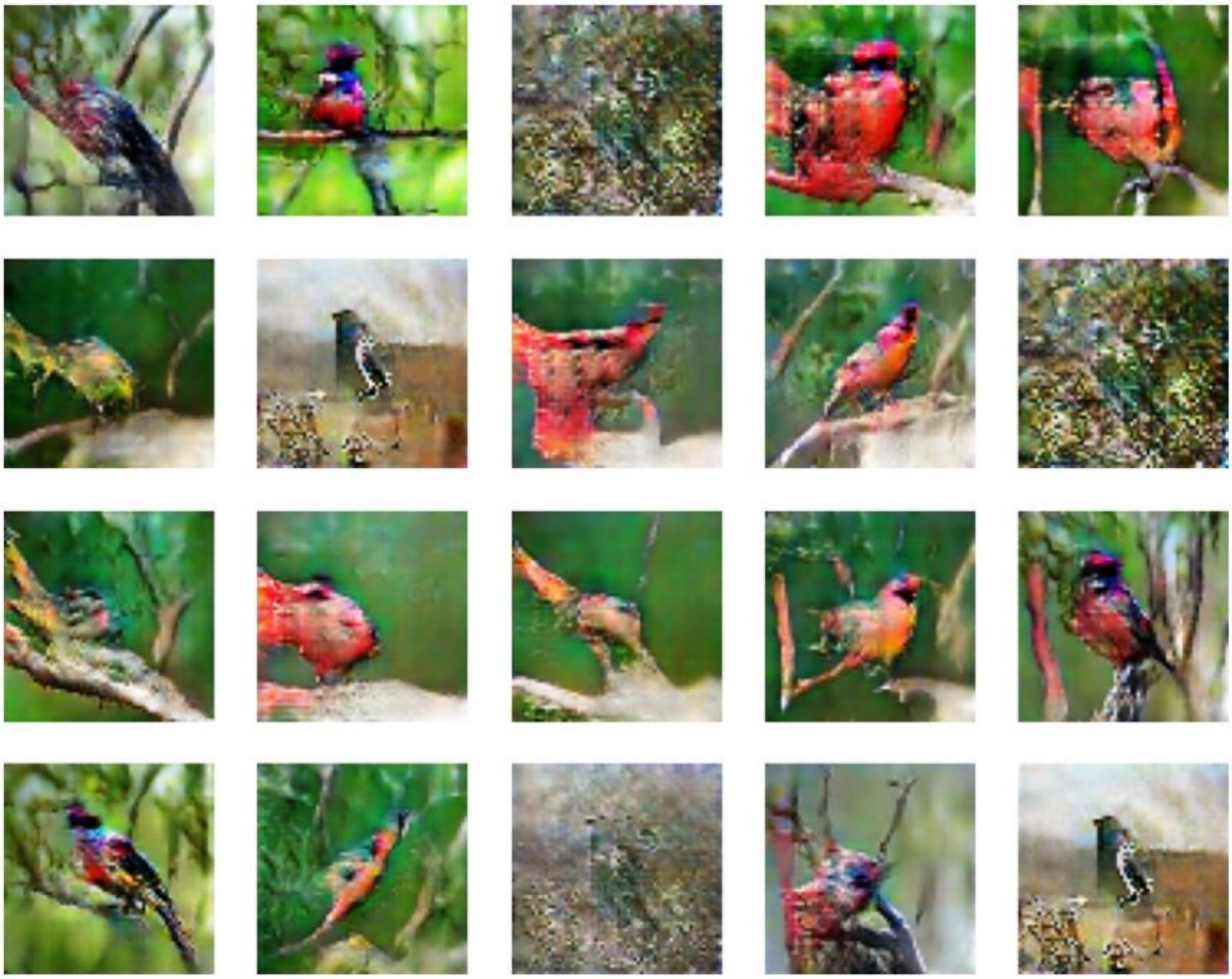

Figure 24 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

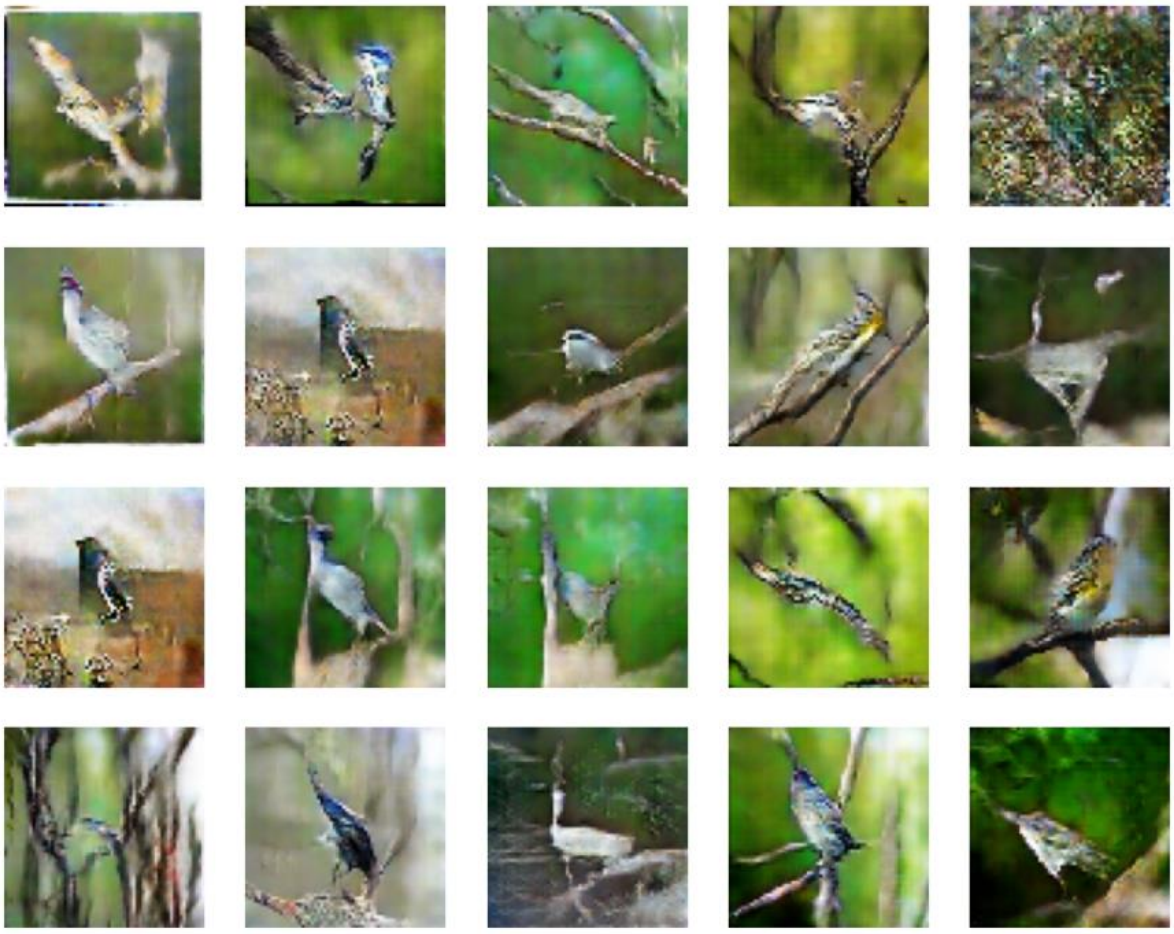

Figure 25 This bird has white head and blue body

Input: “This bird is yellow”

Output:

Figure 26 this bird is yellow

8.2 Instance Selection Thinking (Think of Generation as Classification)

8.2.1 Discussion: Generative Datasets and Model Size

Despite there are available datasets for GANs and other models. These datasets were not built on purpose of generative tasks. The “failure” samples of generative adversarial network are a result of weak representation of sparse/low density regions manifolds in the dataset [39]. The reason for these cases is the dataset were designed to train image captioning algorithms. However, generating text that describes the image is very different from generating an image that matches a text description. Thus, a higher model capacity is sometimes attributed for absorbing the capacity of these models. Moreover, the Generative adversarial model could be smaller yet have the same results or better with half of the trainable parameter if does not have sparse region or bad representation of such models. There were attempts to overcome this failure by post-processing.

These methods were running by rejecting bad samples by using truncation trick or still use the discriminator verdict to judge the results as in [40, 41, 42]. These post-processing methods does not increase the efficiency of the model training; in opposition, it is less efficient as the generation task takes more time. The preprocessing was introduced by DeVries et al [39] by using the idea that is commonly used in classification methods which is instance selection. This method ejects the badly represented datapoints and sparse region in order to have better model training that can care for denser data manifolds. Their experiment shows improvements in Inception score, and Fréchet distance. More importantly, the training time and hardware required is reduced significantly as the model size is reduced. However, the drawback is the diversity of the generated images is reduced heavily. There is a trade-off defined by the retention rate hyperparameter.

8.2.2 Trial: Instance Selection on INT-CLS-GAN

Input: “This bird has red head and blue body”

Output:

Figure 27 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

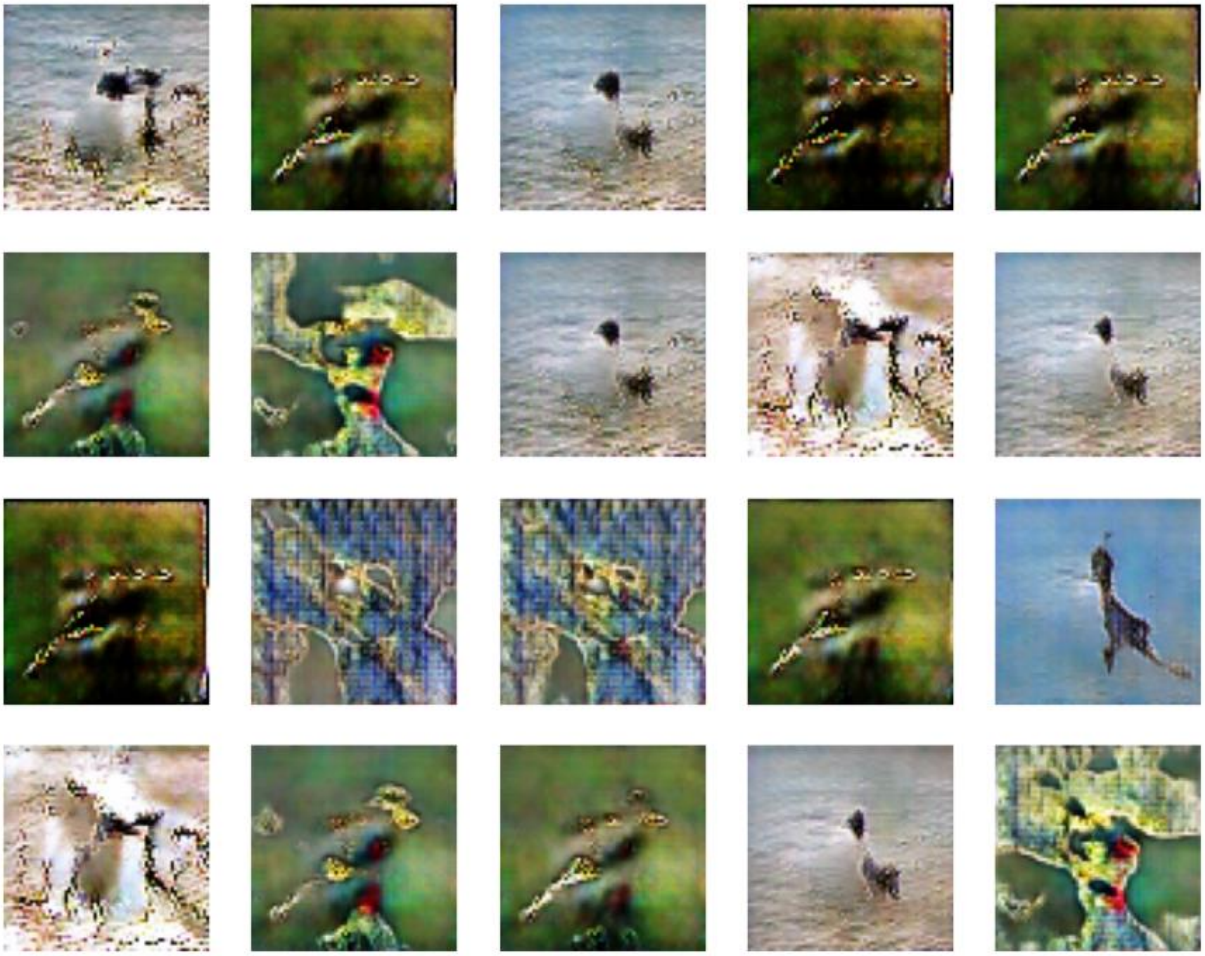

Figure 28 This bird has white head and blue body

Input: “This bird is yellow”

Output:

Figure 29 This bird is yellow

It can be shown that the model despite the odds of being better is not as expected. There can be two reasons for such behaviour. First reducing the captions and the images as the elimination processes occurred based on the images only not the captions as it was assumed as good. Second the model is trained on 671 epochs only which is 329 epochs fewer than the original one which justifies the performance difference. As it is observed the image are less diverse than the other as mentioned in the paper. On deeper analysis one can observer a better features image which potentially good for a stack-based architectures.

8.2.3 Trial: Increase Model Complexity (Instance Selection Power)

As described in the previous trail that the model trains faster now however the output is worse than the original. The model can be more complex on the available hardware. By adding batch normalization before ReLU layers as it is a common architecture practice.

Input: “This bird has red head and blue body”

Output:

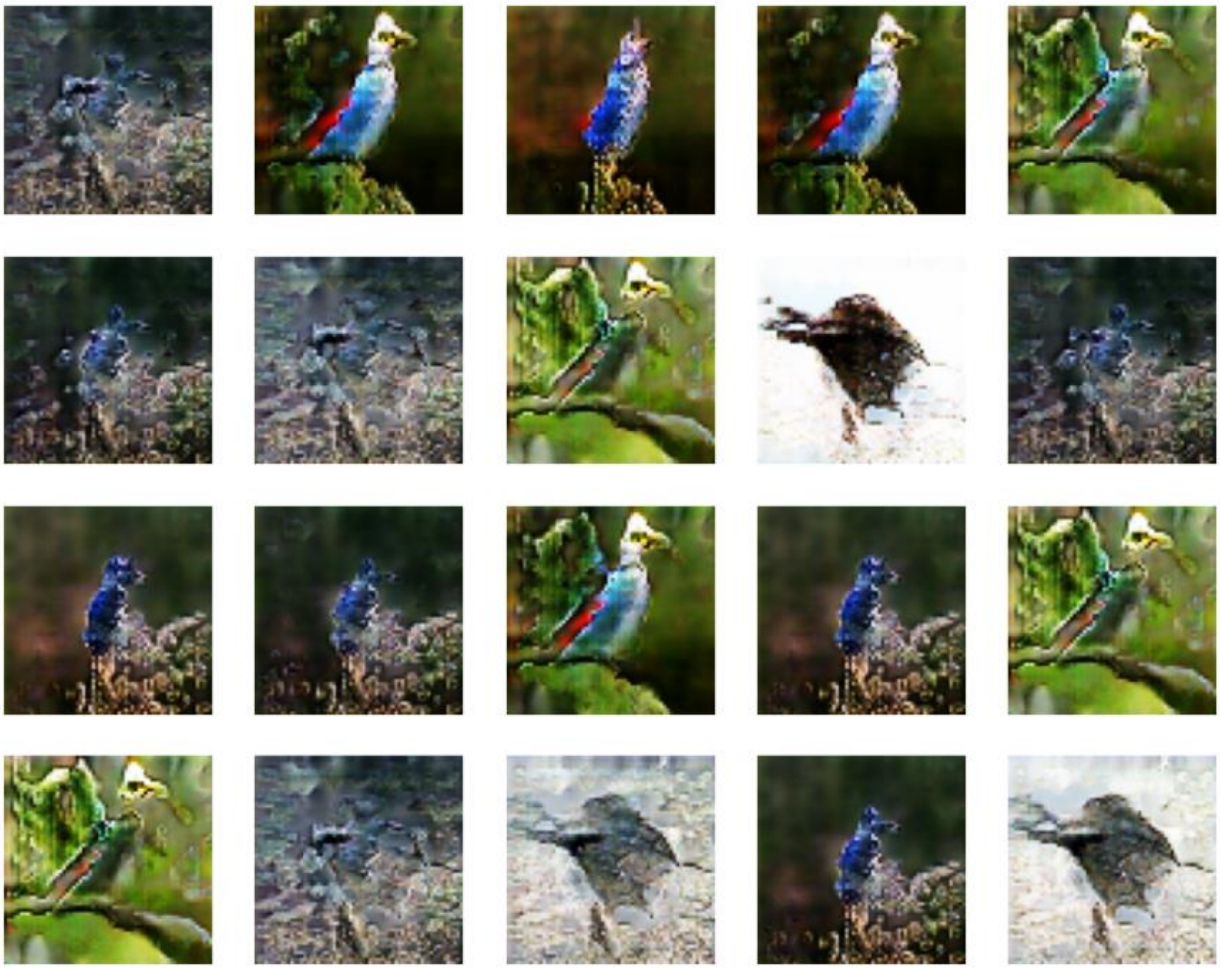

Figure 30 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

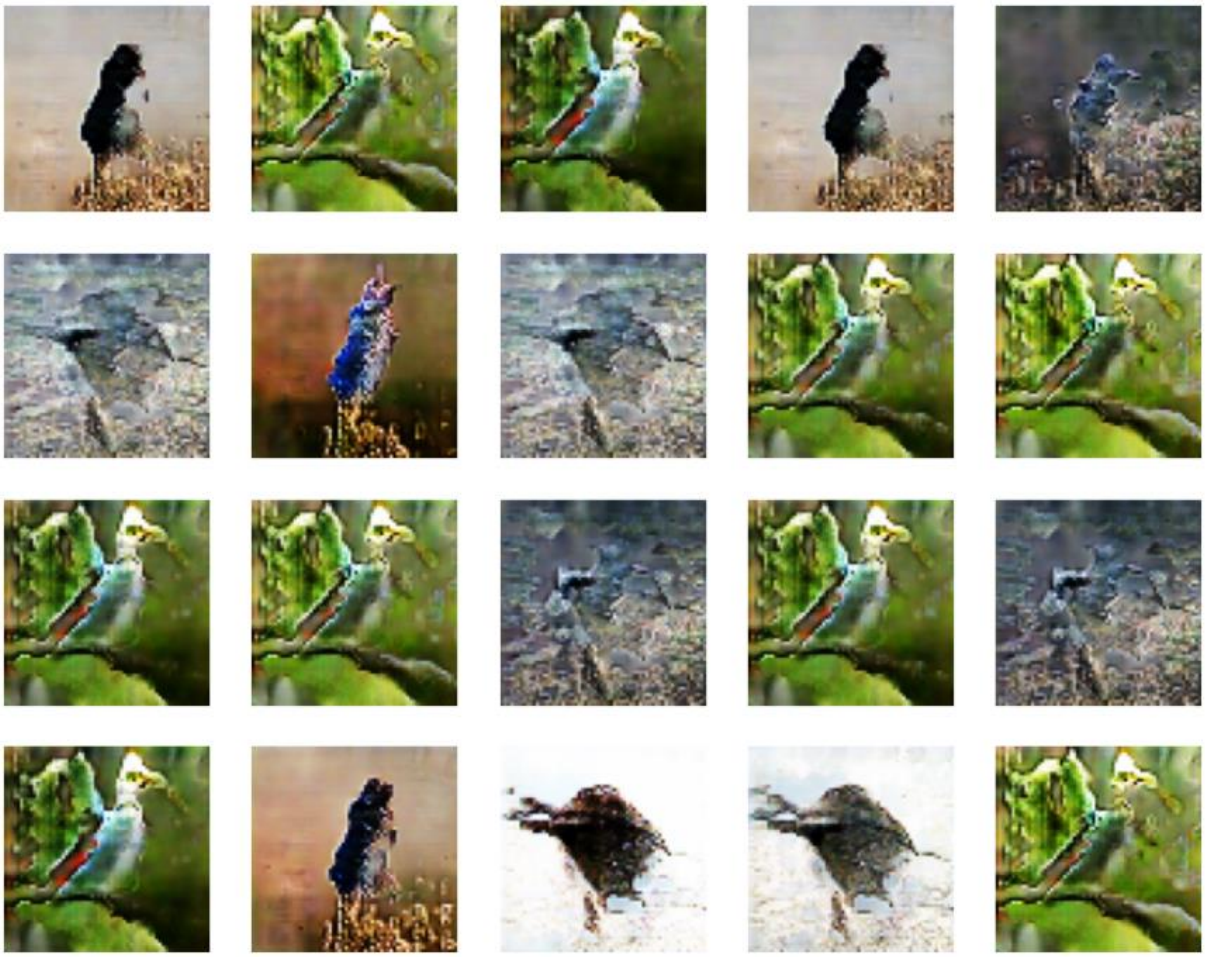

Figure 31 This bird has white head and blue body

Input: “This bird is yellow”

Output:

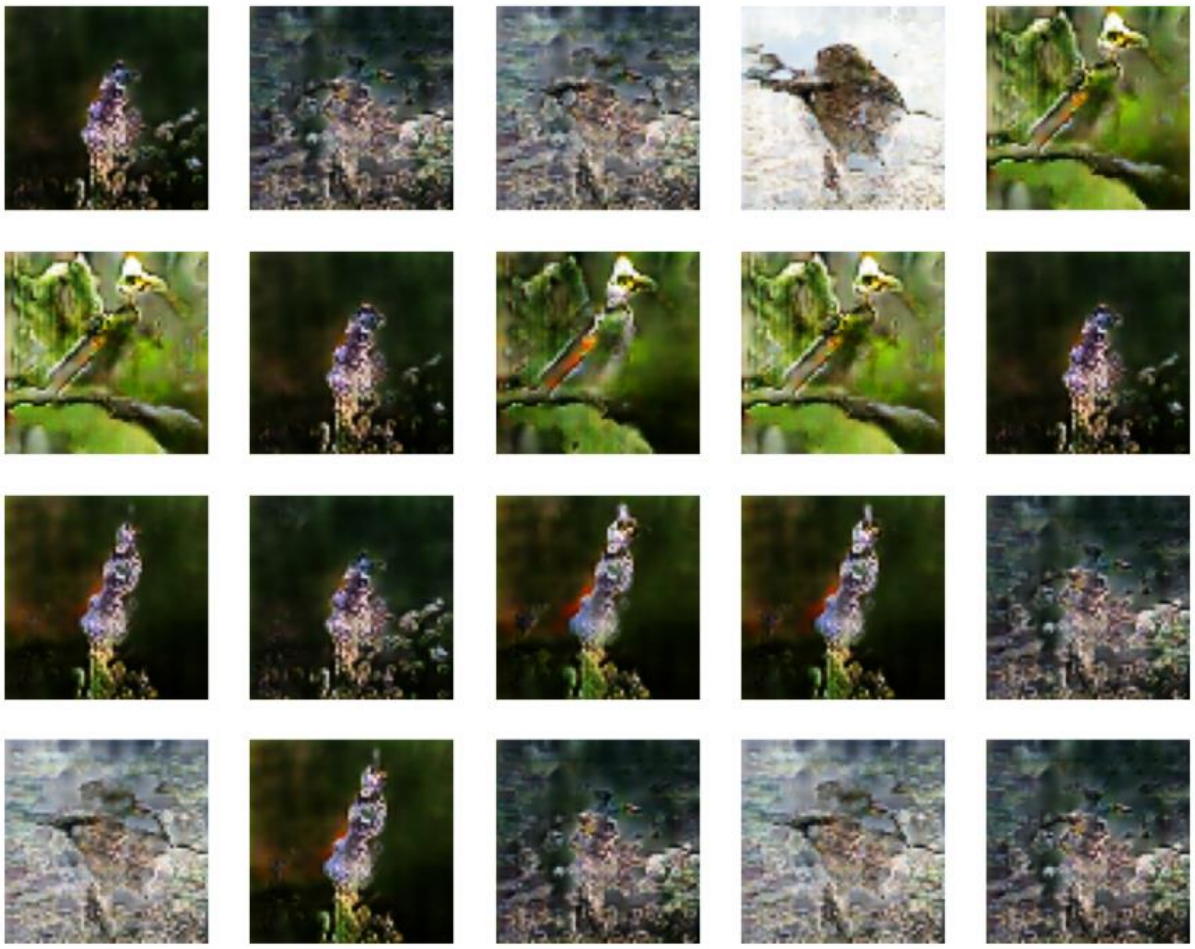

Figure 32 This bird is yellow

This model was trained for 731 epochs it can be observed that it has better match for the text so far compared to the original and the first instance selection with fewer epochs. As shown, there is a mode collapse and very little diversity.

8.3 BERT (Text-Classification for Better Generator Latent Space)

8.3.1 Discussion: Bigger Embedding vector and using BERT

As discussed before the GAN-INT-CLS is driven by the embedding vector to create different images and diversity. Thus, can bigger dimensions contribute to a better result? The training was done by using Google News Bin word-to-vec model which outputs dimension of 300 this model is somewhat limited to some specific domain. However, news is still diverse enough to have most of the vocabulary. Moreover, Google has a better model that is trained for

text classification which is BERT. This was used to experiment taking the output of BERT mini's last layer pooled output. Furthermore, it gives a vector of size 768 which will make the embedding vector domain larger and has many dimensions. The question arises is if the model capacity can absorb such change or not? This was examined by the following trial. It seems that using higher dimension array using BERT output for classification does not work as well in the generative tasks

8.3.2 Trial: BERT Pooled Output as Embedding For GAN-INT-CLS With Encoding

The encoding step was a follow for the paper advice to keep the embedding size constant as three hundred, so it was encoded with dense layer of 300 then the output of it is fed into the network as the embedding vector while maintaining the same architecture.

Input: "This bird has red head and blue body"

Output:

Figure 33 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

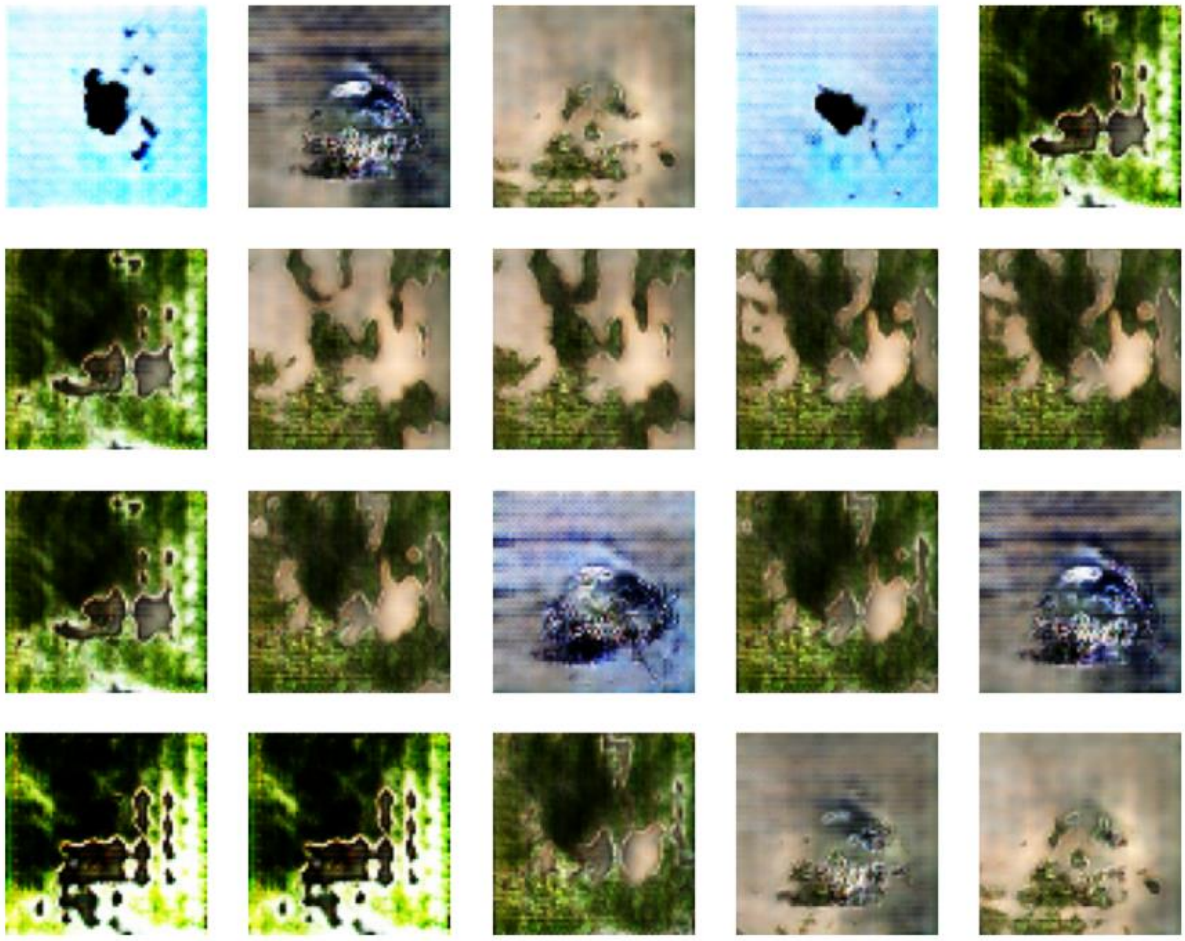

Figure 34 This bird has white head and blue body

Input: “This bird is yellow”

Output:

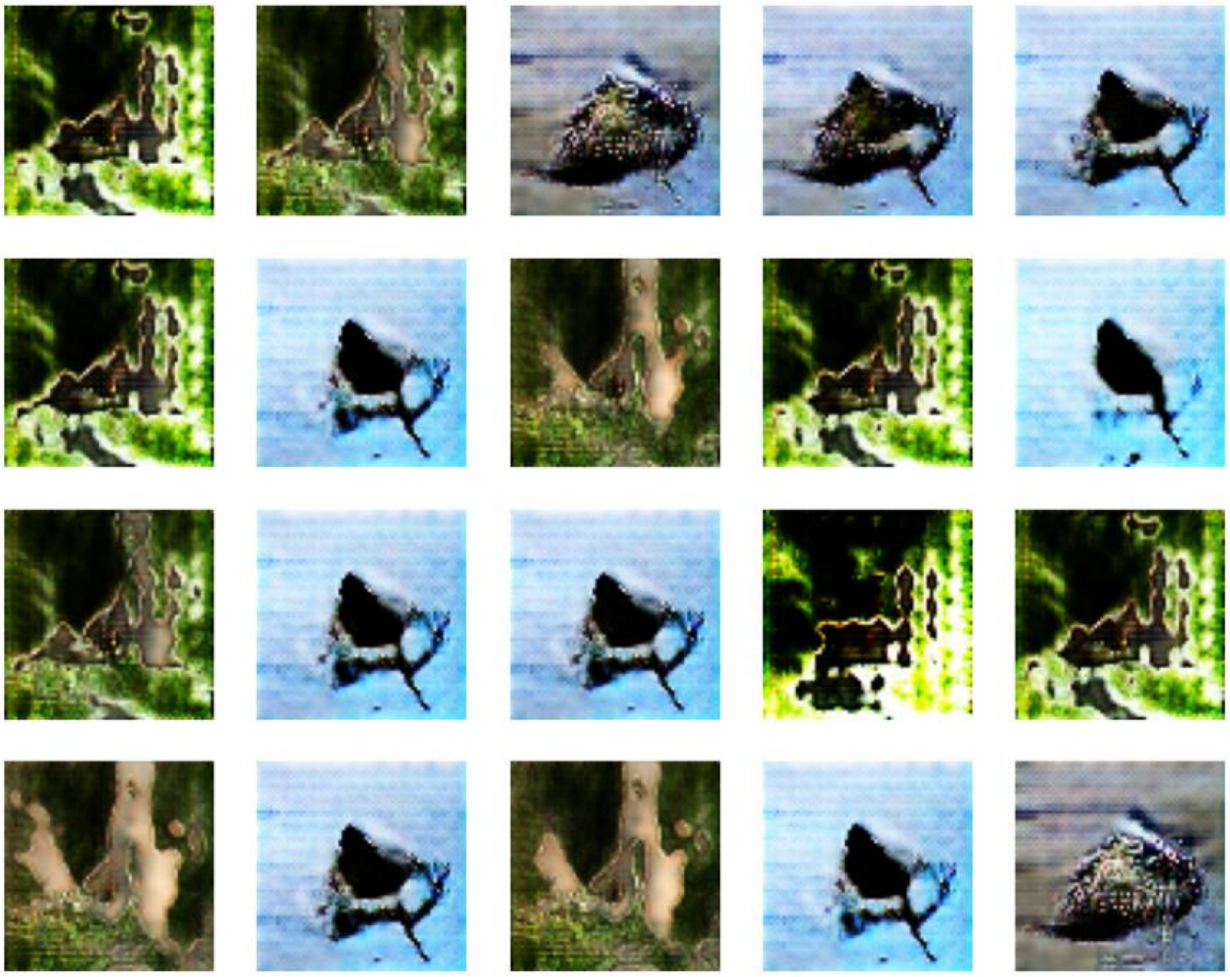

Figure 35 This bird is yellow

It can be shown that these results are not satisfying and yet it is no diverse so that the idea of using BERT classifier output is not a good add for the generative tasks.

8.3.3 Trial: BERT Pooled Output as Embedding For GAN-INT-CLS Without Encoding

Perhaps the results are qualitatively bad because the change of the embedding. Thus, in this trial the BERT embedding will keep its size 768 to check if the vector size will differ or not.

Input: “This bird has red head and blue body”

Output:

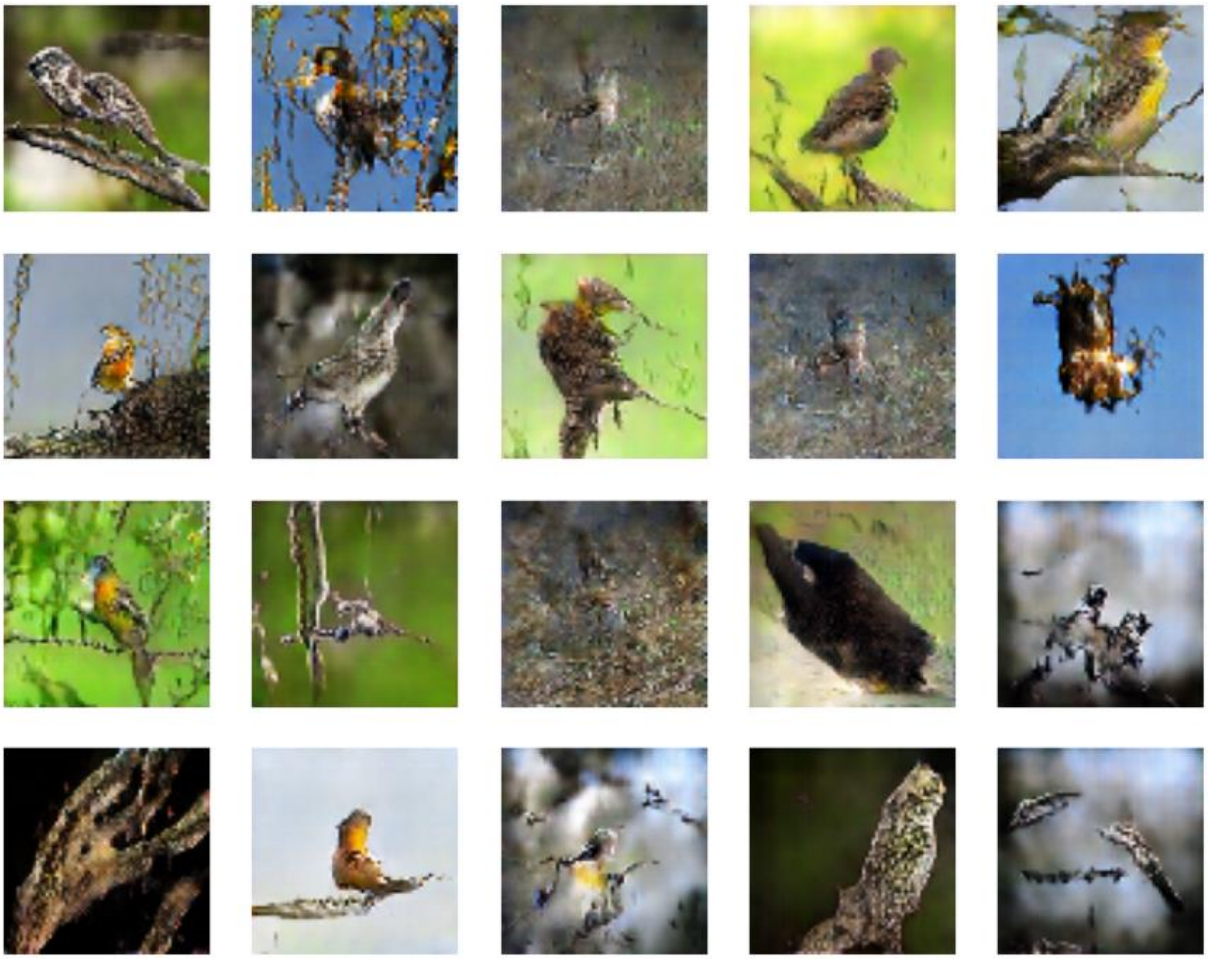

Figure 36 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

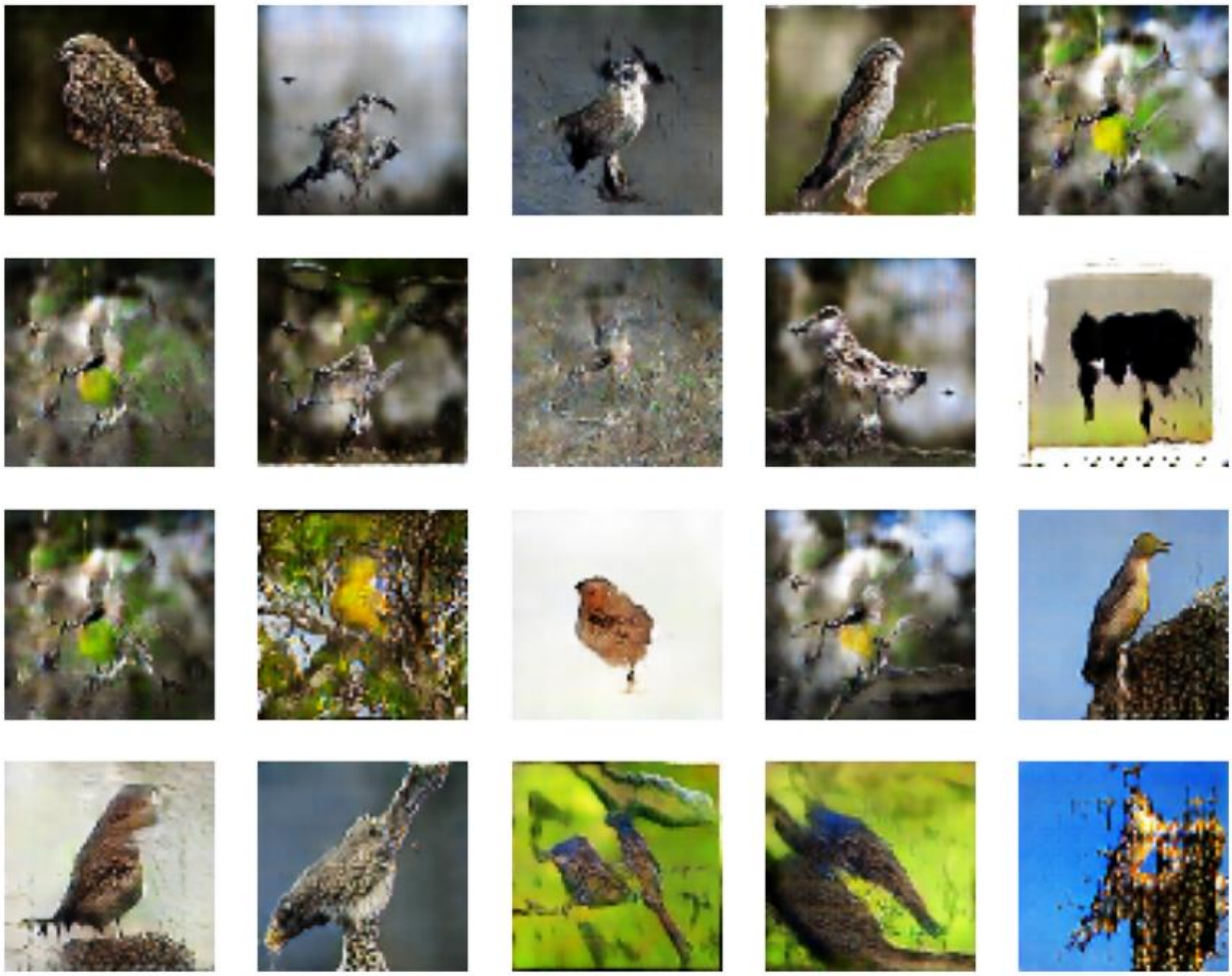

Figure 37 This bird has white head and blue body

Input: “This bird is yellow”

Output:

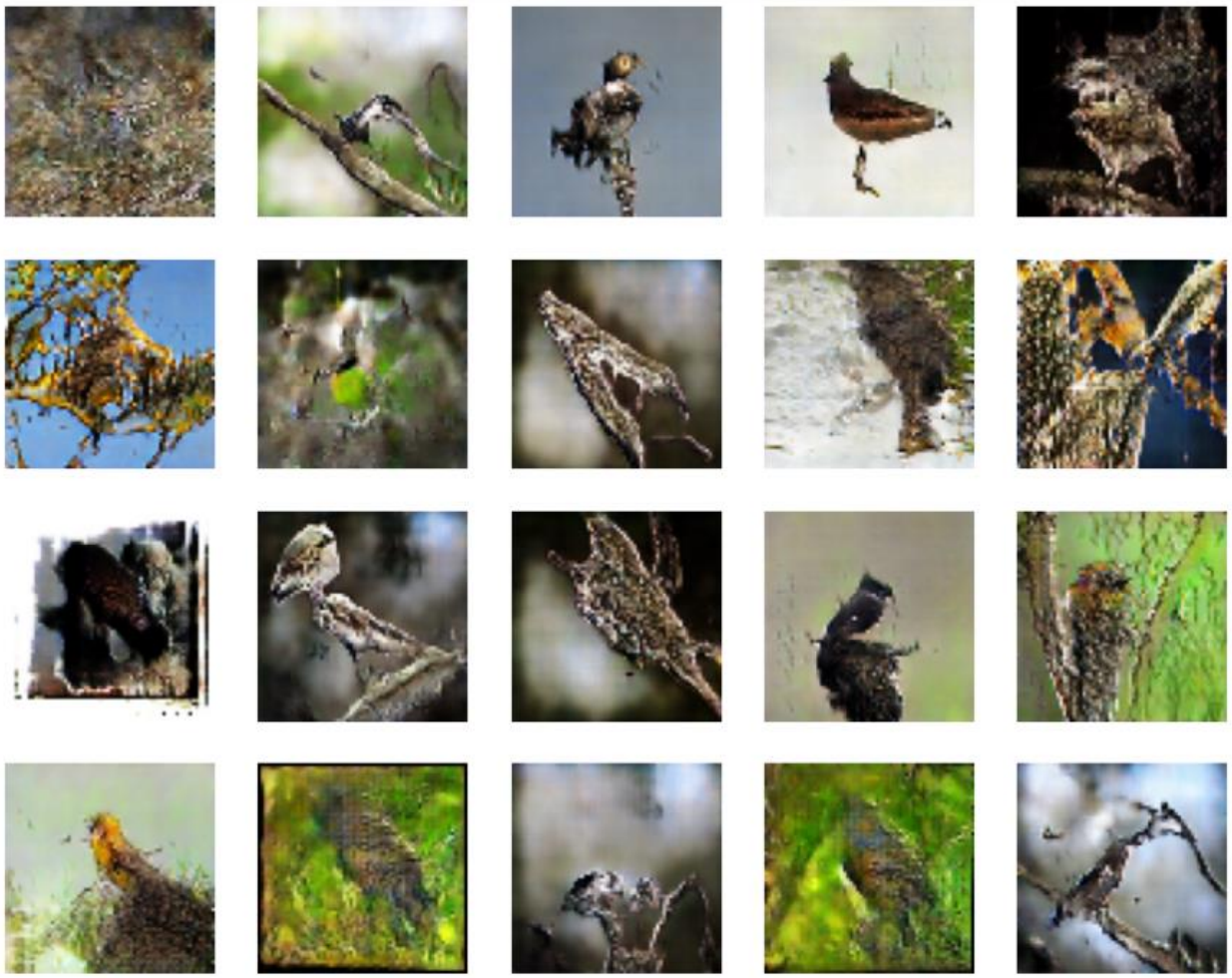

Figure 38 This bird is yellow

It can be concluded that the image fidelity and diversity is higher in this version. However, it is very clear that the image does not match the text description quite well. The reason is that the embedding size made it very hard for the network to train to know the text meaning and move freely in the latent space.

8.4 Create Customized Embeddings from Scratch

8.4.1 Discussion: Manually Train a Word-To-Vec

In all literature usually the text embedding relies on a pretrained embeddings. Besides, all the training trails were based on pretrained embeddings which is very generic and does not reflect the vocabulary and relations of the used dataset. However, this dataset, CUB-2011, is relatively small specially in comparison to the datasets used for creating text embeddings. The text

embedding requires a great data to be collected in order to give a better vector representation for each word. Thus, this trial has already some concerns. However, a better work could be done on this trail if there are available and scrapable caption about bird description on the internet to support the training process. This is an exhaustive process and needs deep work to be achieved and verified. This can be left for future work as it is out of scope and require team of developers.

8.4.2 Trial: Continuous Back of Words Manually trained

Input: “This bird has red head and blue body”

Output:

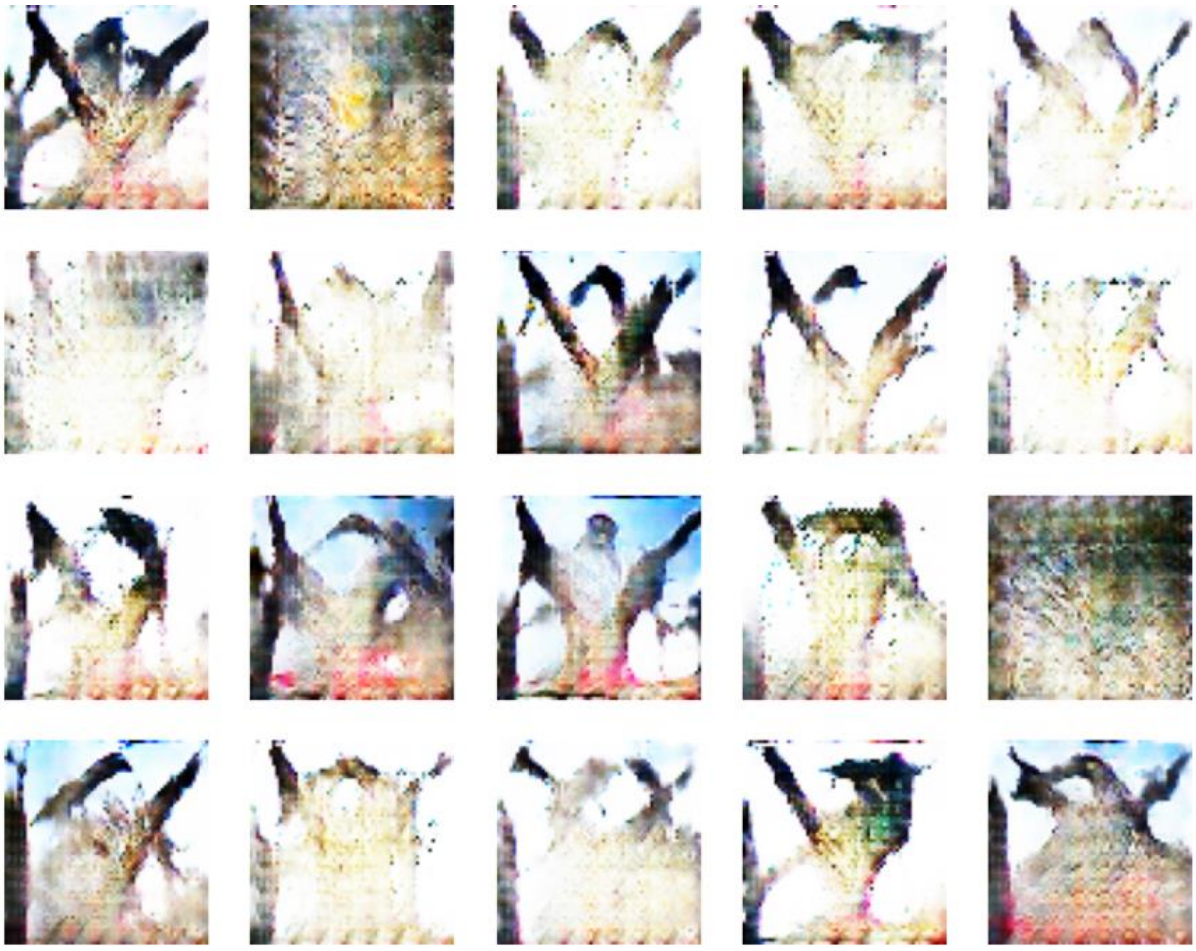

Figure 39 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

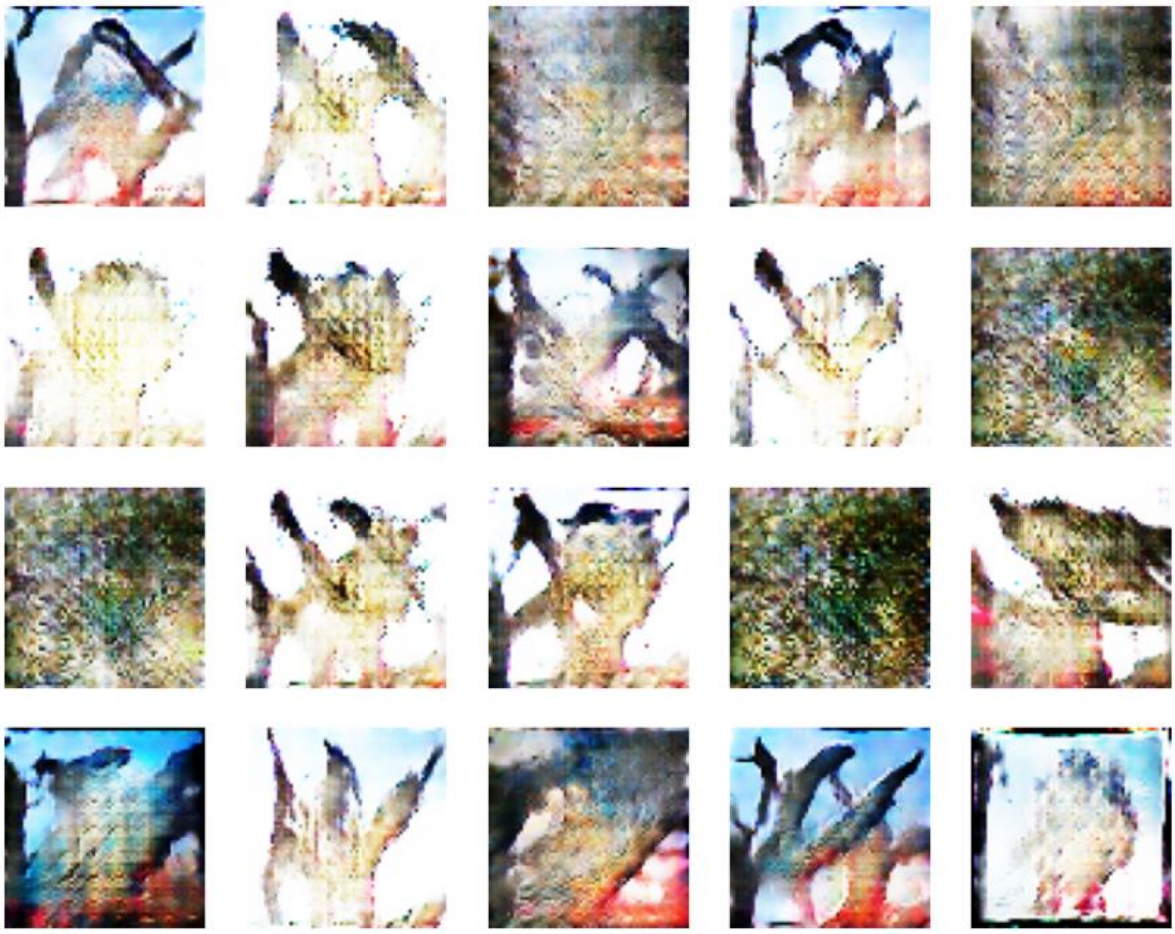

Figure 40 This bird has white head and blue body

Input: “This bird is yellow”

Output:

Figure 41 This bird is yellow

It can be observed that that the CBOW gives high texture and features for the bird structure however it gives no diversity nor consistency for of the text given. Thus, the Continuous Back of Words is not appropriate for this case and with the available dataset size.

8.4.3 Trial: Skip Gram Manually Trained

Input: “This bird has red head and blue body”

Output:

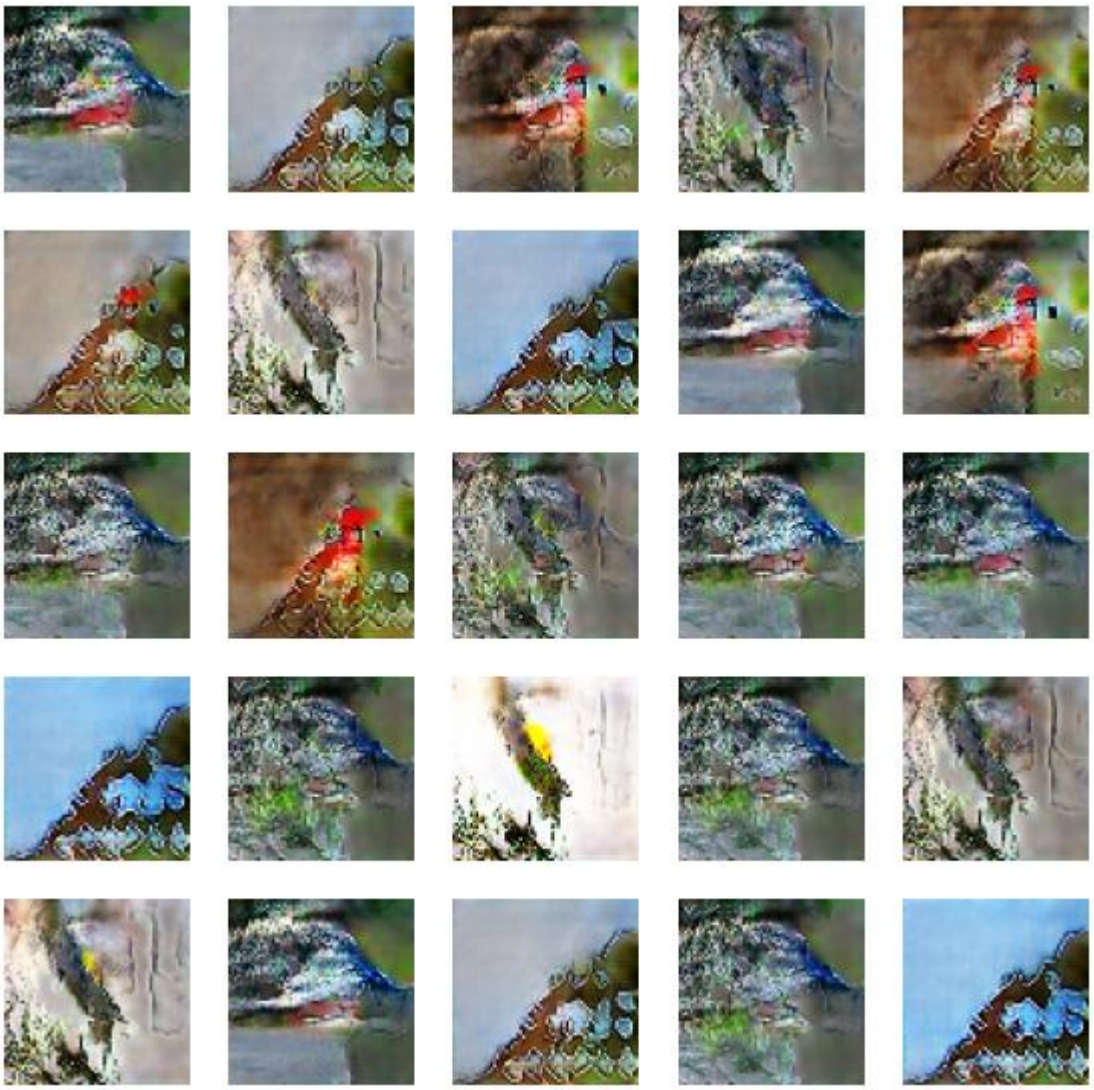

Figure 42 This bird has red head and blue body

Input: “This bird has white head and blue body”

Output:

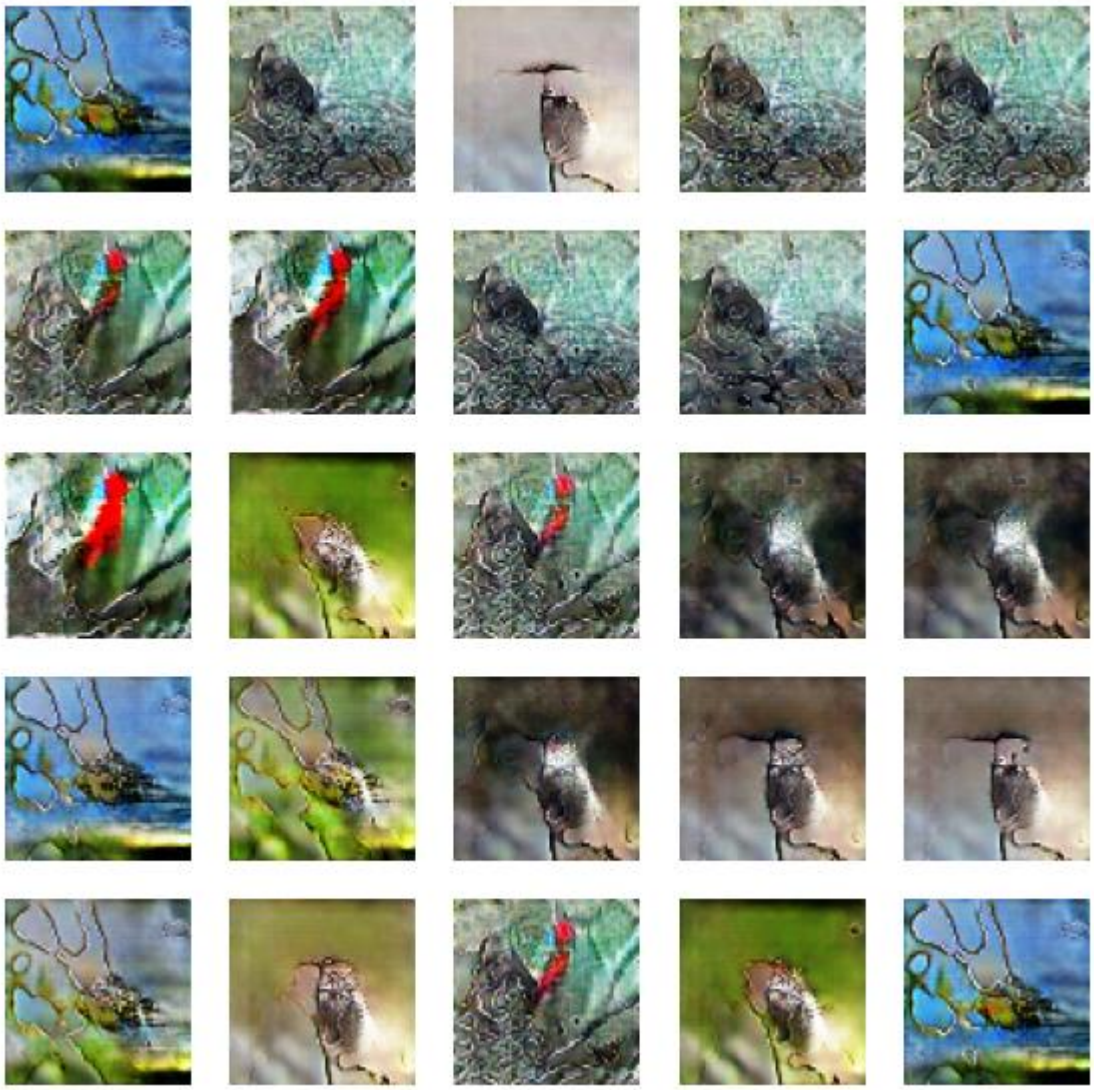

Figure 43 This bird has white head and blue body

Input: “This bird is yellow”

Output:

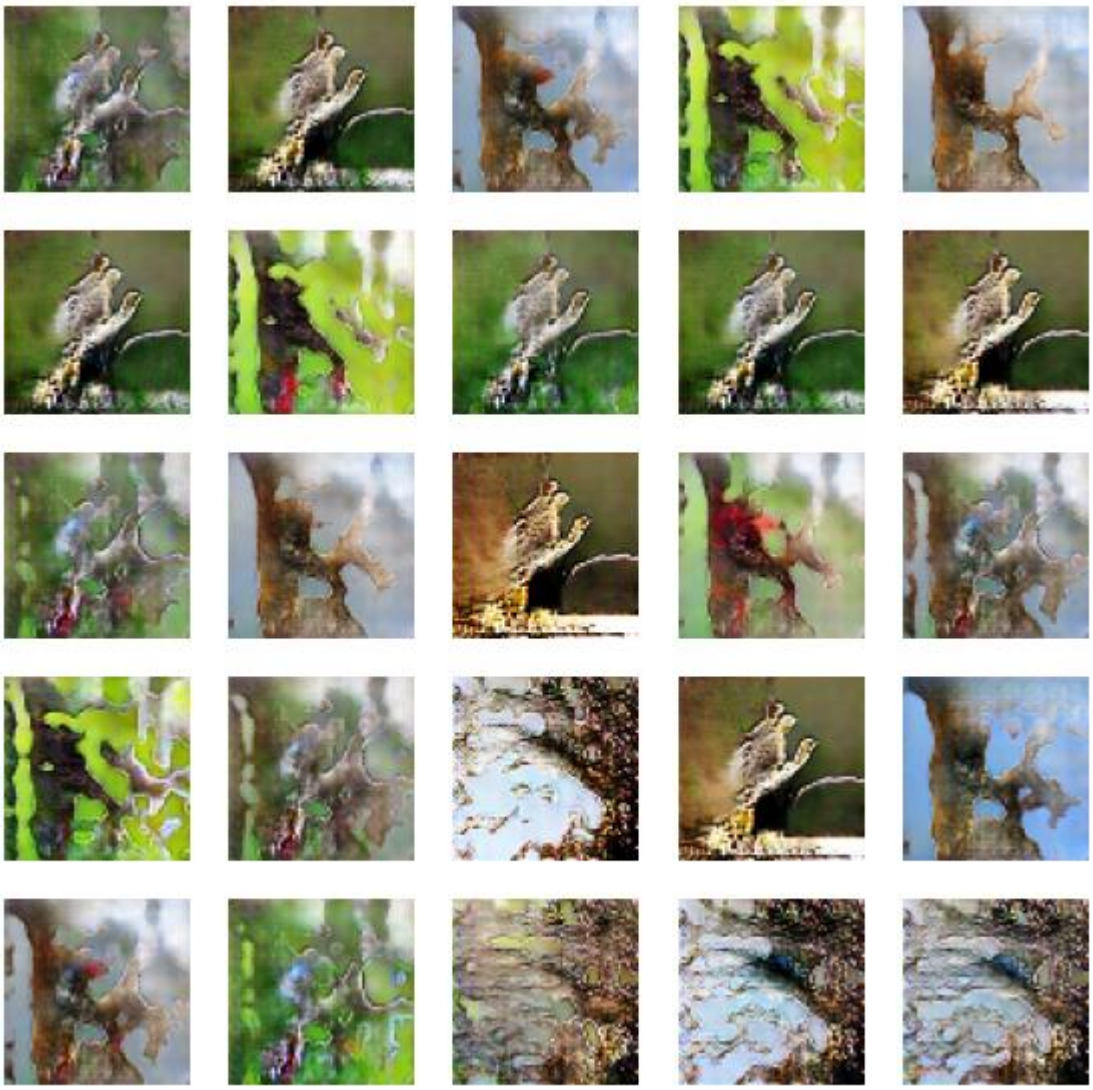

Figure 44 This bird is yellow

It can be shown that the skip gram embedding is better at caring for the text and reflect it in the image, however, the output looks nothing like a bird. That concludes, manually training a word embedding specialized on the training with small size is demonstrably wrong.

8.5 Divide the Problem into Stages and Use Attention (Reverse Captioning Problem)

8.5.1 Discussion: Why Stage Generators Are Better and What the Drawbacks?

Besides the obvious reason which is having a quantitative measure instead of relying on the manual human qualitative measure and clearer the output will become. The inferred reason

from literature is that every stage has its own features that cares about. The first image, as shown in the previous sub-sections, never gives a realistic looking image. Furthermore, the images can be interpreted as “looks like a bird” but it is clearly synthesized. The target of Generative Adversarial Network is diversity and realism. So repetitive results are very subject to occur in 64×64 images as the space of change are very little. Also, the editable regions are visual scale is also negligible. At, for example, 128×128 there is a bigger room for diversity. Moreover, the realism. Since the Generators and discriminators are stacked each can have a separate work. These contributes to a final layer of multi-level of discriminator realism. This discussion is based on the assumption that a one generator that gives a high image at once can be shown to be ineffective due to the vanishing gradient problem. Thus, generating a bigger size image is not an auxiliary visually pleasing option. It is a very important step. It will be shown in the final results and trials for the next sections. The main drawback of this approach is that the generated image heavily depends on the previous generation. Such that the quality of the i^{th} image relies on the $(i-1)^{th}$ image as it will be observed in the AttnGAN. The good samples get better the worse samples remains a failure. That indeed adds a great importance to the 64×64 image. The main features and the attention cannot occur without having a good steppingstone.

8.5.2 Discussion: How the Attention Models Improve the Task and How Is It Related to Image Captioning?

In image captioning the regions of the images, using self-attention, is defined for creating a word that describes it. This is a reversible process. Furthermore, the generator can draw independent from the other regions based on the attention for each layer in the network. Moreover, the words describe three types of the entity, color, and location for the bird synthesis task. Furthermore, the word “head” refers to a location or region in the image. Thus, saying “red head” means the location of “head” must be red. As shown in the figure 46. The rest of the image can be anything, but the region “head” must be red or reddish. The region training is done in a supervised manner. Perhaps, if image segmentation methods are used for improvement to make it supervised can lead to a better result. However, this is not experimented yet in this project nor the literature. Image captioning, in this case, can be used as loss function for assessing the image to know if the captions match the intended results using by comparing the results to the text input using BLEU score or ROUGE.

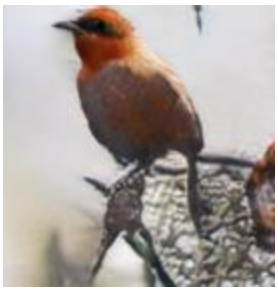

Figure 45 Example of AttnGAN output

8.5.3 Trial: Using Attention GAN for improving the image quality

Input: “This bird has red head and blue body”

Output: 64 × 64

Figure 46 This bird has red head and blue body 64× 64

Output: 128 × 128

Figure 47 This bird has red head and blue body 128×128

Output: 256×256

Figure 48 This bird has red head and blue body 256×256

Input: “This bird has white head and blue body”

Output: 64×64

Figure 49 This bird has white head and blue body64×64

Output: 128×128

Figure 50 This bird has white head and blue body 128×128

Output: 256×256

Figure 51 This bird has white head and blue body 256×256

Input: “This bird is yellow”

Output: 64×64

Figure 52 This bird is yellow 64×64

Output: 128×128

Figure 53 This bird is yellow128×128

Output: 256×256

Figure 54 This bird is yellow 256×256

It can be observed how the discussed idea about the hierarchical structure GAN is very observable. One can investigate the top left image in the yellow birds is that it barely produced a bird. As shown below, the failure on the first stage is reflected into the next stages

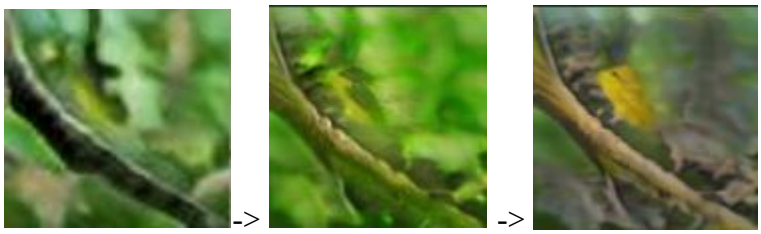

Figure 55 Evolution of bad samples

On the other side, the perfect images were good from the beginning. Thus, it can be concluded that creating the steppingstone is the most important step. Yet, it will never be a perfect image as seen in both, good and bad, results the first image was not perfect in matching the text, however, the structure of the first image was the deciding factor and every other step only improves the structure but never changes it.

Figure 56 Evolution of Good Samples

8.6 Connecting Results, Comparative Analysis and Future Work

8.6.1 Results: Final Best Results

The results for all trials are shown here. As discussed in section (6.1.3). The FID and IS cannot be assessed in the case of the INT-CLS-GAN as they produce a 64×64 image which scaling them up will damage their quality. However, it was attempted to assess them. Notice the original paper does not give official results, thus, assessing them is based on the personal trials not by the original publishers. *However, a workaround the constraints were done to get the FID score by using Inception model from TensorFlow that has minimum 75×75 other than the default of 299 × 299.*

Table 3 shows quantitative comparison for different trails and the official paper results

Model	FID ↓	IS ↑	Epochs
(My model) INT-CLS-GAN	421	NA	1000
(My model) INT-CLS-GAN with BERT	294	NA	311
(My model) INT-CLS-GAN with SkipGram	1122	NA	311
(My model) INT-CLS-GAN with CBOW	1173	NA	191
(My model) AttnGAN	258	2.43	100
(My model) INT-CLS-GAN (instance selection)	435	NA	630
(My model) INT-CLS-GAN (instance selection + extra layers)	537	NA	730
StackGAN [7]	NA	3.7	600
GAWWN [38]	NA	3.62	Unknown
AttnGAN [17]	NA	4.36	550

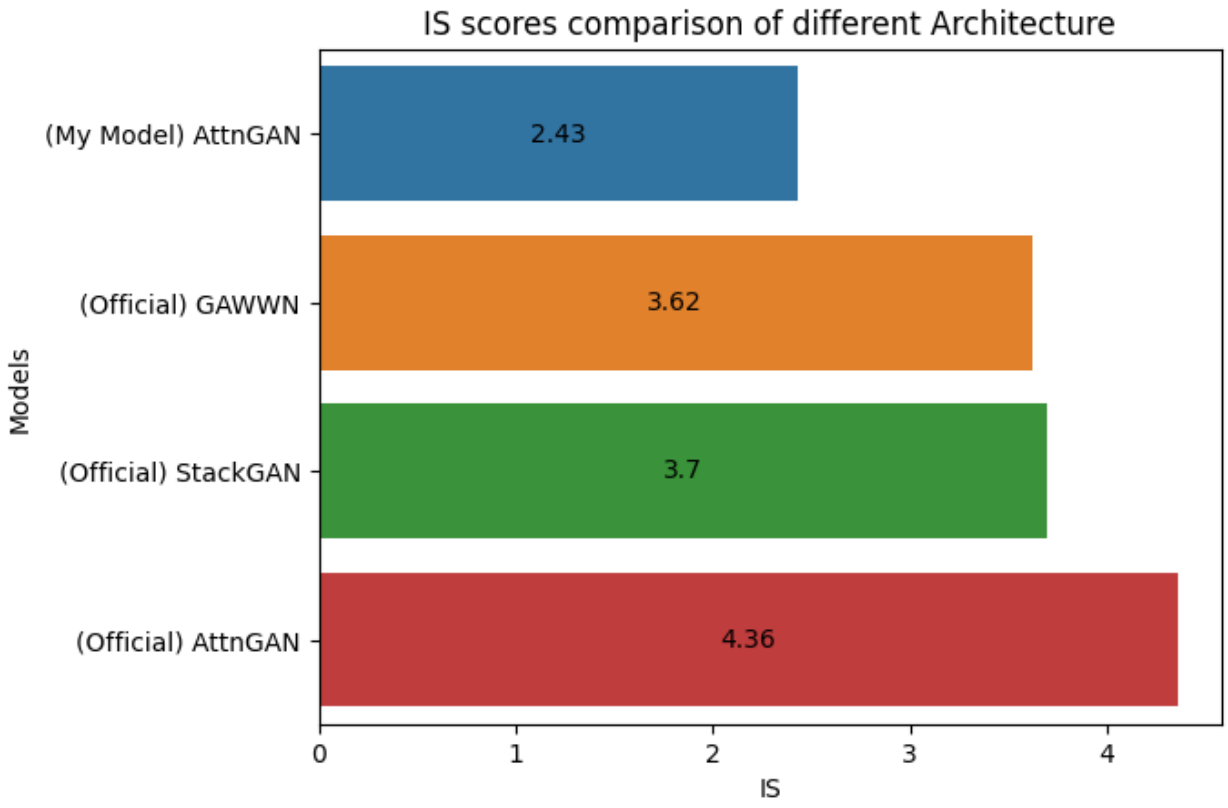

Figure 57 Visualized Comparison of Inception scores with literature

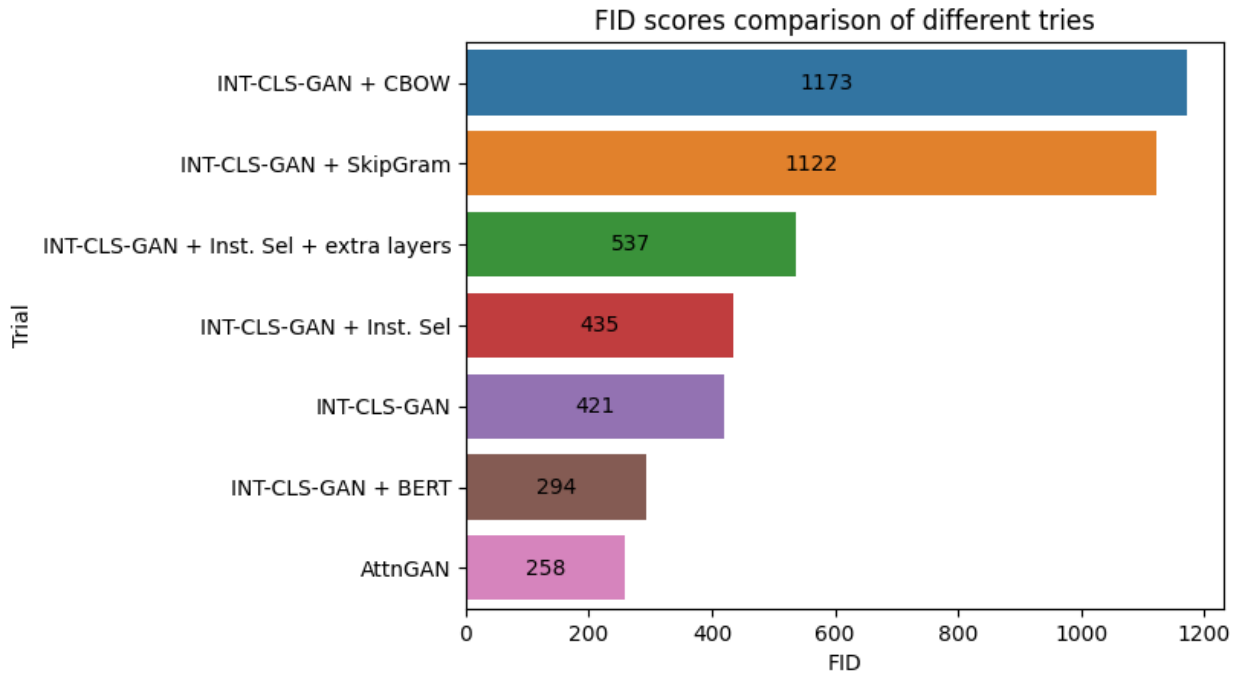

Figure 58 FID scores Comparison of our different trials

Notice that AttnGAN is implemented with PyTorch Framework and the other were implemented by Tensorflow Framework. *Thus, the Inception model can be slightly different.* Besides, these results are the mean average of 100 tests form the test set. The INT-CLS-GAN and its variants are upscaled by resizing to 75 * 75 as the TensorFlow allows this size as minimum size white the AttnGAN was tested on size 299*299 which, of course, impacts the results integrity as different frameworks and size has a great impact on the resulted score. Furthermore, it safer to compare INT-GAN-CLS and their variants quantitatively and compare the AttnGAN qualitatively. The scores from our trials have relatively lower scores than the published score due to the lowering of the batch sizes due to the limits of the hardware. Besides, fewer epochs as there were many trails each takes at least 2-3 days and cannot be done in parallel. Moreover, the testing on the *official trials is done on more samples (3000) while ours is 100 which gives better empirical results than ours.*

9 Future Work

Despite the rigorous development of Text-to-Image GANs from the late 2016 up until 2022. The generative adversarial network is currently in comparison with its new alternative which is diffusion model. Perhaps one of the boldest claims by researchers is claiming that a model is generally better than a model. However, it has already been partially done by Dhariwal et al [43] on the paper titled “Diffusion models beats GANs on Image Synthesis.”. This is claim can be verified by the results that makes diffusion models with very little work relative to the generative adversarial network it gave comparable and better results, however, it remains needs more experiments in generalized tasks to verify the claim. However, this improvement came on the expenses of the performance. The diffusion models require higher hardware powers in order to train. This makes diffusion model is less accessible for some users as it require either cloud providers or a high-end computers. There are some ways to compromise the competition between these to models which is mixing both to work together as in [44] which is training of GAN using the diffusion. Specifically, the text-to-image tasks seems to be going into the direction of the diffusion models as DALL-E outperformed GANs. Recently in June 2022, Google Released “Imagen” which is diffusion model that beats DALL-E and DALL-E2 which is diffusion models. The difference between DALL-E2 and Google Imagen publication dates is roughly a month. Thus, it can be clearly noticed that the direction of research in the text-to-image synthesis is shifting towards the diffusion models. However, research in generative adversarial networks can still potentially continue as it developed many techniques that are generally good that can be deployed in diffusion models. Such that the difference in performance required in deployment the GAN is still commercially better than the expensive diffusion. The main advantage of the diffusion over GANs is that the diffusion generalizes better. The GAN networks can only be better in some specialized datasets and achieve faster results. Such development in GANs in 2022 follows in semantic awareness as in [45] and deep fusion in [46]. The diffusion research remains small in number of papers. Yet, every paper is effective as it requires big teams and high funding which is predicted soon to be changed. The contribution in diffusion models is formally started by Nichol et al [47] in 2021 by introducing GLIDE for better text-to-image synthesis based on work from Ramesh et al [48]. The diffusion model starts to revolutionize from the work of OpenAI DALL-E2 [49] in June 2022 which trained on massive dataset scrapped form the internet. It yet has some black box parts as there are hidden vocabulary that the model was able to work with which is

studied in [50]. The last, yet most recent study on text-to-image, is Google Imagen [51] which currently has the highest state-of-art FID score of 7.27 which is dramatically higher than any GAN model. Noticeably, it was able to solve the challenge of writing text on the generated image which was not possible before. Comparison between Imagen (left) and GLIDE (right) on writing text description on the generated image. The target was to write “text-to-image” in the generated image as illustrated in figure 58.

Figure 59 Comparison between Imagen and DALL-E on writting text on the image

9 Conclusions & Recommendations

In Conclusion, the text-to-image synthesis is a promising research field that is incomplete. Which has a lot of missing areas and black boxes that to be solved in the future by the researchers such as the mix of the diffusion model and GAN model to have high performance model that is realistic and has high perceptual quality as in diffusion models and realistic and light weight as the GAN generator that is relatively lightweight. This research shows different tries on the training of GAN in order to generate a high-quality image. We have demonstrably proved that generating a 64×64 image remains the most important steps for adversarial hierarchical structures. We suggest researching a "fix" that can handle the problem of the failed samples of image of smallest scale. However, in order to generate a high-quality image is to stick to the hierarchical paradigm of training a generative network as the deeper the network is the lesser likely is to train stably. Besides, debugging a nested hierarchical structure is relatively easier than fixing and entire model as each result can be observed and studied separately and independently from the other models. Furthermore, the training can also run faster as the discriminator models can train in parallel and feed the generator back for improving their gradient and search for the minimum.

9.2 Project Contributions

We advise to work on making a discriminator having better feedback mechanism that can give generator a good to where it should move its probability distribution for the generation of the images. The discriminator should be multi-purpose as it should give feedback for realism and for text-image relevance. We concluded that the instance selection opens the door for more light-weight generative adversarial networks that runs as good as the original or, at least, it trains faster and converges in little time. However, this improvement gives the GAN model a good advantage over the diffusion. It is recommended to use instance selection for diffusion models to check how good it would be if the selected instances are in the denser data manifolds. The instance selection also opens the door for running a more complex architectures and gives higher architectural freedom as the data will be used be much lesser. However, it still needs to be used carefully as the retention rate λ can affect the results in different ways as it can cause the generator to lose the diversity for a better fidelity. We have showed that creating embedding from scratch for a small dataset, specifically, CUB birds is not a good idea as it showed the worst quantitative and qualitative results. The BERT results for generator where quite good but the trade-off performance

is not a good deal. The BERT pooled output has a very high dimensions and encoding its output reduces the quality of the image as shown in our experiments. The final recommendation is to care for the input of the GAN, embeddings and random Nosie, which highly impact the results. As it is shown by the research that the diffusion has upper hand and one of the main differences is that the input noise is statistically dependant on the original image and the real output. However, in GAN the case is different as the random noise can heavily alters the output so that it relies on “luck” which can be fine-tuned without hurting the diversity. The training speed was *improved by 54%* in the AttnGAN trial by changing lists to numpy arrays which enabled the training of higher number of epochs in the remaining available time.

Reference

[1] Kosslyn, Stephen M., Giorgio Ganis, and William L. Thompson. "Neural foundations of imagery." *Nature reviews neuroscience* 2.9 (2001): 635-642.

[2] Barua, Sukarna, et al. "Quality evaluation of gans using cross local intrinsic dimensionality." *arXiv preprint arXiv:1905.00643* (2019).

[3] Reed, Scott, et al. "Generative adversarial text to image synthesis." *International Conference on Machine Learning*. PMLR, 2016.

[4] Zhou, Rui, Cong Jiang, and Qingyang Xu. "A survey on generative adversarial network-based text-to-image synthesis." *Neurocomputing* 451 (2021): 316-336.

[5] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).

[6] Odena, Augustus, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier gans." *International conference on machine learning*. PMLR, 2017.

[7] Zhang, Han, et al. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks." *Proceedings of the IEEE international conference on computer vision*. 2017.

[8] Souza, Douglas M., Jônatas Wehrmann, and Duncan D. Ruiz. "Efficient Neural Architecture for Text-to-Image Synthesis." *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020.

[9] Wang, Tianren, Teng Zhang, and Brian Lovell. "Faces à la Carte: Text-to-Face Generation via Attribute Disentanglement." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021.

[10] Pavllo, Dario, Aurelien Lucchi, and Thomas Hofmann. "Controlling style and semantics in weakly-supervised image generation." *European Conference on Computer Vision*. Springer, Cham, 2020.

[11] Zhang, Han, et al. "Stackgan++: Realistic image synthesis with stacked generative adversarial networks." *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018): 1947-1962.

[12] Zhang, Zizhao, Yuanpu Xie, and Lin Yang. "Photographic text-to-image synthesis with a hierarchically-nested adversarial network." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[13] Sood, Ekta, et al. "Improving natural language processing tasks with human gaze-guided neural attention." *arXiv preprint arXiv:2010.07891* (2020). [10] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International conference on machine learning*. PMLR, 2015.

[14] You, Quanzeng, et al. "Image captioning with semantic attention." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[15] Wu, Yonghui, et al. "Google's neural machine translation system: Bridging the gap between human and machine translation." *arXiv preprint arXiv:1609.08144* (2016).

[16] Zhao, Bo, et al. "Diversified visual attention networks for fine-grained object classification." *IEEE Transactions on Multimedia* 19.6 (2017): 1245-1256.

[17] Xu, Tao, et al. "Attngan: Fine-grained text to image generation with attentional generative adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[18] Tan, Hongchen, et al. "Semantics-enhanced adversarial nets for text-to-image synthesis." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

[19] Li, Bowen, et al. "Controllable text-to-image generation." *arXiv preprint arXiv:1909.07083* (2019).

[20] Qiao, Tingting, et al. "Mirrorgan: Learning text-to-image generation by redescription." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[21] Almahairi, Amjad, et al. "Augmented cyclegan: Learning many-to-many mappings from unpaired data." *International Conference on Machine Learning*. PMLR, 2018.

[22] Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *International Conference on Machine Learning*. PMLR, 2017.

[23] Zili Yi, Hao (Richard) Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In ICCV, 2017.

[24] Qiao, Tingting, et al. "Mirrorgan: Learning text-to-image generation by redescription." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[25] Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[26] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[27] Lao, Qicheng, et al. "Dual adversarial inference for text-to-image synthesis." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

[28] Cheng, Yu, et al. "Sequential attention GAN for interactive image editing." *Proceedings of the 28th ACM International Conference on Multimedia*. 2020.

[29] Barratt, Shane, and Rishi Sharma. "A note on the inception score." *arXiv preprint arXiv:1801.01973* (2018).

[30] Heusel, Martin, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." *Advances in neural information processing systems* 30 (2017).

[31] Bynagari, Naresh Babu. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium." *Asian Journal of Applied Science and Engineering* 8 (2019): 25-34.

[32] Bińkowski, Mikołaj, et al. "Demystifying mmd gans." *arXiv preprint arXiv:1801.01401* (2018).

[33] Hinz, Tobias, Stefan Heinrich, and Stefan Wermter. "Semantic object accuracy for generative text-to-image synthesis." *arXiv preprint arXiv:1910.13321* (2019).

[34] Frolov, Stanislav, et al. "Adversarial text-to-image synthesis: A review." *arXiv preprint arXiv:2101.09983* (2021).

[35] Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset.

[36] Xia, X., Xu, C., & Nan, B. (2017, June). Inception-v3 for flower classification. In 2017 2nd International Conference on Image, Vision and Computing (ICIVC) (pp. 783-787). IEEE.

[37] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[38] Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., & Lee, H. (2016). Learning what and where to draw. Advances in neural information processing systems, 29.

[39] DeVries, T., Drozdzal, M., & Taylor, G. W. (2020). Instance selection for gans. Advances in Neural Information Processing Systems, 33, 13285-13296.

[40] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

[41] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

[42] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29.

[43] Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems, 34, 8780-8794.

[44] Wang, Z., Zheng, H., He, P., Chen, W., & Zhou, M. (2022). Diffusion-GAN: Training GANs with Diffusion. arXiv preprint arXiv:2206.02262.

[45] Liao, W., Hu, K., Yang, M. Y., & Rosenhahn, B. (2022). Text to image generation with semantic-spatial aware GAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18187-18196).

[46] Tao, M., Tang, H., Wu, F., Jing, X. Y., Bao, B. K., & Xu, C. (2022). DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16515-16525).

[47] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., ... & Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741.

[48] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... & Sutskever, I. (2021, July). Zero-shot text-to-image generation. In International Conference on Machine Learning (pp. 8821-8831). PMLR.

[49] Lee, S. DALLE-2.

[50] Daras, G., & Dimakis, A. G. (2022). Discovering the Hidden Vocabulary of DALLE-2. arXiv preprint arXiv:2206.00169.

[51] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., ... & Norouzi, M. (2022). Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. arXiv preprint arXiv:2205.11487.

Appendix I

The first generator does not have attention model as the attention model has two input the image features and the text features. The first model creates the initial images based on the text features only. The following generator gets the image and text features to use it on the next generation phase. Formally,

$$h_0 = F_0\big(z, F^{ca}(\bar{e})\big);$$
$$h_i = F_i\Big(h_{i-1}, F_i^{attn}(e, h_{i-1})\Big)\ for\ i = 1,2, \dots m;$$
$$\hat{x} = G_i(h_i)$$

Where h_0 the first hidden state, h_i is the i^{th} hidden state. The $\bar{e}$ refers to the global sentence vector while e refers to he matrix of word vectors. The F^{ca} refers to the conditional augmentation (more details in stackGAN paper [11]), F_i^{attn} is the i^{th} attention layer proposed in the attnGAN paper [17], F_i refers for a decoder network, and $\hat{x}$ refers to the output image scale.