Spatial-Channel Aware Generative Adversarial Network for Text to Image Translation

By: Mengistu Abebe Legesse

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2021

Adama, Ethiopia

Title: Spatial-Channel Aware Generative Adversarial Network for Text to Image Translation

By: Mengistu Abebe Legesse

Advisor

Prof Yun Koo Chung (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical and Computing

Presented in partial fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September , 2021

Adama, Ethiopia

DECLARATION

I hereby declare that this Master's Thesis entitled "**Spatial-Channel Aware Generative Adversarial Networks for Text to Image Translation"** is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Mengistu Abebe Legesse

Name Signature Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Spatial-Channel Aware Generative Adversarial Networks for Text to Image Translation**" prepared under my guidance by Mengistu Abebe Legesse submitted in partial fulfillment of the requirements for the degree of Master of Science computer science and engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Prof Yun Koo Chung (Ph.D.)

Major Advisor Signature Date

_____ _____ _____

Co-Advisor Signature Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled "**Spatial-Channel Aware Generative Adversarial Networks for Text to Image Translation**" and developed by Mengistu Abebe Legesse, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Major Supervisor	Signature	Date
Co-Supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Mengistu Abebe have read and evaluated the thesis entitled "**Spatial-Channel Aware Generative Adversarial Networks for Text to Image Translation"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in computer science and engineering.

Chair Person	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Chair Person	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

DEDICATED
TO
MY MOTHER ALEMAZE ASEFA
and
MY FATHER ABEBE LEGESSE

ACKNOWLEDGMENT

To start with, I would offer my actual thanks to Almighty God, who has conceded incalculable endowments, knowledge, and opportunity to me. Besides, I might want to thanks the Virgin Mary the Mother of God.

I would like to express my deepest gratitude to Prof. Yun Koo Chung (Ph.D.) head of the Computer Vision and Robotics SIG, for his outstanding leadership, support, encouragement, and guidance. I would offer my actual thanks to Dr. Worku Jifara (Ph.D.) for his outstanding leadership, support, encouragement, and guidance. The deepest gratitude for Dr. Mesfin Abebe (Ph.D.) for his worth comments. The deepest gratitude for Anteneh Tilaye (M.Sc.) for his worth comments, guidance, and kind support in evaluating the outputs produced. I would offer my actual thanks for menilk sahilu (M.Sc.) for his worth comments, and kind support in evaluating the outputs produced I'm additionally appreciative for all the help and exhortation I have gotten from computer vision postgraduate understudies, and companions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF CODE SNIPPET

LIST OF ABBREVIATIONS AND ACRONYMS

AFF	Affine Transformation
AI	Artificial Intelligent
AlexNet	Alex Network
ANN	Artificial Neural Network
AttnGAN	Attentional Generative Adversarial Network
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bi-directional Long Short-Term Memory
BN	Batch Normalization
CBN	Conditional Batch Normalization
CBOW	Continuous Bag of Words Model
CG	Cycle Generative Adversarial Network
CNN	Convolutional Neural Network
CNN-RNN	Convolutional Neural Network-Recurrent Neural Network
CUB-200	Caltech-UCSD Birds -200
CV	Computer Vision
DAMSM	Deep Attentional Multimodal Similarity Model
DBM	Deep Boltzmann Machine
DCGAN	Deep convolutional Generative Adversarial Network
DF	Deep Fusion
DF+B	Deep Fusion plus Bidirectional Encoder Representations from Transformers
DF+C	Deep fusion plus caption loss
DF+SA	Deep Fusion plus spatial attention in all block
DF+SA+CA	Deep Fusion plus spatial plus channel-wise attention in all block
DF+SA+CA+D	Deep Fusion plus spatial plus channel-wise attention plus Deep Attentional Multimodal Similarity Model loss in all block
DF+SA+D	Deep Fusion plus spatial attention plus Deep Attentional Multimodal Similarity Model Loss
DFBlocks	Deep Fusion Blocks

DFGAN	Deep Fusion Generative Adversarial Network
DNN	Deep Neural Network
FC	Fully Connected
FID	Fréchet Inception Distance
GAN	Generative Adversarial Networks
GLAM	Global-Local Collaborative Attentive Module for Cascaded image generation
GLOVE	Global Vector
GoogleNet	Google Network
GPU	Graphics Processing Unit
IDEs	Integrated Development Environment
IS	Inception Score
KL	Kullback–Leibler
LDA	Linear Discriminant Analysis
LeNet	LeCun Network
LSTM	Long Short-Term Memory
MA-GP	Match Aware- Gradient Penalty
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron
Ms-COCO	Microsoft Common Object in Context
NLP	Natural Language Processing
PixelRNN	Pixel Recurrent Neural Network
Q1	Question one
Q2	Question two
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
RNN	Recurrent Neural Network
SDGAN	Semantics Disentangling Generative Adversarial Network
StackGAN	Stack Generative Adversarial Network

STEM	Semantic Text Embedding Module
STREAM	Semantic Text Regeneration and Alignment Module
SVM	Support Vector Machines
UK	United Kingdom
UML	Unified Modeling Language
VAE	Variational Autoencoder
VGGNet	Visual Geometry Group Network
Word2Vec	Word to Vector

LIST OF NOTATIONS AND SYMBOLS

Notation	*Meaning*
X	Indicates the given Text domain
x_t	Indicate the List of words
Y	Indicates the image domain
y_g	Indicate the target image
G_{XY}	Indicates the generator make from X domain to Y domain
Y_s^{gen}	The generated image from a given text domain
$\approx$	Indicate approximate to
T_G	Indicate the ground truth image
X_o	Indicates the original text description
X_t	Indicates the reconstructed text
e_1	Indicate the error of the first generator
e_2	Indicates the error of the second generator
e_3	Indicates the error of the third generator
$\sum$	Indicates the summation
exp	Indicates the exponential
Z	Noise data sample
γ	Scaling parameters
β	Shifting parameters
w	Indicates Word vectors
S	Indicates sentence vectors
V	Image features
C_j	Indicates the word context vectors
m^n	Indicates the attention matrix at n stage
L	Indicates the number of words
$\alpha_{i,j}^n$	Indicates the attention wight at i words, j sub-regions

ABSTRACT

Generative Adversarial Network is one of the emerging deep learning technologies that is used for many image generations techniques. From this application text to image translation is one of them. The domain difference between the text and the image is the main challenging task. To mitigate this problem different state of art methods was employed, but those approaches have limitations because almost all the current text to image approaches employ a stack of multiple generators and discriminators, this leads to the training process getting slow and the desired output image might not be synthesized well if the first image isn't effectively synthesized, subsequent generators will struggle to improve it to suitable quality of the image. Another methodology utilized one set of generators and discriminators to alleviate the previous issues but this approach generates an image by stacking the entire text description in a single sentence vector, but this lacks the word level information relevance of the text description. still, the quality of the image and the semantic consistency is an open task in the text-to-image translation. To address those issues, this study proposed spatial and channel-wise attention mechanisms to enhance the quality of the generated image. and this study proposed additional constraints deep attentional multimodal similarity model loss to improve the semantic coherence between the synthesized image and the written description at word-level and sentence level. Spectral normalization was introduced to stabilize the training procedure and mitigate the modal collapse problems. The extensive qualitative and quantitative results indicate the effectiveness of the proposed model. Based on the inception score analysis the proposed model improves the baseline work CycleGAN from 3.70±0.045 to 4.31±0.14 and also improves the baseline work of deep fusion generative adversarial Network (DFGAN) from 3.45±0.065 to 4.31±0.14. The average human evaluation of the proposed model improves the DFGAN by 46.66 percent. This thesis concludes that spatial and channel-wise attention mechanisms have a great influence on the generation of quality images and DAMSM loss also has a great improvement in semantic consistency between the text and image information.

Keywords: GAN, CycleGAN, DFGAN, spatial attention, channel-wise attention, Deep Attentional Multimodal Similarity Model, spectral normalization

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

In recently, natural language processing and computer vision have become popular fields of research with the emergence of deep learning. Computer vision has been used to tackle a variety of issues in a variety of fields. Object detection, object recognition, classification, recognition, and domain transfer are just a few of the current study areas in computer vision. The major emphasis of this study is domain transfer especially on text to image translation.

The image caption is a frequent and tough problem in natural language processing and computer vision: The process of creating a brief description of an input photo or image is known as image captioning. The converse issue is a text-to-image translation: When a text description is given, a picture must be created that corresponds to this description(Bodnar, 2018). In computer vision (CV) and natural language processing, creating a high-quality realistic image based on provided text descriptions has become an appealing and hard challenge)(M. Zhu et al., 2019)(Cai et al., 2019)(Xu et al., 2017a) (Reed et al., 2016). Therefore this area attracted the attention and research of many scientists (L. Li et al., 2020).

The generating of images from natural language may have an expansive application of utilizations in the future when the innovation is commercially available. Individuals may construct customized furniture for their homes simply by explaining it to a computer rather than investing lots of time and energy looking for the right design. It also has tremendous potential applications such early childhood education, content production, and advertisement design, for example, might all benefit from greater collaboration between content creators and machines using natural language, and also art development, computer-aided design, photo-editing, and video gaming (Cheng et al., 2020)(Zhang et al., 2019)(Zakraoui et al., 2019)(Qiao et al., 2019a)(El-Nouby et al., 2019)(Sharma et al., 2018)(Xu et al., n.d.).

In recent years, the performance of caption generation has improved significantly due to the introduction of recurrent neural networks (RNNs). Meanwhile, the text-to-image translation from the existing method provides a rough sketch of the desired image but does not capture the true

essence of what the text describes (Gorti & Ma, 2018)(Dong et al., 2017)(Zia et al., 2020). Recent innovations of Generative Adversarial Networks (GAN) have become a highly popular approach for producing realistic data. Generative Adversarial Networks, presented by Ian Goodfellow (Goodfellow et al., 2020), are a kind of neural network that utilizes several neural networks and pits them against each other to produce a rendered image with extremely high accuracy and the output that is almost unclear from the original. The GAN model architecture includes two distinct networks termed generator and discriminator, both of which are based on the two-person zero-sum game theory. The generator's goal is to synthesized fake content by taking feedback from a discriminator and a discriminator that determines whether the created content is real or not (Jain & Jayaswal, 2020). Effective training of a GAN requires arriving at a balanced state between two restricting objectives.

In 2018, several artificial intelligence headlines screamed about the marketplace in the United Kingdom, the seller christie's has offered its first computer-rendered artwork, entitled Portrait of Edmond Belamy as shown in figure 1.1. The piece, made by a French art collective referred to as apparent, offered for a whopping $ 432,500. Some artists use computer-generated artwork, many others value it and are ready to pay for it (*Christie's Just Sold Its First Piece of Art Generated by AI. - Vox*, n.d.).

Figure 1. 1. A Generative adversarial network generated the portrait of Edmond Belamy.

Source: Taken from (*Christie's Just Sold Its First Piece of Art Generated by AI.* - *Vox*, n.d.)

The challenges of text-to-image translation can be subdivided into two manageable sub-problems: the first is embedding the text into a feature representation that preserves the image's important visual data, the second is generating a high-quality realistic image based on text feature representation. The translation is difficult because of the domain difference between text and visual feature representation.

Text to image translation: Recent researchers have done much research on the text to image translation. (Reed et al., 2016) introduced Conditional GANs. It only generated 64x64 images based on text descriptions, but the generated image usually lacking details and intensely deep object parts, for example, birds' beaks and eyes. In addition, Conditional GANs could not produce higher resolution images (128x128) without providing additional annotations of objects. (Zhang et al., 2017) used two multiple stacks of GAN. the author divides the problem down into subordinate problems. The first was stage would produce images with a smaller dimension (poor resolution) of the target image, and less accuracy. The subsequent stage was utilized to create a realistic high-resolution image utilizing the consequence of the principal stage as input. The original textual description of the image filling in the details of the sketch, but it still has a limitation in producing realistic high-resolution images and semantic consistency problems.

Another architecture was AttnGAN (Xu et al., 2017b), AttnGAN proposed attention methods that used to pay attention to relevant visual information to synthesize fine-grained features in distinct sub-regions of the picture. It has certain drawbacks, for example, the created image lacks text-image semantic coherence, and the synthesized image duplicates a portion of the image. (Qiao et al., 2019b), To enhance the linguistic coherence between the textual and visual data, MirrorGAN presented the concept for image captions for generating written information from the created picture. This model achieves a remarkable outcome when contrasted with the previous methods but still, it has a limitation as it only uses the fundamental technique for textual embedding and the training is not jointly end-to-end. (Hinz et al., 2019a), introduce SD-GAN that used Siamese structure helps to generated semantic consistent images from the written information.

Attention mechanisms: In current years, more than a few kinds of attention mechanisms have been utilized to create text-to-image translation. (Xu et al., 2017b) to find the relationship between the output image's subarea and the words in the input written information, a word-level spatial

attention mechanism was used. (B. Li et al., 2019) introduced a channel-wise attention mechanism at the level of words that considers both spatial and channel information. In addition to a spatial attention mechanism at the level of words.

Most of the above text-to-image generation approach employs multiple stacks of modules, which provides a significant increase in the development of high-quality images, but the individual method has its set of issues. Firstly, training a massive number of networks takes longer than training a single model, and this influences the generative model's convergence and stability. Second, the initial image has a big influence on the final polished result; when the first image is poorly synthesized, subsequent generators will struggle to refine it to suitable image quality.

In addition, certain methods perform better results as compared to the previous one by using one generator and discriminator, these approaches synthesizing an image from the sentence-level feature. Sentence level vector stores the whole information in a single vector, this results in a loss of crucial fine-grained details information at the level of words, which prevents the creation of high-quality images and maintaining the semantic coherence among the fake picture and the textual information. So, generating high-quality images and enhancing the semantic coherence among the created image and textual information is an open task that needs improvement. As a result, text to image translation is still a hot topic in academia. Finally, this paper proposed Spatial-Channel Aware Generative Adversarial network which introduces constraints like spatial attention, channel-wise attention, and Deep attentional multimodal similarity model (DAMSM) loss to solve the above problems, and certain constraints are identified to enhance and preserve the content of the generated image.

1.2. Motivation of the Study

As deep learning got more popular, the need for enormous amounts of data increased as well. Deep learning works best when it has a lot of high-quality data to work with, and as the amount of data available grows, so does its performance. One of the technologies of deep learning is GANs. GANs were introduced in 2014, and authors opened up a lot of doors. The utilization of GANs has helped an assortment of uses, including medical image translation, image-to-image translation, text-to-image translation, and others. Fashion, art creation, graphic design, and cinematic effects, photo-editing, and video games, and advertisements may all benefit from this network's growth.

The task of creating high-quality images from textual descriptions is an ongoing and challenging scientific topic. Recent methods have a semantic gap between the synthetic picture and the text description. In addition, certain methods don't preserve the content of the synthetic image and generate lower-quality images. This dilemma inspires us to work on this area. Most academics are interested in using GAN to synthesize high-quality pictures based on written descriptions since it is a powerful network and the created image is sharper than the other deeper generative models. As a result, it's worth looking at text-to-image synthesis algorithms, particularly GAN-based methods, as well as applying some advanced techniques to text-to-image generation. This research tries to address the challenges by building on prior research.

1.3. Statement of the Problem

The task of creating high-quality images from textual information is an ongoing and difficult scientific topic. The synthesized image should reflect the textual description as well as being of acceptable quality. The issue of text to image translation can be broken down into reasonable sub-issues: the first is embedding the text into a feature representation that preserves the image's essential visual data, the second is generating a realistic image based on feature representation of the text. The difference between the text domain and the image domain is quite difficult to translate (Tsue et al., 2020).

1.3.1. Notation Overview

In-text to image translation, having the text and image collection: $X = \{x\}_{x=1}^{T}$ in the source domain and $Y = \{y\}_{g=1}^{T}$ in target domain where X denotes the text domain and Y denotes the image domain and x is the text description, y is the target image. The objective of this task is to learn two mapping functions between source domain X and target domain Y, i.e., $G_{XY} : X \rightarrow Y$. Here the mapping functions G_{XY} are implemented as generators for synthesizing an image.

1.3.2. Problem Formulation

Case 1: The majority of the past works use stacking multiple GANs. In this scenario, the first image has a significant impact on the refined output; when the first image is poorly synthesized, subsequent generators will struggle to improve it to acceptable image quality. As a result, it has an impact on the generative model's convergence and stability, as well as the quality of synthesized images (Tsue et al., 2020)(Xu et al., n.d.)(Zhang et al., 2019)(Gorti & Ma, 2018)(Qiao et al., 2019b)(Qiao et al., 2019a).

Let consider three stacks of generators, each generator generates a different scale of image $G = \{G_1,\ G2, G_3\}$.

Where G_1, $G_{2,}G_3$ denote the first, the second, and the third generators respectively.

$$e(G_1) = e_1 \quad e(G_2) = e_1 * e_2, \quad e(G_3) = e_1 * e_2 * e_3 \qquad eq.(1.1)$$

where e_1 ,e_2 , e_3 indicates the errors in $G_1, G_{2,}G_3$ respectively. Therefore, the error will be amplified at the final stage of the generator. Therefore, this leads to generating low-quality images and may lack semantic coherence between the text information and the synthesized image.

Case 2: Other methods use one generator and discriminator, these approaches synthesizing an image from the sentence-level feature. Sentence level vector stores the whole information in a single vector, this leads to a lack the crucial fine-grained details information at the level of words, which causes high-quality image from being generated and maintaining the sematic consistency between the fake image and the textual information.

Therefore, this work formulates the text-to-image translation model to generate high-quality images and to keep the semantic consistency of the textual information and the synthesized images.

Let $Y_S^{gen} = G_{XY}(x)$ where the Y_S^{gen} indicates the generated image from a given text description. This work going to minimize the semantic difference between the generated images and the text description.

$min = \left|\left|Y_S^{gen} - X_t\right|\right| \approx minimize\ the\ semantic\ gap$ and $maximize\ the\ quality\ of\ Y_S^{gen}$, where Y_S^{gen} is the synthesized image, and X_t is the text description. The proposed work will solve those the above problems.

1.4. Research Questions

The following research questions will be attempted to be addressed in this work.

RQ1. What are the effective constraints to improve the text-to-image translation?

RQ2. What will be the effect of attention mechanisms regarding improving the quality of the generated image?

RQ3. What will be the effect of deep attentional multimodal similarity model loss regarding improving the text-image semantic consistency?

1.5. Objective

1.5.1. General Objective

The general objective of the study is to design and implement text-to-image translation using Spatial-Channel Aware Generative Adversarial Networks.

1.5.2. Specific Objective

The following specific objectives are addressed to achieve the general objective.

- To identify learning constraints for improving the generated image or text-to-image translation.
- To design the generator and discriminator network by adding certain constraints.
- To embed attention mechanisms to DFGAN.
- To embed deep attentional multimodal similarity model loss to DFGAN.
- To evaluate the performance of the model.

1.6. Scope and Limitations

1.6.1. Scope of the Study

Within the time and resources available, the scope of the thesis study comprises the following:

- Translate a given text domain to an image domain
- Add learning constraints to the Generative Adversarial model
- Embed attention mechanisms to generator model to enhancing the quality of image in a text to image translation.
- Embed deep attentional multimodal similarity model loss to generator model to improve the semantic coherence between a generated image and textual information.
- Adding spectral normalization for stable training and mitigating the modal collapse problems.

1.6.2. Limitation of the Study

- The system doesn't consider the background of the image during the generation.
- One-to-many translations from text to the image are not considered.

- This work only focuses on synthesizing an image, it doesn't consider video synthesizing

1.7. Contribution and Beneficiaries of the Study

1.7.1. Contribution of the Study

The contribution of this study is listed as follows

- Enhancing the quality of the synthesized image by bringing consideration the attention mechanism model into the network, and giving more attention to fine-grained data at the word level to enrich the details of the generated image.
- Improve the semantic consistency of the synthesized images and the corresponding text descriptions by introducing the deep attentional multimodal similarity model loss (DAMSM loss).
- Spectral normalization has been introduced to mitigate the modal collapse problems and for stable training of the discriminator.

1.7.2. Beneficiaries of the Study

The main benefits of this study can be used in a variety of areas including

- Art generation and computer Aid-design: - Instead of spending hours looking for the perfect design, people might make personalized furniture for their homes by just describing it to a computer.
- Early childhood education: - Images tell thousands of words than text so teaching children though image help the children to memorized things.
- Photo-editing and video games: - It's also used in video game applications to manage and create characters in the game, as well as to interact with other characters in the game based on text descriptions.
- Content production and advertisement design: - Anyone with no prior expertise with graphical design software may create a logo, banner, business card, or other design by describing what they want to create.

1.8. Organization of the Thesis

The remainder of this thesis is organized in the following way:

Chapter Two: Explain the background literature and related works regarding text encoding, attention mechanisms, and text to image translation. This chapter also explains the theoretical framework of machine learning, Deep Learning, and GANs.

Chapter Three: Explain research methodologies such as data collection methods, data preprocessing, design tools, development frameworks, and platforms, and evaluation methods are also discussed.

Chapter Four: Explain the proposed model architecture in detail. Model Architecture, text encoding, generator network, discriminator network, and pseudocode for implementation and models mathematical correspondent descriptions have been made.

Chapter Five: Explains how the desired proposed model is implemented. and the working environment setup, spatial-channel aware GANs implementation, with training implementation, described using snip code. Experimental classes are also discussed.

Chapter Six: describe the desired proposed solution findings, compare the new result with the prior solution, and explain the meaning of the findings.

Chapter Seven: Concludes the research and gives bearings to conceivable future work

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

Lately, the software was coded with guidelines and a bunch of rules addressing numerical formulas (For, 2005). Such techniques are uncommon because they attempt to demonstrate complex data and circumstances from this present reality. To address these issues, artificial intelligence has evolved, which attempts to intelligent machines in the same way that people and animals do. Subsequently, scholarly conversations about the brain and the morals of making counterfeit creatures with the human-like agreement have arisen.

Machine learning (ML) is a branch of artificial intelligence (AI) that aims to increase machine intelligence without explicitly scripting their actions. There are various types of ML among those. The first machine learning is supervised learning which every input data is a map to its output labels. Most algorithms that used this principle are support vector machines, random forests, and others. Another kind of ML is unsupervised learning, unlike the supervised approach, it organizations unsorted data according to similarities and patterns besides any prior training of data. Every input data doesn't have its correspondence output label. Semi-supervised learning is a type of ML that stands in between the two-machine learning. The model is executed with a huge sum of unlabeled information and a little sum of labeled information. One more classification of ML is reinforcement machine learning algorithms are a kind of learning strategy that the agent cooperates with its current environment through delivering actions and rewards. Its method of learning is trial and error. Machine learning is very important when designing a system, but it struggles to analyze huge amounts of data such as images and videos, therefore deep learning was developed to solve this problem.

Deep learning is a sort of ML that is propelled by the design and elements of the human mind. It implements a deep neural network (DNN) and it became well-known due to the ability to process a colossal amount of data and the increase of high-performance computing facilities. Deep learning algorithms have a lot of layers, and information flows through each one. Each layer can extract features and pass them on to the next. The first layers extract low features, and the next layers

combine the features to form a complete representation. Artificial Neural Networks (ANN) (ANN) (Bekesiene et al., 2021) were the first deep learning methods to emerge. ANN is a set of associated elements known as Artificial neurons, which are made up of perceptrons in neural layers.

Convolutional neural networks (CNNs), often known as ConvNets, were invented in the 1980s by Yann LeCun (Lecun et al., 1998), a postdoctoral computer scientist. They were mostly used for images since authors could recognize handwritten digits. Compared to other classification methods, it requires less pre-processing. There are several CNN models designs, including LeNet, AlexNet, VGGNet, GoogleNet, ResNet, and ZFNet, which are pre-trained models with a large data set, imageNet. Object detection, semantic segmentation, and captioning are some of the areas where CNNs are employed.

Generative models are the most remarkable new developing deep learning techniques. The Markov chain, maximum likelihood estimation (MLE), and approximation inference are all prominent techniques utilized in the advancement of generative models (Srivastava & Salakhutdinov, 2014)(Ngiam et al., 2011)(Salehi et al., 2020). GANs is a subtype of the generative model. GANs are interesting, what makes them interesting? GANs are a significant step forward in the area of Machine Learning, particularly for generative models.

Before diving into the details, let's get a general idea of what GANs are for. GANs were presented by (Goodfellow et al., 2020) GANs are part of the generative model family. It implies that they can create or adapt new content. Naturally, GANs appear to be a little bit "magic," at least at times, because of their potential to create new content.

Nowadays, the essential applications of GANs include such as image-to-image translation, contextual image editing, style transfer, image super-resolution, and classification, and many potential medical applications (Salehi et al., 2020). Text to picture translation is one more utilization of GANs. The previous applications translate contents within the same domain, but a text-to-image translation is somewhat different since it creates an image based on a written description, which is a difficult process due to the semantic difference of the two domains. let see the detailed explanation of GAN in the next section.

2.2. Generative Adversarial Network

Generative Adversarial Networks (GANs) are an appealing approach for simulating complicated real-world patterns by addressing a min-max optimization issue among the generator and discriminator. GANs contained two separate networks called generator and discriminator. GAN is based on a two-person zero-sum game in which one participant's earnings (or loses) are exactly equal to the other's losses (or profits).

The generator attempts to create fake information by learning the measurable dissemination of genuine information that is unclear from true information to misdirect the discriminator into thinking about these as genuine data sources. The generator taking input from the discriminator to learn more. Then, the discriminator decides if the created image looks genuine or fake.

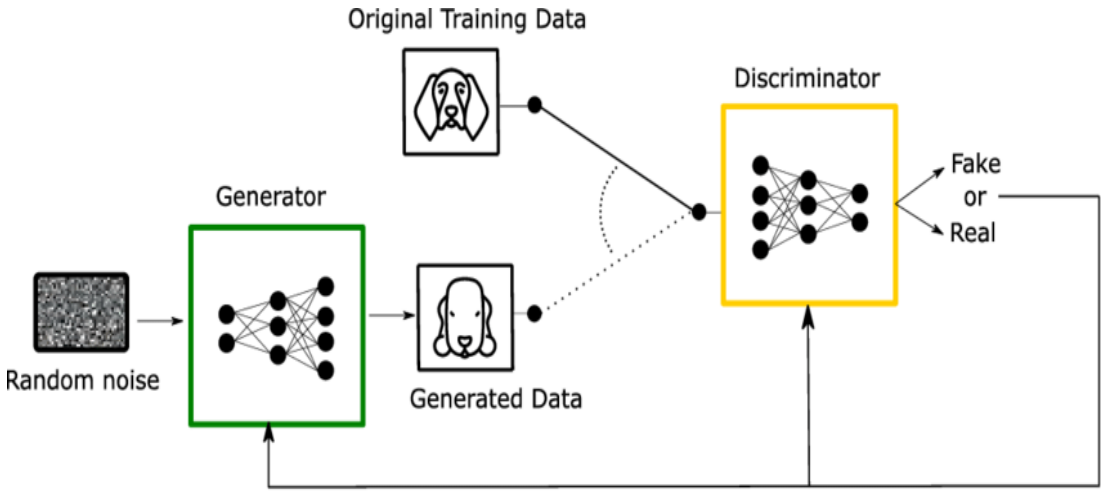


Figure 2. 1. Generator discriminator network architecture

Source: Adapted from (Barrios et al., 2019)

2.2.1. Internal Structure of GAN

GANs are a system for training generative models, the possibility of GANs is to cause two networks to go up against each other. GAN has two autonomous models the generator and the discriminator. A generator network intends to sort out how genuine data is scattered and a discriminator model assesses the probability of samples starting from training information data rather than the generator. The generator network create an image x' by accepts noise vector z as input, z samples from $p_z(Z)$ which is the standard normal distribution. The discriminator accepts two inputs the genuine picture and the synthesized image. where genuine pictures are samples

from real data distribution which is $p_{data}(x)$. A min-max target capacity of the generator and discriminator is used to prepare the two models simultaneously.

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}(x)}[\log \mathrm{D(X)}] + E_{Z\sim p_z(Z)}[\log(1 - \mathrm{D(G(z))})] \qquad eq.(2.1)$$

This urges the generator to fit $p_{data}(x)$ to trick the discriminator with the created test x'. Both of them are prepared by backpropagating the loss in $Eq.(2.1)$through their different models to refresh the parameters.

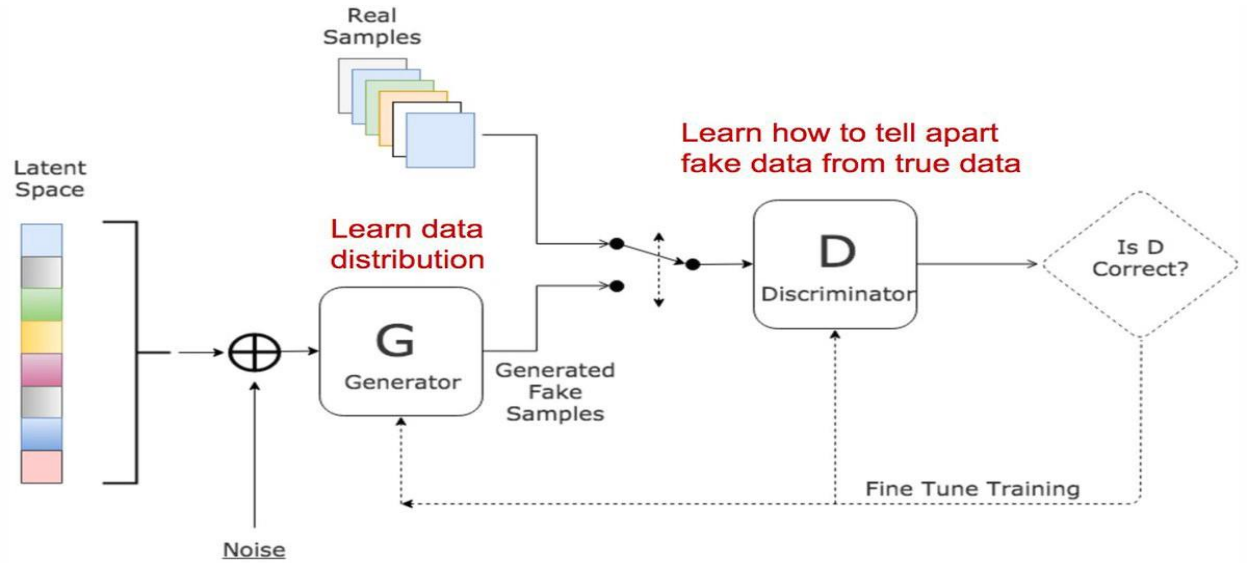


Figure 2. 2. Generator discriminator network architecture

Source: Adapted from (*A Review of Generative Adversarial Networks — Part 1 | by Roy Ganz | Analytics Vidhya | Medium*, n.d.)

2.2.2. GAN Training

Machine learning creates new models or uses current models to learn from the actual world to generate generalizable models that can make accurate predictions or identify patterns. GANs were utilized similarly to make another model for learning real-world issues

Both the generator and discriminator networks strive to outperform one another, and as a result, both improve. These two networks advance in terms of their different aims due to the competition between them. The objective is to decrease the loss constraint so that the model learns the distribution of the output probabilities for the input.

The most difficult question in GAN is how to ensure steady training of the generator-discriminator network. For both networks to learn at an identical time, there needs to be a wholesome competition between the generator and the discriminator. The loss function's parameters update fast since both are computed from the discriminator's output. The generator no longer receives enough gradient updates for its parameters as the discriminator converges quicker, and therefore fails to converge. Apart from being difficult to train, GANs can also suffer from partial or entire modal collapse, which occurs when the generator produces nearly identical outputs for various latent encodings. Now let's see overall the training of Generative Adversarial Networks.

1) Train the Generator:
    a) The generator network takes random noise as input and synthesizes a fake for instance x'
    b) To categorize x', employ the Discriminator network.
    c) To optimize the error of the discriminator, compute the classification error and backpropagate to modify the learnable parameters of the generator.s
2) Train the Discriminator:
    a) Genuine arbitrary sample x is taken as the training dataset input
    b) It accepts synthesize a fake example x' as input.
    d) To categorize x and x', employ the Discriminator network.
    c) Analyze errors in categorization and backpropagate the total error to modify the learnable parameters of the Discriminator, to reduce mistakes in classification.

2.3. Text Encoders

In machine learning, text encoding is the process of converting meaningful text (input sequence) into numerical features or vector representations while preserving the context and relationship between words and sentences, so that a machine can recognize patterns in any text and understand the context of sentences.

2.3.1. Text Tokenization

The principal phase of text encoding is text tokenization, which converts the input textual information into tokens. A tokenizer can be configured to filter out punctuation (or not), combine characters in lowercase or uppercase, and so on, depending on the job requirements. The text is then split into tokens.

2.3.2. Word Embeddings

The second stage is word embedding, which maps each token to its associated word embedding. Word embedding is a method for converting word sequences (tokens) into continuous vectors that can handle lexical and semantic characteristics. Word embeddings are more economical in computing and more effective in generalization because Word embeddings use dense and low-dimensional vectors rather than symbol representations and one-hot encoding, for example. As a result, the method is frequently grouped with deep learning.

Besides, numerous algorithms were proposed to find out word embeddings more efficiently among those. Word2Vec (Mikolov, 2014) proposed by Tomas Mikolov is a predication-based supervised model. It has two methods Skip Gram and Common Bag of Words (CBOW and both are used to obtain Word embedding (both involving Neural Networks). Global Vectors (GloVe) (Pennington et al., 2014) builds a global word-word co-occurrence network to catch the global insights utilizing grid factorization strategies and use the expectation-based technique in Word2Vec, occurring in generally better word embeddings

Aside from using these ways to deal with taking in word embeddings from scratch, their pre-prepared embeddings might be reused in other language-related errands in a static or updated way, which can further develop execution and shortening preparing time. Word embeddings for phrases, sentences, and paragraphs can be utilized to create text embeddings. Simple operations on vectors and matrices, such as unweighted averaging, can be utilized to do this. Recent works on pre-trained text encoding methods are discussed below.

Recurrent neural networks (RNN)(Sherstinsky, 2020) is recognized for the ability to represent variable input contextual relationship since it contains an internal state which can store and utilize the effects of past calculations for present operations. This enables it appropriate for sequential data processing. As a result, the layers slope approaches zero if the chains rule is applied and eliminates after several stages. Vanishing slopes, especially in networks designed to mimic a language, is a prevalent problem. To conquer the vanishing gradients and long-term dependency issues of conventional RNNs, long momentary memory (LSTM) was developed.

Long Short-Term Memory (LSTM) is indeed an RNN framework that can understand long-term relationships. First presented by (Hochreiter, 1997) and numerous additional studies in recent times became renowned (Graves, n.d.). To evaluate the internal behaviors, an LSTM network

consists of memory cells and gate units. Three gates govern the cell state function. These gates specify logically what kind of a certain vector is now to be examined at a certain interval t (Ahmadi, 2018). LSTM gives better memorization of the past state but to memorize the future state of the data bi-LSTM was introduced.

Bidirectional Long Short-Term Memory (Bi-LSTM)(Schuster & Paliwal, 1997) is introduced to overcome the limitations of a regular RNN described above, It is LSTM extender have to get the semanticized version of particular text information. Throughout the bidirectional LSTM layer, the semantical intended message is captured using two hidden states, and the final hidden state indicating the sentence characteristics is used (Schuster & Paliwal, 1997).

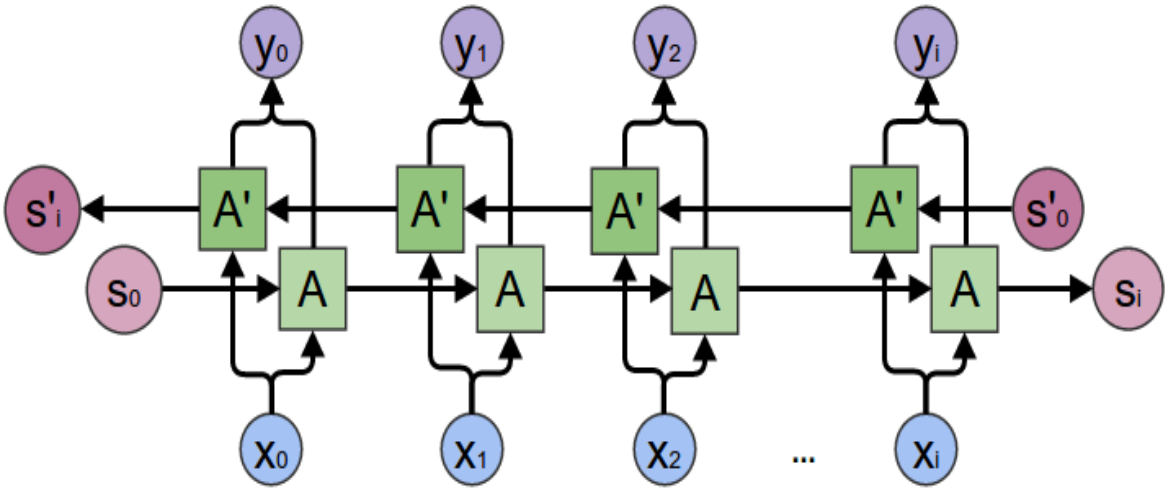


Figure 2. 3. Bidirectional Long Short-Term Memory (Bi-LSTM)

Source: Adapted from (*Bi-LSTM. What Is a Neural Network? Just like Our… | by Raghav Aggarwal | Medium*, n.d.)

Text encoder has a critical function in a text to image translation that is utilized to encode the text description of human-understandable format into a machine-understandable format. Encoding a text influence a generation of an image because if the text doesn't encode well using a better text encoder method, a semantical inconsistent and low-quality image might be generated. Other techniques are used for better quality generation of an image which is attention mechanisms. Next, let look at the attention mechanism explanation.

2.4. Attention Mechanism

Attention mechanisms have emerged from studies of human vision. In psychology, due to data processing bottlenecks, only a fraction of all obvious data is visible by a human. Rusted by this mechanism, scientists have sought to monitor the visually particular model of imitation of humans' perceptual process, to mimic and broaden the dispersion of human attention throughout the observation of pictures and videos (Yang, 2020).

Significant advancements in the domains of computer vision and natural language handling have been accomplished throughout past times. It was shown that processes of attention might function on models and also work mostly with the perception system of the mental and sight of humans (Yang, 2020).

The primary attention component was introduced by (Bahdanau et al., 2015) to deal with the issue of the sequence to sequence model (Sutskever et al., 2014) that can't deal with long input sequences well. From that point forward, various variations of attention mechanisms had been emerged and effectively used in machine interpretation, image focusing, and so forth. As of late, this idea has likewise been stretched out to the field of computer vision.

Attention mechanisms can collect task-related knowledge appropriately and decrease distraction from less crucial data. As of late various types of attention, mechanisms have been utilized for text to image translation from those, (Xu et al., n.d.) constructed, the AttnGAN model provides designed spatial reflection on the individual words to direct the generator into subgroups in comparison with the main word. However, spatial examination only concerns words that do not examine the channel data containing incomplete locals. (B. Li et al., 2019) presented, the reliability of the channel and spatial attention at individual words have been demonstrated to achieve a better quality picture while taking into consideration spatial and channel data.

Attention mechanisms have crucial importance for text-to-image translations that are used to give more attention to the relevant part in textual information during the generation of the image to have better image quality and synthesizing of a realistic image. Another technique is used for better quality generation of an image is image encoding-decoding. Next, let look at the image captioning explanation.

2.5. Image Caption

Image captioning is a course of producing textual information of the given image. Image captioning is the inverse method of the text to image translation which generates the text description for a given image. The encoder and decoder were the two sections of image captioning. The encoder for extracting the picture feature is usually the CNN model. The decoder accepts the output of the encoder as an input then it gives us the generated textual information of the image.

Recent methods are based on deep neural networks. Semantic Text Regeneration and Alignment Module (STREAM) module introduced by (Qiao et al., 2019b)(Tsue et al., 2020)(Gorti & Ma, 2018)(Monica & Rao, 2020) are one of the image captioning methods. The text information of the produced picture provided by STREAM could be regenerated and was used to concentrate semantically on the source text information. It employs the encoder-decoder principles. The generated image encoder by CNN (Monica & Rao, 2020)(Qiao et al., 2019b)(Gorti & Ma, 2018)(Tsue et al., 2020) was pre-trained on ImageNet, and then the features of the image decoder by an RNN.

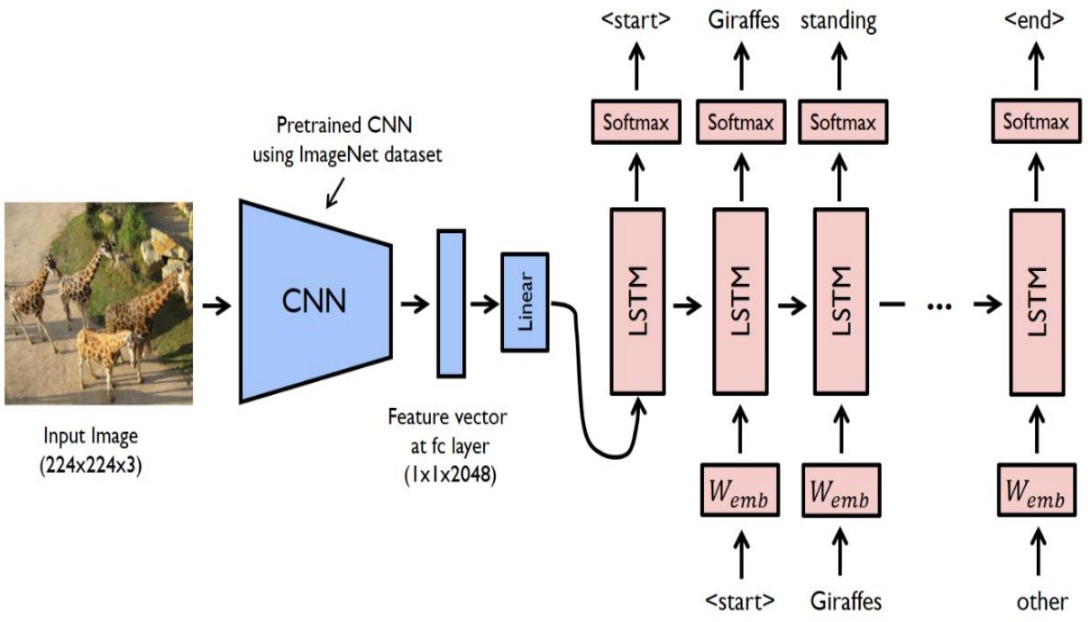


Figure 2. 4. Image Caption

Source: Adapted from (*Automatic Image Captioning Using Deep Learning*, n.d.)

2.6. Related Works

2.6.1. Traditional Machine Learning and Deep Learning

Various sorts of multiple information modalities can be utilized to depict a similar occasion. For instance, pictures, which are frequently addressed with pixels or picture descriptors, can likewise be portrayed with going with text (e.g., client tags or captions) (Sohn et al., n.d.).

Multimodal information has indeed been learned with different techniques. From those, (Huiskes et al., 2010) further, develop the grouping precision of Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA) models by utilizing captions (subtitle), or labels (tag), besides standard low-level picture, includes essentially.

This is a model that has been used (Srivastava & Salakhutdinov, 2014), for the study of multimodal expression in data. This model utilized a massive amount of unlabeled data. The model combines several kinds of data into a single form. It used the feature representations for classifications and retrieval. (Sohn et al., n.d.) a new multimodal paradigm for deep learning depending on a range of inputs. The minimum variation of information objective enables us to learn well-shared representations of multiple heterogeneous data modalities with a better prediction of missing input modality.

Another method is (Welling, n.d.), graphic probability issue models were developed by Variational Autoencoders (VAE) to enhancing the lower probability of information. (Van Den Oord et al., 2016) Autoregressive models (PixelRNN) that used neural organizations to display the contingent dissemination of the pixel space have likewise produced engaging generating images.

The bulk of the approaches mentioned above integrated an image were generated by scanning and supervised learning procedure (Y. Zhu et al., 2007). Approaches might utilize correlations between keywords and images to connect picture text information, which identifies instructive and "pictured" text units. The principal drawback of conventional text-to-image learning methods is that new picture content cannot be created; the properties of a picture can only be modified. Generating images from the textual description is an ongoing research area, text to image translation was an attractive and hot research area due to the emerge of the Generative Adversarial network. Next, let look at recent related work which is done through GAN.

2.6.2. Related Works

In the previous section, the transitional approaches were discussed for text to image generation. however, these approaches have a drawback as stated in the previous section. Now let see the related work of text to image generation using GAN.

GANs have recently demonstrated remarkable outcomes to produce high-end pictures. so, specialists utilize this generative Adversarial network from those work. (Reed et al., 2016) authors have been depicting the principal completely differentiable, start to end model that figures out how to develop images from text, expanding on conditional GANs. To encode the information of text, authors present a character-level convolutional-recurrent network. Conditional GAN has been only generating 64x64 images based on text descriptions, however, frequently lacks specifics and very rich object sections such as birds' eyes and beaks, Additionally, authors couldn't produce higher resolution (128x128) images without providing additional annotations of objects.

(Zhang et al., 2017) proposed StackGAN, the authors used two multiple stacks of GAN. They decomposed the problem into sub manageable problems. The first was Stage-I GAN would generate images of a lower dimension (poor-Resolution) of the target picture and lesser accuracy. The subsequent stage was utilized to create a realistic high-resolution image utilizing the consequence of the principal stage as input and source text of the picture, the drawing details can be completed out, but still, it has a limitation of generating realistic high-resolution images and sematic consistency problems.

Another architecture was AttnGAN developed by (Xu et al., n.d.) it was an attention-based model. AttnGAN acquaints a clever thought with text to image task which is consideration components that are utilized to focus on the crucial visual data to combine fine-grained words at various subregions of the picture. However, the previous models rely on sentence-level encodings. AttnGAN used the bidirectional LSTM as a language encoder to extract both sentence-level and word-level features from the input sentence description. Stack of multiple GANs has been used for generating high-quality images and authors used DAMSM to compute the text-picture loss for better semantic coherence of text-picture pair, but still, it has some limitations from those, the generated image lacks fine details in the quality of the information and there is also semantic consistency issue.

(Gorti & Ma, 2018) Authors used cycle consistent GAN to address the issue of generating a high-quality image from its corresponding text description and to reduce the distance between the genuine images and the created images. Authors divide the stage into sub-stages like a StackGAN architecture. The first stage produces a low-resolution image and it serves as input for the second stage. The second stage produces a 128x128 realistic high-resolution image by taking the output of the first stage as input and the original text description of the image to add more details to the synthesized images. Authors implement an image captioning GAN for reducing the difference between the real images and the synthesized images by comparing the original text and the regenerated text description. its achievement on the generation of quality images and minimize the gap between actual and synthesized pictures but it has problems in generating high-resolution images and also a gap between the real images and the synthesized image.

(Hinz et al., 2019a) MirrorGAN, Proposed a clever global-local collaborative attentive module to use both local word consideration and global sentence consideration and to improve the variety and to address the semantic consistency between text depiction and visual substance challenges. The MirrorGAN has three modules. First is the semantic text embedding module (STEM) which generate the word and sentence level embedding, second global-local collaborative attentive module (GLAM) is used for cascaded the image Generation, generation, third is the semantic text Regeneration and alignment module (STREAM) used to generate the text description from the generated image. Authors got striking outcomes when contrasted with the past techniques but still, authors have limitations authors only utilize the basic method for text-embedding in STREAM and authors don't train the network end-to-end.

(Tsue et al., 2020) Proposed cycle text to image translation with BERT, the authors used the combination of original CycleGan and the attention GAN (AttnGAN) together to synthesize a high-quality image and the authors used Bidirectional encoder representational text for text embedding, authors generate better result as compare to the other but still, it has the problem of semantic consistency and the difference between genuine images and the created images. And also, the AttnGAN limitation is reflected in the consequence of the generated image.

(Tao et al., 2020) proposed deep fusion Generative Adversarial network, the authors were introduced one generator and discriminator to the synthesized high-quality image. This paper introduces a deep fusion block to merge the written information and the image information during

the image creation stage. Match-aware zero centered gradient penalty is introduced with keep up with to balance out the training of GAN. This DFGAN generate remarkable image but DFGAN encodes the whole text description as a single vector as a condition for image generation, but this kind of trend lack fine-grained word-level information. All those above approaches achieved a remarkable result but due to the nature of GAN most of those methods facing some challenges. Next, Let's discuss the challenge of GAN.

2.7. Discussion on the Challenges

The difficulties of this work are gotten from innate issues of GANs, requests for producing high-quality images. GANs' learning strategy is characterized to be unpredictable and frequently prompts 'mode collapse. Mode collapse is one challenge of GAN that the generator figures out how to make tests from a couple of methods of conveyance. Wide procedures were created to alleviate the training instability and further develop the sample variety by planning new structures with new learning destinations, utilizing regularization strategies. For instance, Spectral normalization was utilized. The subsequent problem was vanishing gradients, to moderate these issues a few methods have been utilized to match aware gradient penalty (MA-GP), which upgrades the convergence of the generator. The last challenge was the performance of our GPU resource problems the performance of our GPU is low as compared to others, this leads us to modify different things in our model.

2.8. Summary of Related Work

Table 2. 1. a summary of previous works on text to image translation

Related Papers	Datasets (Data Collection)	Architecture	Text encoder model	Evaluation Matrix's	Limitations
(Reed et al., 2016) Generative Adversarial Text to Image Synthesis	❑ CUB dataset ❑ Oxford-102 dataset of flower	deep - convolutional (GANs) (DCGAN)	convolutional-recurrent network	❑ Inception scores average human ranks.	❑ Semantic consistency between the image and the text ❑ Realistic image generation ❑ Mismatch of the generated image it's correspondent text description ❑ Low image quality
(H. Zhang et al., 2017) Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks	CUB dataset	StackGAN	CNN-RNN	❑ Inception scores ❑ average human ranks.	❑ Semantic consistency between the image and the text ❑ Realistic image generation ❑ It lacks the fine-grain details of the generated image
(Xu et al., n.d.) Fine-Grained Text to Image Generation with Attentional	❑ CUB bird dataset and ❑ MS COCO dataset	AttnGAN	Bi-LSTM	❑ Inception scores ❑ R-precision	❑ The shape, color, and part of the generated image are not preserved.

Generative Adversarial Networks					❑ Semantic consistency between the image and the text
Qiao et al., 2019b (Hinz et al., 2019a) MirrorGAN: Learning Text-to-image Generation by Redescription	❑ CUB bird dataset and ❑ MS COCO dataset	MirrorGAN	Bi-LSTM	❑ Inception scores ❑ R-precision ❑ Human perceptual test	❑ The color and the part of the generated image are not preserved as compared to the ground truth image.
(Tsue et al., 2020) Proposed cycle text to image generation with BERT,	CUB bird dataset	CycleGAN	Bi-LSTM	❑ Inception scores ❑ Human perceptual test	❑ Semantic consistency between the image and the text ❑ The shape, color, and part of the generated image are not preserved.
(Gorti & Ma, 2018) Text-to-Image-to-Text Translation using Cycle Consistent Adversarial Networks	Oxford-102 dataset of flower	StackGAN with Cycle Consistency	Skip-Thought Vector network	❑ Inception scores ❑ Image color relevance score	❑ Semantic consistency between the image and the text ❑ Realistic image generation
(Tao et al., 2020)Deep Fusion Generative Adversarial Network (DFGAN)	CUB bird dataset and MS COCO dataset	DFGAN	Bi-LSTM	❑ Inception scores	❑ Lack of fine-grained word-level information during image generation. ❑ Image quality issue

Most of the above previous text-to-image creation methods used a stack of massive generators and discriminators to perform crucial improvement in the generation of high-quality images but those methods have limitations, those are: Firstly, training a massive number of networks takes longer than training a single model, and this influences the generative model's convergence and stability. Second, the initial image has a big influence on the final polished result; when the first image is poorly synthesized, subsequent generators will struggle to refine it to suitable image quality. Consequently, this structure ignores the quality of the images generated in the early stages.

Likewise, certain techniques perform better outcomes when contrasted with the past one by utilizing one generator and discriminator, these approaches generating an image from the sentence-level feature. Sentence level vector stores the whole information in a single vector, this leads to a lack the important fine-grained details information at the word level, this cause prevents the generating of high-quality images and maintaining the sematic consistency between the generated image and the text description. So, the above approaches achieved a remarkable result but still have a problem of generating quality of the image and semantic coherence of text-image pair. Due to those challenges, this area is an open task and those problems still need improvements. Therefore, Text-to-image translation is still an active research area. Finally, this paper proposed Spatial-Channel Aware Generative Adversarial network which introduces constraints like spatial attention, channel-wise attention, and DAMSM loss to solve the above problems, and certain constraints are identified to enhance and preserve the content of the generated image.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

The methodology employed in the research of GAN-based text to image translation, as well as the mathematical formulation of the evaluation metrics, are described in this chapter. This chapter goes through the methodology, dataset, and experimental measures used to address the study questions in further depth.

Let see the general block diagram of the proposed model

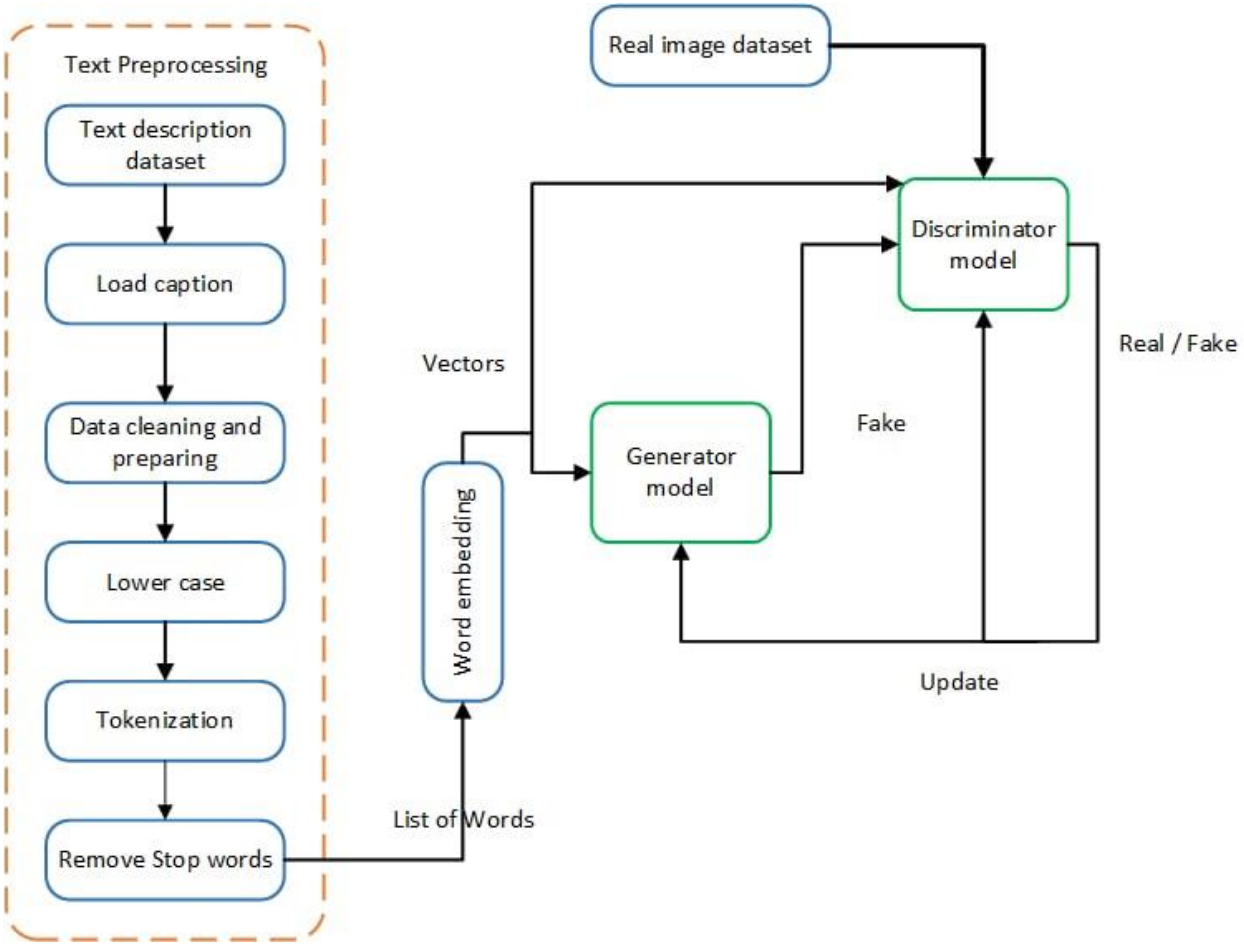


Figure 3.1. shows the overall diagram of the text to image translation

3.2. Dataset

Even though this research employs a deep learning technique to tackle text-to-image translation issues, data is an important component of the research. This text-to-image translation may be done using common datasets that are publicly available for research purposes. this dataset is: -

- **Caltech-UCSD Birds 200 (CUB-200)** is an image dataset of 11,788 pictures of 200 distinct bird species that is frequently used as a benchmark for text-to-image translation. The size of these images varies and includes a Bounding Box, Rough Segmentation, and Attributes. There are 10 captions for each image in the collection.[1]

Figure 3. 1. Training dataset Sample (from CUB-200 datasets)

Source: Adapted from (He et al., 2019)

Table 3. 1. shows the CUB-200 datasets the text description and its corresponding image.

Text description	Image
A small bird with a red head, breast, and belly, and black wing. Or the small bird has a red head with feathers that fade from red to gray from head to tail.	

[1]The CUB-200 dataset found at:
http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

a blue bird with darker markings on its wings of black and brown. Or a small blue bird, with green primaries, and a short bill.	
a large bird with a long neck, and white crown, and a gray, black, and white rest of the body. Or this bird has a white crown as well as a red bill	

The CUB-200 bird dataset is utilized to test the proposed model. This is a well-known dataset for text-to-image translation. This work splits this dataset into 8,855 images for training and 2,933 images for testing and each image has 10 captions.

Why CUB-200 bird dataset? This dataset is selected because of two reasons. The first reason is due to our GPU's speed, this work employed this dataset, which is very easy to analyze when compared to other datasets. The second reason is that this dataset has been utilized in many other studies, and this study used the CUB-200 bird dataset to compare to other studies.

3.2.1. Pre-processing Techniques

Pre-possessing is a sort of method that is utilized to change over the information into a machine-reasonable format. This work tries to generate an image from a text description. To achieve this the input data should be pre-possessed into a suitable format. Let's see the text and image pre-possessing methods.

A. Text Preprocessing

To train the proposed model, the input data must be properly preprocessed. Now let's see how to preprocess the text description. The text description is a human-readable format, but the machine learning models understand the input data in digital format, so the text description must be

converted into numbers. To do this there is a different preprocessing method was used, let discuss them separately.

- **Data cleaning and preparing**: The first step of NLP that is used to remove the punctuation, to get the alphanumeric key (i.e., number and letter).
- **Lower Case**: one of the processing of NLP that is utilized to convert over all the input text descriptions into lowercase by taking the result of the cleaned and prepared data as an input. This step is used to avoid duplication of words.
- **Tokenization**: another step of text processing that is utilized to decompose the text description into smaller units such as word-level, character-level, or sentence-level. In this work, the text description was breakdown into word-level tokenization. There are various types of tokenization from that regular tokenization was used to extract the token from the text by utilizing the regular expression. Why regular tokenization? Because the text description might have the number and the letter combination that why this work used this kind of tokenization.
- **Word embedding:** is the main part of NLP that is utilized to convert the list of words into numerical data (i.e., it transforms the given words into vector representation).

B. Image preprocessing

- **Resize the image:** The original image has different heights and widths, for example, some image has a size of 223×320 and others has 350×350 so this should be converted into the same size of the image. The proposed model accepts the image in the size of 256 × 256 due for this reason the original image should be converted into a model acceptable size. Converting the image into 256 × 256 sizes is used to minimize the number of the parameter in the model. To achieve this, bounding boxes were used to resize the image so each image has its bounding box. Next, let's see the normalization methods.
- **Image Normalization:** is a technique of image processing that is utilized to transform quite a number of pixel intensities values. The resized image was normalized into the [-1,1] range. The formula for normalized the data into [-1,1] is calculated by

$$image = \frac{image - mean}{std} \qquad eq.(3.1)$$

where the std represents the standard deviation. The value of mean and std is set to 0.5. To calculate the normalization, the image should convert into the [0,1] range. How? When the image

passes through the transform PyTorch module, it gives an expected range value of the tensor image [0,1] define by tensor dtype.

3.3. Development Tools

To design and implement the proposed work, a variety of development tools were employed. These include prototype development platforms, UML modeling tools, and other tools that are important for this research. The sections that follow provide you a short introduction to several development tools.

3.3.1. Design Tools

Design tools are methods for developing, presenting, and interpreting design concepts. Microsoft Visio (*What Is Microsoft Visio® | Lucidchart*, n.d.) is a program that allows users to make various diagrams. This is used for designing this work. It's a simple yet effective visual design tool for making professional-looking flowcharts, network diagrams, and flowchart diagrams.

3.3.2. Hardware Development tools

In order to implement this research, the following hardware tools will be needed.

Table 3. 2. Hardware tool

No.	Tools	Used for
1.	GPU	To increase the computation and to fasten the training
2.	Hard Disk	Used as storage for large datasets.
3.	RAM	To work together with the GPU to speed up the training process.

3.3.3. Software Development Framework tools

For implementing the research by coding, writing different software and coding tools will be used and those are listed below with their description.

- **TensorFlow*:*** TensorFlow is a freely available software framework for high-performance numerical analysis and processing. Why tensorflow? Because it is support wide range of tools, frameworks, and community resources, it allows developers to build machine learning applications.

- **Keras:** Keras is a python interface to a free available machine learning framework. TensorFlow is accessed using Keras, which is a user interface.
- **PyTorch** is a Python machine learning (ML) framework based on Torch. It's perhaps the most broadly utilized platform for deep learning research.
- **Anaconda**: The proposed approach is implemented with the anaconda application version 1.9.7 with 64-bit support, which is used to install the latest version of Python together with all of its numerous modules and IDEs.
- **Jupyter Notebook**: is the most widely used and convenient IDE for working with Python among AI and deep learning researchers.
- **Python**: - Python programming used to implement and demonstrate this thesis requires many of the drivers used to configure and package to install some device with the computer.

3.4. Baseline Works

The effectiveness of this approach was tested using a variety of benchmark models. The proposed model was compared against text-to-image translation models based on GAN. DFGAN (Tao et al., 2020), and CycleGAN (Tsue et al., 2020) are taken as baselines for the experiments.

- CycleGAN is a hybrid of AttnGAN and the original CycleGAN, and it generates high-quality pictures using a stack of many generators and discriminators. This model presents attention mechanisms for better semantic consistency between the text content and the created pictures. In addition, this model introduces cycle loss which is utilized to keep up with the semantic consistency between the text and the created picture by subtracting the original input text to the reconstructed text from the generated image.
- DFGAN generated high-quality pictures using a single generator and a discriminator. DFGAN provides the notion of a deep fusion mechanism for fusing text and picture features in the image process creation.

Why DFGAN and CycleGAN as baseline models? Because those models are used to show the viability of the proposed model from both the approaches which employ multiple generators and discriminator and which employ one pair of the generator and discriminator. The purpose of those models is used to show how the proposed model has improved the quality and the semantic coherence issue in the text to image translation by adding certain constraints. Both models achieved the best performance in CUB-200 datasets.

3.5. Feature Extraction Network

This section is going to describe the extraction technique required for text and images feature extraction. The feature of the text description can be extracted into a word vector or a sentence vector using a variety of approaches. The majority of the methods are covered in depth in Chapter 2, so let's focus on the text and image encoders used in this study. Bi-LSTM is a widely used technique in natural language processing. It was utilized two RNNs to create a pre-trained bidirectional RNN. The first RNN learns the input sequence, while the second RNN learns the reversal of the sequence. A pre-trained bidirectional RNN (Schuster & Paliwal, 1997) was used as a text encoder which encodes the provided text information into a word feature and a sentence feature with length *L*.

Why was Bi-LSTM selected? because the performance of the bi-LSTM is better as compared to other deep learning models for the times series data. As shown in figure 3.2 the performance of each model was demonstrated using RMSE (Chandra et al., 2021). The lower value indicates better performance.

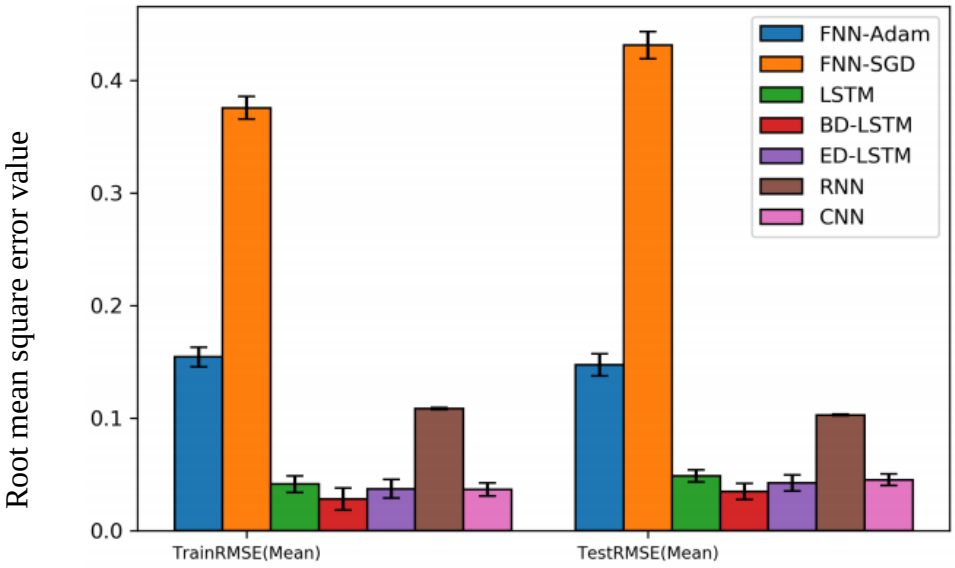


Figure 3.2. the performance evaluation of different text encoder methods through RMSE

Source: Adapted from (Chandra et al., 2021)

An image encoder is used to extract the feature of the image. Image encoders used deep neural networks such as convolutional neural networks (CNN), CNN is widely used to analyze images.

Since its debut, there have been numerous varieties of CNN models, inception v3 being one among them. Inception v3 (Szegedy et al., 2016) is most commonly used in image recognition models. On the ImageNet dataset, this model has an accuracy of more than 78.1%. To extract the information of the produced image, this thesis employed the inception v3 model.

Why inception was selected? Because the inception has a lower number of parameters as compare to VGG-16, ResNet-50, Google Net so it has a low computation cost in terms of memory and time while keeping the quality of the image (Szegedy et al., 2016). This model has a top-1 accuracy of more than 78.1%.

3.6. Evaluation Methods

There are different evaluation methods to describe the text-to-image translation model's performance on a test data set. There are different evaluation methods to present a qualitative analysis and a quantitative metric to evaluate the generated image such as the inception score, Fréchet Inception Distance score (FID), human evaluation from those inception score and human evaluation was used to evaluate this work.

3.6.1. Inception Score

The Inception Score (IS) is a measure for assessing an image's quality and diversity automatically. IS is a well-known method for assessing GANs. It uses a pre-trained inception V3 network to features extracted from both generated and true images (trained on ImageNet). A greater inception score implies that a model is of excellent quality.

$$\mathrm{DKL(P|Q)} = -\sum_{x \in X}^{n} P(x) \log\left(\frac{Q(x)}{P(x)}\right) \qquad eq.(3.2)$$

$$IS(G) = \exp\left(E_{x \sim pg} DKL(P(y|x)|P(Y)\right) \qquad eq.(3.3)$$

Where $P\ (y \mid x)$ denotes the conditional distribution, $P(y)$ is the marginal distribution, y denotes the Inception model's projected class label, and x denotes a generated sample. One probability distribution is compared to a reference distribution using the KL divergence metrics.

3.6.2. Human Evaluation Study

In this research, 30 person volunteers were asked to decide if a particular image is real or fake after viewing actual and fake images. Also, volunteers were requested to decide if the synthesized image

matches the text description. According to the figure of entities, the average score value is calculated. This research employed two question sets for the volunteer users inspired by the text to image translation human evaluation research. In the first question set users were asked to answer which one of the generated images is more realistic or natural. In the second question set, participants try to determine which generated image is more semantically coherent with the provided text description by reading the provided written description. The network will perform better if the human assessment score is greater.

CHAPTER FOUR

4. PROPOSED SPATIAL-CHANNEL AWARE GAN FOR TEXT TO IMAGE TRANSLATION

4.1. Chapter Overview

This chapter presents the proposed solution architecture with mathematical expressions and its necessary diagrammatic representation to text-to-image translation problems for improving the quality of an image and the sematic consistency between text description and generated image. This chapter is divided into seven main sections; the first introduces the proposed model architecture to translate the text-domain to image domain, the second section deals with text encoder, the third section deals with the generator network of the model, the fourth section deals with the discriminator network of the model, the fifth section explains the deep attentional multimodal similarity model, the sixth section explains the objective loss function, and the last section explains Pseudocode to the model to be trained.

4.2. Proposed Model Architecture

Spatial-Channel Aware Generative Adversarial network is the proposed model that is used to address the generating realistic high-quality image and the semantic gap between the generated image and the text description. The proposed model introduced novel constraints such as spatial attention, channel-wise attention, and DAMSM loss by extending Deep Fusion Generative Adversarial Networks. (Black background blocks indicate the newly added constraints).

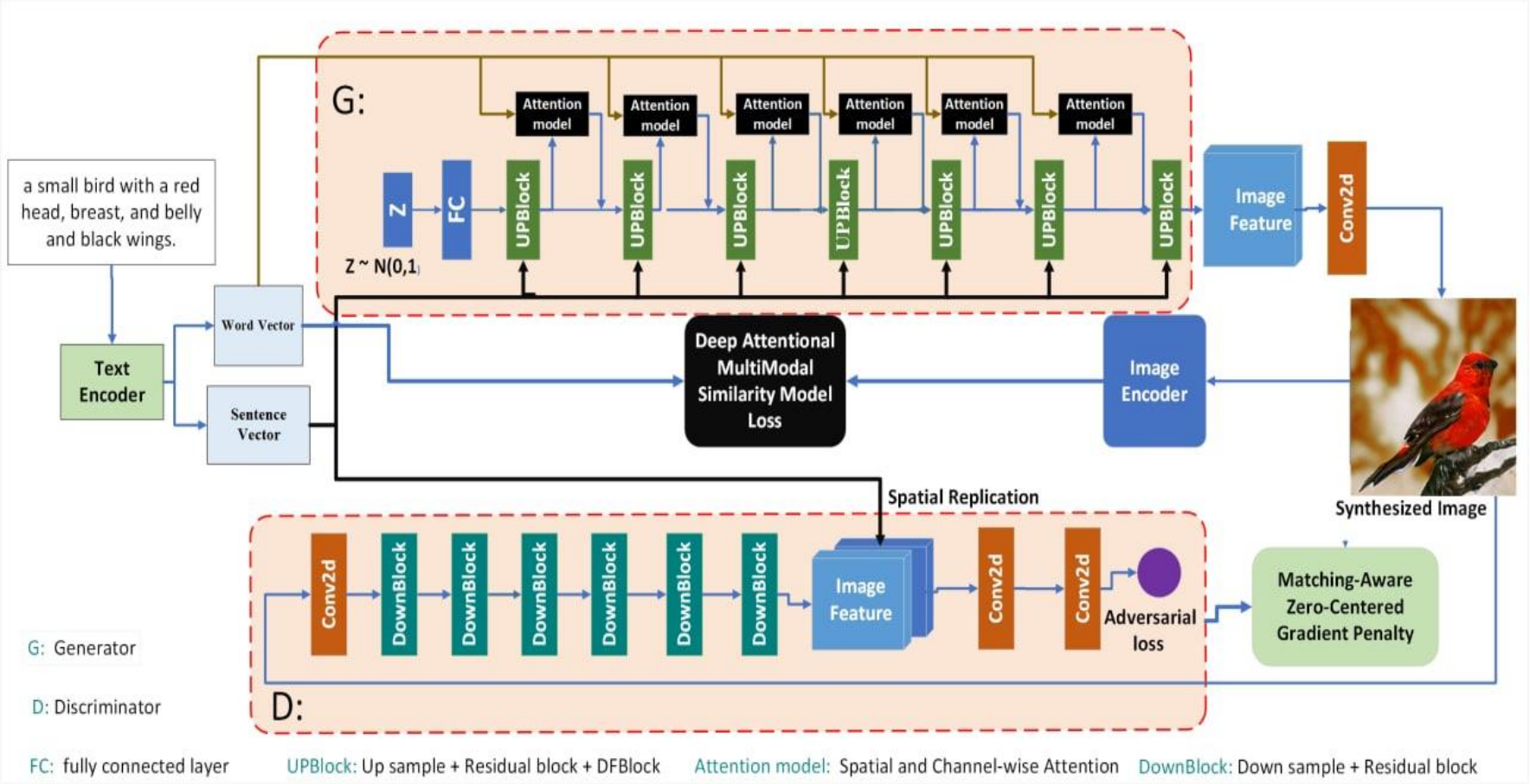


Figure 4.1. Proposed Model architecture

The proposed model design, seen in Figure 4.1, employs a single pair of generators and a discriminator to create a high-quality picture while maintaining semantic consistency in text-to-image translation. The generator and discriminator architectures are rebuilt utilizing ResNet (Sangeetha & Prasad, 2006) principles. A text encoder, generator, and discriminator make up the proposed model as a whole. Spatial attention, channel-wise attention, and DAMSM loss are three new components incorporated in the generator. The spatial attention helps the generator focus on the essential parts of the text description.

In contrast, channel-wise attention assists the generator in focusing on the crucial channels in the picture generation. Meanwhile, a novel loss function is used to preserve semantic coherence between the synthesized mage feature and the word and sentence features. Finally, spectral normalization is incorporated in the discriminator, which is utilized to stabilize the training and eliminate modal collapse issues. In the next section, the components are discussed in detail. Let's examine each of the Spatial-Channel Aware GANs building components individually.

4.3. Text Encoder

The text encoder is the initial stage in this approach, and it is responsible for converting the input sentence description into word and sentence vectors. First, the input sentence description was cleaned and prepared before generating the word and sentence vectors. All punctuation was removed from the text description, leaving just alphanumeric characters, then unified into lowercase and tokenized. After that, each description was broken down into a series of tokens. Each token is a dictionary, each with its index. As a consequence, each text will be converted into a set of indices, which will then be utilized to extract relevant word embeddings from the embedding layer, resulting in an embedding matrix. Then this matrix is fed into the dropout layer; the result of the dropout layer will be input to the bi-LSTM to extract the final word feature $w \in R^{DxT}$ and a sentence feature $s \in R^{D}$, where D is the dimension of the word feature and T is the number of tokens(words). Each word feature w_i, i.e., Each column in w is comprised of the bi-LSTM's two hidden states concatenated together, and concatenating the last two hidden states yields s , which is the sentence embedding. Next, let's discuss the next component of the proposed model which is the generator.

4.4. Generator Network

The second component is the generator, which takes two inputs: the sentence vector and a noise vector generated from a Gaussian distribution to guarantee that the image generated is varied. First, the fully connected layer is given the noise vector as input, and the fully connected layer's output is transformed into (-1,4,4); after that, a series of UPBlocks is applied to synthesized the desired image.

UPBlocks are made up of the upsample layer, residual block, and deep fusion blocks (DFBlocks). Upsample is one of the parts of UPBlocks used to upsample the visual feature by scale factor 2.

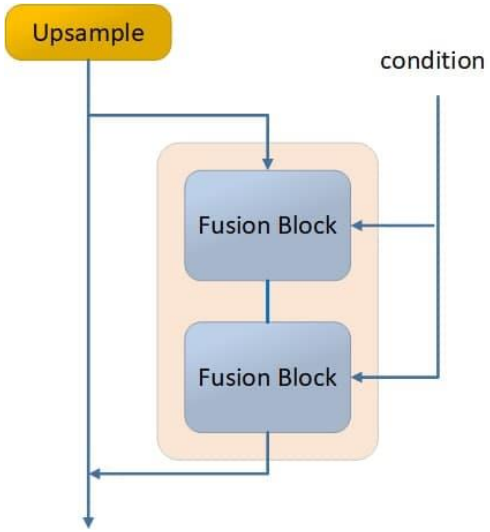


Figure 4. 2. The internal component of UPBlock

After upsampling the image features, the UPBlock fuse the text and image feature with two fusion blocks based on the condition (i.e., the condition is the input text description), as shown in Figure 4.2. Next, let's look at the details of Residual block and DFBlock.

4.4.1. Residual Block

This block is motivated by the idea of ResNet (Sangeetha & Prasad, 2006), which designs the residual block dependent on the idea of ResNet, and it contains two fusion blocks, as shown in figure 4.3.

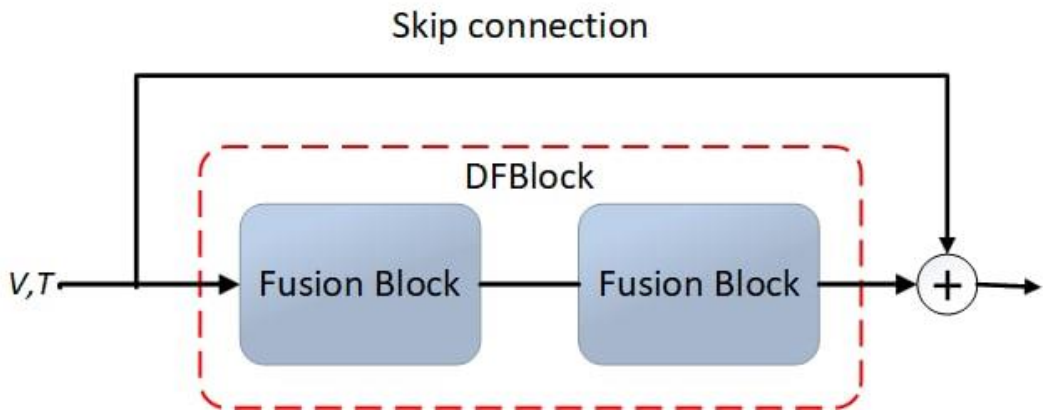


Figure 4. 3. The Residual block diagrams

V and T indicate the image and text description feature respectively. As shown in Figure 4. 3, the result of the fusion block will add to the output of skip connection, and the result will be input to the next block. All seven UPBlocks contain this residual block in their inside. The internal components of the fusion block are going to discuss in the next section.

4.4.2. Deep Fusion Block

DFBlocks is the key component of the generator that allows the image and text features to be fused successfully throughout the image production process. The DFBlock consists of affine transformations, a LeakyReLU layer, and convolution layers. Let's look at the affine transformation more in-depth.

The Affine Transformation on activation maps is extensively utilized in normalization strategies. The activation map is first normalized, then, at that point denormalized utilizing the Affine Transformation in batch normalization (BN) (Sangeetha & Prasad, 2006), the affine parameters are unconditional in BN. BN change activation maps to a normal distribution using the average and variance processed inside a batch, it could be viewed as an inversion of Affine Transformation, Batch Normalization limits contrast between visual feature during creation and influence the proficiency of earlier Affine Transformations, causing the text-image coherence to suffer (Sangeetha & Prasad, 2006). It will harm the conditional generating process. Conditional batch normalization (CBN)(Sangeetha & Prasad, 2006) uses an extra network to anticipate affine parameters from conditions; it manipulates the activation map by using the conditional information as seen in Figure 4. 4. In conditional image generation, CBN has been used often (Chen et al., 2017)(Hinz et al., 2019b).

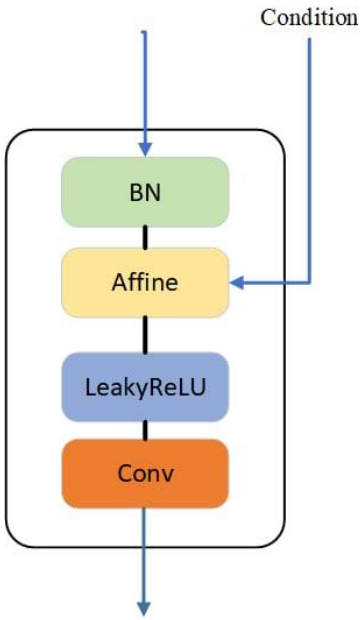


Figure 4.4. CBN Block

Source: Adapted from (De Vries et al., 2017)

The affine transformation is extracted from CBN in the proposed approach. Visual feature maps are altered using affine transformations based on natural language descriptions. The proposed model used two multilayer perceptron (MLPs). As shown below

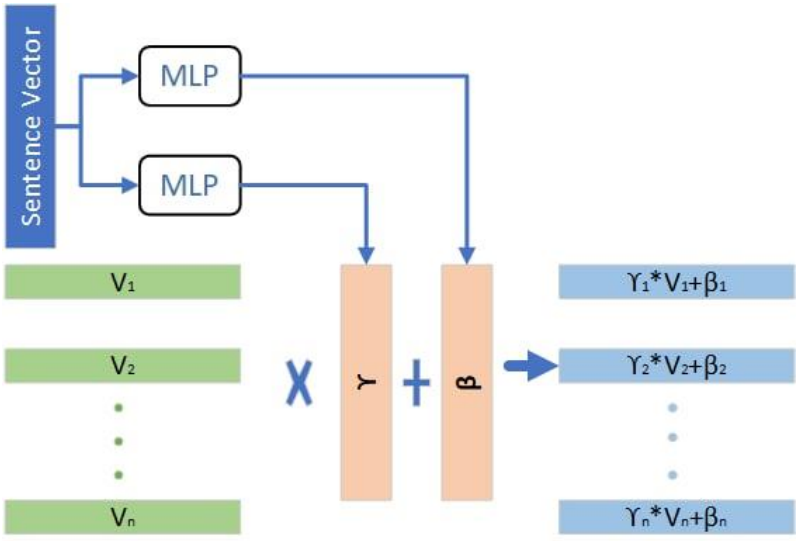


Figure 4.5. Explain the affine transformation

Source Adapted from: (Tao et al., 2020)

The above MLPs are utilized to foresee the language-adapted channel-wise scaling parameters and shifting parameters from sentence vectors s separately.

$$\gamma = MLP_1(s) \qquad \beta = MLP_2(s) \qquad eq.(4.1)$$

when the input is activation map $X \in R^{B\times C\times H\times W}$, the result of MLP is going to be a vector of size C. First, based on the scaling parameter γ, a channel-wise scaling operation performs on X, then based shifting parameter β, a channel-wise shifting operation performs on X.

$$AFF(X_i|s) = \gamma_i.X_i + \beta_i \qquad eq.(4.2)$$

Where AFF is affine transformation, γ_i is the scaling parameter that is utilized to scale X_i which is the image feature map and the β_i is used to shifting the parameter of the X_i in the i^{th} channel. Based on the channel-wise scaling and shifting, the generator attempt to synthesized high-quality images and tries to handle the semantic coherence between the text-image pair. Furthermore, in the wake of decaying the effectiveness of Affine Transformation from CBN, then, at that point the image and text feature can be fuse more in freeways. Next, the output of the affine transformation was given to LeakyReLU.

The LeakyReLU is a sort of activation function that is a further developed adaptation of the ReLU activation function. ReLU activation function makes the gradient zero value if the input is under zero, this prompts deactivate the neuron networks around that area (Pedamonti, 2018). LeakyReLU attempts to determine the issue of the ReLU activation function by process tiny linear parts. Then the result of the LeakyReLU was given to the convolution layer. The convolution layer is one of the common sorts of neural networks that are intended for chipping away at 1D, 2D, and 3D information. Convolution 2D is utilized for this work. The DFBlock contains a stack of affine transformation, LeakyReLU layers, and convolution layers are used as shown in figure 4.6.

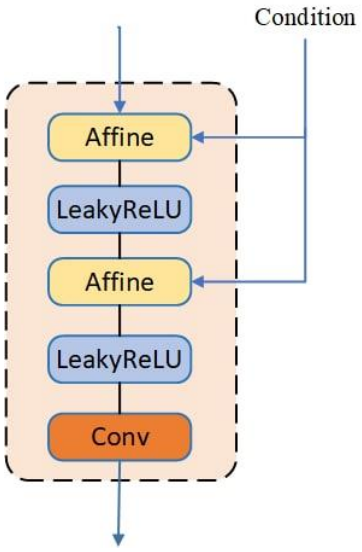


Figure 4. 6. The internal component of DFBlock

In figure 4.1, all the seven UPBlock contain the upsample, Residual block, and DFBlock. The output of the first UPBlock is used as input for the next block. After each UPBlock of the generator, the spatial and channel-wise attention mechanisms were applied. Let's look at the detail of those attention mechanisms.

4.4.3. Attention Mechanism

The significance of attention mechanisms in addressing the semantic difference between vision and language is significant. This mechanism shows its effectiveness in various research areas such as image caption, object detection, and others. There are various types of attention mechanisms among that spatial attention, and channel-wise attention is utilized in this work. Let's look at each of them in detail.

Spatial Attention: - is only used to correlate the words and the subregion of the generated images without taking channel information. This work used a spatial attention mechanism that highly correlates each of the words to its corresponding sub-region of the images, while it mainly focuses on the color descriptions.

Let's see the spatial attention in detail. The spatial attention models S^{attn} is a sort of attention mechanism that accepts two inputs: the word features $w \in R^{D\ X\ T}$ and the visual feature from the previously hidden layer as shown in figure 4.7. At different stages, the shape of the v varies. For

comprehension, it represents as $v \in R^{D'X\,N}$, where D' is the dimension of the single hidden feature and N is denoted the number of sub-regions to be generated in the image.

Before passing the word feature to the spatial attention model the word feature should transform into a common semantic space of the hidden image feature. This is achieved by running them through a new perceptron layer or a 1 × 1 convolutional layer.

$$w' = Uw \text{ where } U \in R^{D'XD}. \qquad eq.(4.3)$$

Then, let's compute s′ the similarity matrix for all pairs of word features and hidden visual features in the text-image training pair.

$$s' = v^T w' \qquad eq.(4.4)$$

Next, the word-context vector c_j is calculated by:

$$c_j = \sum_{i=0}^{T-1} \beta_{j,i} w'_i \,, \quad \text{where } \beta_{j,i} = \frac{\exp(s'_{j,i})}{\sum_{k=0}^{T-1} \exp(s'_{j,k})} \qquad eq.(4.5)$$

s'_{ji} and $\beta_{j,i}$ indicate the weight of the model used to generate the j^{th} subregion of the image from the i^{th} word. c_j indicates the weighted sum of the word features related to j^{th}. Then for the image feature v, the word-context vector can be represented as $S^{attn}(w, v) = (c_o, c_1, \ldots, c_{N-1}) \in R^{D'XN}$. Next, this word-context vector is given to channel-wise attention as input, as shown below.

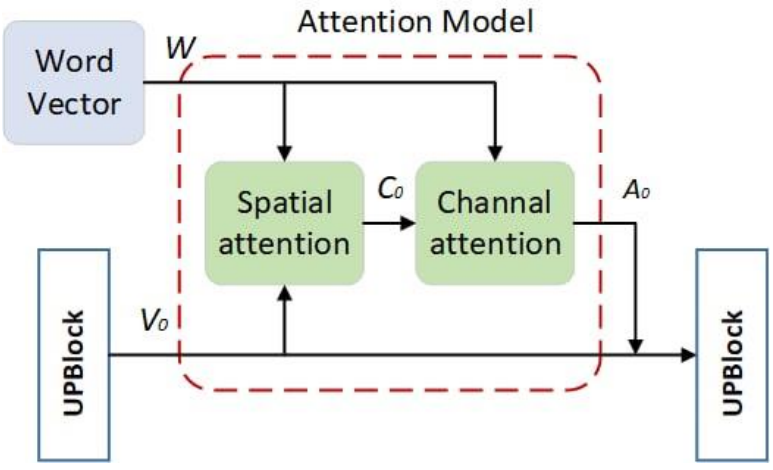


Figure 4. 7. *S*patial and channel-wise attention

In figure 4.7, spatial attention takes two inputs the word vector and the image feature, then it produces C_0 as output. the channel-wise attention takes two inputs: word vector and output of the spatial attention, then it produces the final attention which is A_0. A_0 is concatenate to image feature which is V_0 then the output is going to be input to the next UPBlock. Let's look at the detail of channel-wise attention.

Channel-wise attention: - it is another type of attention mechanism that exploits the connection between the words and the channels. This attention mechanism focuses on the important part of the channels.

Let's see the working principle of channel-wise attention. At a n^{th} stage of the model, channel-wise attention accepts two inputs: word feature w_n and the output hidden visual feature of the spatial attention $v_i \in R^{C\times(H_n * W_n)}$, where C, H_n and W_n represents the channel, height, and the weight of the visual feature map respectively at the n^{th} stage.

To calculate channel-wise attention, first, use a 1 × 1 convolutional layer to transform the word feature w_n into the same semantic space as the visual feature v_i.

$$w'_n = Uw \text{ where } U_n \in R^{(H_n * W_n)\times D} \qquad eq.(4.6)$$

The next step is to compute the channel-wise attention matrix $m^n \in R^{C\times L}$ by computing the visual feature v_n and the converted word feature w'_n, which is represented as:

$$m^n = v_n w'_n \qquad eq.(4.7)$$

As a result, m^n gathers correlation values among channels and words in all spatial regions. Then, the next step is normalized channel-wise attention matrix α^n using generated using the SoftMax function.

$$\alpha^n_{i,j} = \frac{\exp(m^n_{i,j})}{\sum_{l=0}^{L-1} \exp(m^n_{ij})} \qquad eq.(4.8)$$

The attention weight $\alpha^n_{i,j}$, represents the correlation between j^{th} word in sentence S and the i^{th} channel in the visual feature v_n. A greater number implies a large correlation.

Next, the final channel-wise attention feature is to obtain $f^\alpha_n \in R^{C\times(H_n * W_n)}$, where $f^\alpha_n = \alpha^n (w'_n)^T$

In f_n^{α}, each channel is a dynamic representation that is weighted by the relationship between the words and the visual features' respective channels. As a consequence, channels with high correlation values have a higher response to related words, which can assist extract word features into different channels and limit the impact of unimportant channels by doling out a lower connection.

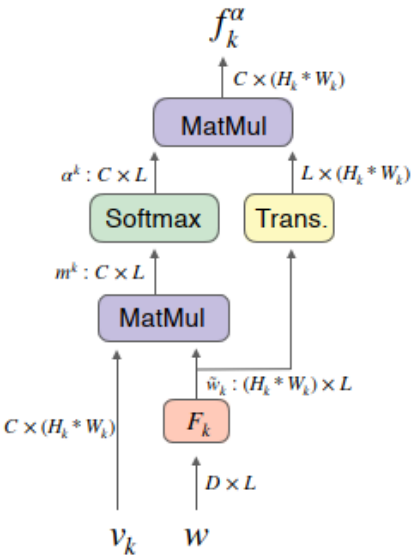


Figure 4. 8. shows the channel-wise attention

Source: Adapted from (B. Li et al., 2019)

In this thesis, both attention mechanisms are used as shown in figure 4.1. Finally, the channel-wise attention feature and the word-context vector are computed at each stage then after that the word-context vector is used as an input for calculating the channel-wise attention feature. Then, the output of the channel-wise attention feature at each stage is concatenated to the image feature at each stage, then fed to the next stage. The output of the final UPBlock is pass to the convolution layer to convert the image feature into the image with three channels (RGB). Next, let look at the discriminator network of the model and what are the internal components of the network inside it in detail.

4.5. Discriminator Network

The discriminator network is comprised of DownBlock and convolution layers. First, the generated image v, is passed through the convolution layer to convert the generated image to the feature

maps. DownBlock consists of down sampling and residual network. Then, at that point, the result of the convolution layer is down sampled by this series of DownBlock. Let's look at the detailed components of DownBlock.

DownBlock is designed by the concept of ResNet (Sangeetha & Prasad, 2006), this block contains a convolution layer, Spectral normalization, and leaky ReLU.

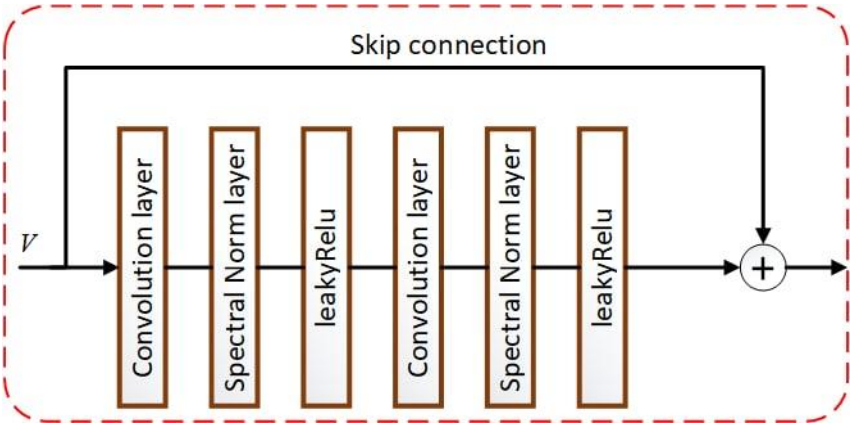


Figure 4. 9. DownBlock diagram

As shown in figure 4.9, the output of the last LeakyReLU is added to the skip connection of 1×1 convolution, then the output is used as input to the next block. DownBlock is used to extract the feature of the generated image. Then, the image feature and the sentence feature are concatenating together then the concatenating output is going to pass through two convolution layers to compute the adversarial loss. This adversarial loss is used to predict the semantic consistency and visual realism of the input. The discriminator urges the generator to generate high-quality images and semantic coherence of text-image pairs. Next, let's look at the components of discriminators.

4.5.1. Spectral Normalization

This work utilized spectral normalization to settle the training of the discriminator and apply it after each convolution to keep away from unusual gradients. The only hyperparameter of the spectral normalization that is going to tune is the Lipschitz constant of the discriminator function.

$Let\ g: h_{in} \rightarrow h_{out} \rightarrow each\ layer$, The Lipschitz norm is denoted by

$\left|\left|g\right|\right|_{Lip} = sup_h\ \sigma(\nabla g(h))$ where $\sigma(A)$ is the spectral norm of the matrix A.

$$\sigma(A) = \max_{h:h_0 \neq 0} \frac{\left||Ah|\right|_2}{\left||h|\right|_2}, \qquad eq.(4.9)$$

4.5.2. Matching-Aware Zero-Centered Gradient Penalty (MA-GP)

One of the regulatory approaches used in this work was the conditional Matching-Aware Zero-Centered Gradient Penalty (MA-GP). By pushing the actual and matched inputs to the point where the discriminator loss function gradients are minimal, the generator is encouraged to create a more realistic quality image and a text-image semantically coherent image. It smoothes the surface of the loss function of the genuine information point and its surroundings, allowing the generated information point to converge with the genuine information point.

This MA-GP utilized the zero-centered gradient penalty to genuine images with its respective sentence. The formula of MA-GP is represented as follow: -

$$kE_{x\sim P_r}[(\left||\nabla_{\mathrm{x}} D(x,s)|\right| + \left|\left|\nabla_{\mathrm{s}} D(x,s)\right)^p] \qquad eq.(4.10)$$

Where the terms k and p are hyper-parameters and this work fixes the value of $k = 2$ and $p = 6$ in the network.

4.6. Deep Attentional Multimodal Similarity Model (DAMSM)

The DAMSM is designed to match the semantic consistency of the input sentence and the output visuals by measuring the similarity between the created image and the input sentence at the word level and sentence level. It contains one text encoder and image decoder which pre-trained with the ground-truth of the text-image data pairs. To compute the DAMSM loss, the text encoder extracts the features of the input text and the image encoder extracts the feature of the generated image. Let's see the detailed description of DAMSM.

The text encoder is explained in detail in section 4.3. Next, let’s look at the image encoder explanations in detail.

4.6.1. Image Encoder

The image encoder is used to extract the local and global features of an image. As an image encoder, the inception-v3 model, which was pre-trained on ImageNet. Inception-v3 accepts the image in the size of $3 \times 299 \times 299$, So the input image should be reshaped into $3 \times 299 \times 299$. The “mixed 6e” layer was utilized to extract the local feature of an image $f \in R^{768\,X\,289}$. Then by

using a perceptron layer or 1×1 convolution, the visual features are transformed to a common semantic space for the text feature.

$$v = Wf, v \in R^{DX289} \qquad eq.(4.11)$$

Where 289 indicates the sub-region of the visual features D is the dimension of the word feature. every column of f denotes a sub-region of an image. Besides, the final average pooling layer of inception-v3 is used to extract the global feature $f' \in R^{2048}$ of the generated image. Finally, by using a perceptron layer, the picture features are transformed to a common semantic space for the text feature.

$$v' = W'f', v' \in R^{D} \qquad eq.(4.12)$$

Where D is denoted as the dimension of the picture and the textual feature space. The i^{th} column v_i is the visual feature vector for the i^{th} sub-region of the image. Next, The DAMSM loss is discussed in detail.

4.6.2. The DAMSM loss

The DAMSM loss is a loss function that is utilized to determine the likeness between the whole synthesized image and the entire sentence in both the word level and the sentence level. It is added into the network of the generator, let's dive into the detailed description.

First steps, in a text-image pair, a similarity matrix for all pairs of word feature e_j and local visual feature v_j is calculated as defined following:

$$s = e_T v, \qquad eq.(4.13)$$

Where $s \in R^{Tx289}$ and $s_{i,j}$ is the dot multiplication between the i^{th} word in the description and the j^{th} subregion in the image to access the similarity, which is exactly dotted product attention. It is normalized as follows:

$$S'_{i,j} = \frac{\exp(s_{i,j})}{\sum_{k=0}^{T-1} \exp(\mathrm{S_{k,j}})} \qquad eq.(4.14)$$

Then, at that point, the attention model is worked to process the region-context vector r_i for the i^{th} word. It is the weighted sum of the local visual features related to the i^{th} word, that is,

$$r_i = \sum_{j=0}^{288} \alpha_{i,j} v_j \ , \quad \text{where} \quad \alpha_{i,j} = \frac{\exp(\gamma_1 s_{i,j})}{\sum_{k=0}^{288} \exp(\gamma_1 \mathrm{S_{k,j}})} \qquad eq.(4.15)$$

Note that $\gamma_1 = 5$ is a metric that indicates how much attention should be devoted to those local visual features.

Next, cosine similarity is used to measure the relevance between the region-context vector r_i and the word feature e_i is defined as:

$$R(r_i, e_i) = \frac{r_i^T e_i}{||r_i|| \, ||e_i||} \qquad eq.(4.16)$$

The matching score between the whole text description T and the whole image Q is defined as follows:

$$S(T, Q) = \log\left(\sum_{i=1}^{T-1} \exp\left(\gamma_2 R(r_i, e_i)\right)\right)^{\frac{1}{\gamma_2}} \qquad eq.(4.17)$$

Where $\gamma_2 = 5$ is a metric that controls how much the most significant pair of region-context vector and word feature should be emphasized.

Finally, given a group of text-image pairs$\{(T_i, Q_i)\}_{i=1}^{M}$, $M = 10$ the posterior probability of description T_i being matching with the image Q_i is calculated by:

$$P(T_i|Q_i) = \frac{\exp\left(\gamma_3 S(T_i, Q_i)\right)}{\sum_{j=1}^{M} \exp\left(\gamma_3 \mathrm{S(T_i, Q_i)}\right)} \ , \qquad eq.(4.18)$$

Where $\gamma_3 = 10$ is a smoothing metric by the experiments. Note that only T_i matches with Q_i, while the other $M - 1$ description candidates are mismatching ones. In the same ways, Let's calculate the posterior probability of image Q_i being matching with description T_i.

$$P(Q_i|T_i) = \frac{\exp\left(\gamma_3 S(Q_i|T_i)\right)}{\sum_{j=1}^{N} \exp\left(\gamma_3 \mathrm{S}(Q_i|T_i)\right)} \ , \qquad eq.(4.19)$$

Then, the loss function at the word level $\mathcal{L}^w$ is denoted as

$$\mathcal{L}^w = -\sum_{i=1}^{M} \log P(T_i|Q_i) - \sum_{i=1}^{M} \log P(Q_i|T_i) \qquad eq.(4.20)$$

Where w stands for "word"

On the other hand, when taking the sentence feature e' and global visual feature v' into account, the matching score in Equation 4.17 can be redefined as:

$$S(T,Q) = \frac{{v'}^T e'}{|v'||\, e'|} \qquad eq.(4.21)$$

that is the posterior probability of image Q_i being matching with description T_i. Thus, the loss function at the sentence level $\mathcal{L}^s$ is defined as

$$\mathcal{L}^s = -\sum_{i=1}^{M} \log P(T_i|Q_i) - \sum_{i=1}^{M} \log P(Q_i|T_i) \qquad eq.(4.22)$$

After replacing it in Equation 4.18, the loss function at the sentence level $\mathcal{L}^s$ is achieved. Eventually, the DAMSM loss is the sum of the two losses, i.e.,

$$\mathcal{L}_{DAMSM} = \mathcal{L}^w + \mathcal{L}^s \qquad eq.(4.23)$$

Finally, As shown in figure 4.1 this DAMSM loss is used in this thesis and added to the generator to improve the generated image and to keep the semantic coherence between the text feature and the generated images.

4.7. Objective Functions

This work used a hinge version of the adversarial loss function to stabilize the training process. Since hinge loss is utilized for multi-class classification to punishes misclassifications and prompts better exactness. The hinge loss is utilized to limit the loss between -1 and 1.

The discriminator loss expresses as follow: -

$$\begin{aligned} L_D = & -E_{x\sim P_r}\big[\min\big(0,-1+D(v,s)\big)\big] \qquad eq.(4.24) \\ & -\left(\frac{1}{2}\right)E_{G(z)\sim P_g}\big[\min\big(0,-1-D(G(z),s)\big)\big] \\ & -\left(\frac{1}{2}\right)E_{x\sim P_{mis}}\big[\min\big(0,-1-D(x,s)\big)\big] \end{aligned}$$

$$+ kE_{x\sim P_r}[\left(\left|\left|\nabla_{\mathrm{x}} D(x,s)\right|\right| + \left|\left|\nabla_{\mathrm{s}} D(x,s)\right|\right)^{p}]$$

Where v is the image feature, s is the sentence vector, z is noise, and D is the discriminator network. In generator loss, the DAMSM loss is added as represented below

$$L_G = -E_{G(z)\sim P_g} D(G(z),s) + \mathcal{L}_{DAMSM} \qquad eq.(4.25)$$

4.8. Training Pseudocode

This work has been introduced a highlight about the training algorithm of the proposed model architecture.

Algorithm 1: Spatial-Channel Aware Generative Adversarial network with novel components spatial, channel-wise attention, and DAMSM loss.

Input: $s, w, noise$

Output: Generated image: h

1. $Take\ a\ sample\ of\ batch\ of\ text\ decription: T$
2. Train D:
3. $Translate\ T\ to\ h:\ fake = G_{Th}(s, noise, w)$
4. $D_{real}(h,s), D_{fake}(fake,s), D_{mismatch}(h,s)\ with\ mismatch\ of\ s\ and\ h$
5. $dloss = D_{real}(h,s) + \frac{1}{2} * (D_{fake}(fake,s) + D_{mismatch}(real,s))$
6. $Update\ the\ dloss\ to\ minmise\ the\ classification\ loss.$
7. $Train\ G:$
8. $G_{TQ}(s,n,w)$:
9. $Reapeat\ until\ 6\ block:$
10. $\boldsymbol{S = S^{attn}(w, h_0)}$
11. $\boldsymbol{C = C^{attn}(S, w)}$
12. $\boldsymbol{h = cat(C, h_0)}$
13. $\boldsymbol{h = fuse\ s\ and\ h}$
14. $\boldsymbol{Return\ h}$
15. $fake = G_{TQ}(s, noise, w)$
16. $g_{loss} = D_{fake}(fake, t)$

17. $$\boldsymbol{DAMSM\_loss} = \mathcal{L}^{w} + \mathcal{L}^{s}$$

18. $$g_{totalLoss} = g_{loss} + \boldsymbol{DAMSM\_loss}$$

The bold line is the added constraints

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

In this chapter, the implementation of the proposed model architecture is discussed. The working environment, Spatial-Channel Aware Generative Adversarial Networks implementation, and experiments undertaken will be discussed.

5.2. Working Environment

This part clarifies the hardware tools that have been used to execute GAN experiments just as depicting the hardware stack.

- Laptop: The Laptop is utilized for fostering a model design.
  - Operating system: Windows 10
  - Processor: Intel ® Core™ i5-7200U CPU @ 2.50GHz 2.70GHz
  - Graphics: Intel ® Graphics 3000
  - Primary Memory (RAM): 12.00 GB
  - System Type: 64-bit Operating System, x64-based Processor
- Desktop: The desktop computer is used for developing a text-to-image translation.
  - Operating system: Windows 10
  - Processor: Intel ® Core™ i5-4580 CPU @ 3.29GHz x 4
  - Graphics: Intel ® HD Graphics 4600
  - GPU: GeForce RTX 2070 Super 6 GB RAM
  - Primary Memory (RAM): 8.00 GB
  - System Type: 64-bit Operating System, x64-based Processor

Visual studio code is utilized as an improvement IDE, with python mediator 3.6 on a PC. For carrying out the proposed model. In addition, PyTorch 1.0 was used. In the accompanying section rundown of experiments class coordinated for evaluating the proposed hypothesis is discussed.

5.3. Environmental Setup

In this study, particular programming and IDEs have been used.

Anaconda: in an application used to present the extraordinary adaptation of python with its different modules and IDEs, for executing the proposed model an anaconda application form 1.9.7 with 64-bit support is utilized.

Jupyter Notebook: is the most renowned and helpful IDE among AI and significant learning examiners to work with python and utilizes Jupiter note pad version 6.0.0. Next, let's look at the detailed implementation of the proposed model.

5.4. Spatial-Channel Aware Generative Adversarial Network Implementation

The proposed model employs one generator and discriminator to generate a high-quality image and to keep up with the semantic coherence between the created image and the written information as described in chapter four in detail. The proposed model is made out of a text encoder, generator, and discriminator. Let's see the implementation of each component of the proposed model.

The text encoder is the initial stage in the process, which uses a pre-trained text encoder bi-LSTM model to transform the given input text description into word and sentence vectors. The number of inputs =300, number of hidden layers=128, number of output layers =1, the drop out parameter is 0.5 the word embedding =256, and the number of the word is 18.

```
        if self.rnn_type == 'LSTM':
            self.rnn = nn.LSTM(self.ninput, self.nhidden,
                               self.nlayers, batch_first=True,
                               dropout=self.drop_prob,
                               bidirectional=self.bidirectional
)
```

Snippet code 5. 1. Text encoder code

Then, the generator is the second part of the model, and it accepts two inputs: the sentence vector and a noise vector taken from a Gaussian distribution to ensure that the image created is varied. First, the noise vector is given as input to the fully connected layer, The fully connected layer takes

two inputs the noise vector is 100 and the number of the output layers is ngf*128. The value of ngf (network growth factor) is 64.

```
self.fc = nn.Linear(nz, ngf*8*4*4)
```

Snippet code 5. 2. The output of fully connected layer

Then the output of the fully connected layer is reshaped into (-1,4,4), then fed into the first UPBlock.

```
out = self.fc(x)
out = out.view(x.size(0), 8*self.ngf, 4, 4)
out = self.block0(out,c)
```

Snippet code 5. 3. Reshaping the output of a fully connected layer

after that, a series of up-sampling blocks is applied to upsample the image feature. UPBlocks are comprised of the upsample layer, residual block, and DFBlocks. The upsample is used to up the image feature by a scale factor of 2. DFBlocks that used to fuse the image and the text feature during the image generation process. DFBlock is are comprised of a series of affine transformations, LeakyReLU layer, and 2 convolution layers with 3×3 filter size, a stride of 1, and padding 1.

```
def residual(self, x, y=None):
        h = self.affine0(x, y)
        h = nn.LeakyReLU(0.2,inplace=True)(h)
        h = self.affine1(h, y)
        h = nn.LeakyReLU(0.2,inplace=True)(h)
        h = self.c1(h)

        h = self.affine2(h, y)
        h = nn.LeakyReLU(0.2,inplace=True)(h)
        h = self.affine3(h, y)
        h = nn.LeakyReLU(0.2,inplace=True)(h)
        return self.c2(h)
```

Snippet code 5. 4. Deep fusion blocks

After the DFBlock the concept of the residual net is implemented the skip connection as express below.

```
def forward(self, x, y=None):
      return self.shortcut(x) + self.gamma * self.residual(x, y
)
```

Snippet code 5. 5. Execution of DFBolcks with skip connection

In the above residual implementation, the convolution $c1$ and c2 are expressed below

```
self.c1 = nn.Conv2d(in_ch, out_ch, 3, 1, 1)
self.c2 = nn.Conv2d(out_ch, out_ch, 3, 1, 1)
```

Snippet code 5. 6. Convolutions used in DFBlocks

After each block of the generator, the attention mechanism is applied. Then DAMSM loss is calculated and added to the generator loss for better semantic coherence between the visual feature and the word feature. After the DFBlock the convolution layer is employed to transform the image feature into images. This convolution layer contains a 3×3 filter size, a stride of 1, and padding 1.

```
self.conv_img = nn.Sequential(
            nn.LeakyReLU(0.2,inplace=True),
            nn.Conv2d(ngf, 3, 3, 1, 1),
            nn.Tanh(),)
```

Snippet code 5. 7. Convolution is used to convert image features into image

A down block and convolution layers make up the discriminator. First, the generated image transforms into an image feature using the convolution layer contains a 3×3 filter size, a stride of 1, and padding 1.

```
self.conv_img = nn.Conv2d(3, ndf, 3, 1, 1) #ndf=32
```

Snippet code 5. 8. Convolution is used to convert an image into an image feature

Then, the output is downsampled by a series of down blocks. The DownBlocks contain two series of convolution, spectral normalization, and LeakyReLU. The first layer of the DownBlocks

contains convolution with a 4× 4 filter size, a stride of 2, and padding 1, and the output a pass to spectral normalization and serve as input to LeakyReLU. After that, the second layer contains convolution layer is used with filter 3 × 3, stride 1, and padding 1. Then, the output serves as input for spectral normalization. Then the output passthrough LeakyReLU.

```
self.conv_r = nn.Sequential(
            #spectral_norm
       spectral_norm(nn.Conv2d(fin, fout, 4, 2, 1, bias=False)),
       nn.LeakyReLU(0.2, inplace=True),
            #spectral_norm
      spectral_norm(nn.Conv2d(fout, fout, 3, 1, 1, bias=False)),
      nn.LeakyReLU(0.2, inplace=True),)
```

Snippet code 5. 9. Spectral Normalization after each convolution

After the above process, the sentence vector and the image features are concatenated as shown in the following code.

```
def forward(self, out, y):
        y = y.view(-1, 256, 1, 1)
        y = y.repeat(1, 1, 4, 4)
        h_c_code = torch.cat((out, y), 1)#768
        out = self.joint_conv(h_c_code)
        return out
```

Snippet code 5. 10. Concatenating the image feature and the text feature

After the image and text feature is concatenated the concatenating channel is 768, it passed through the two convolutions which are implemented as follows: the first convolution takes input channel =768 and output channel =64 with the filter size of 3 × 3, stride 1, and padding 1. The second convolution channel =64 and output channel =1 with a filter size of 4 × 4, stride 1, and padding 0.

```
self.joint_conv = nn.Sequential(
          nn.Conv2d(ndf * 16+256, ndf * 2, 3, 1, 1, bias=False),
```

```
            nn.LeakyReLU(0.2,inplace=True),
            nn.Conv2d(ndf * 2, 1, 4, 1, 0, bias=False),)
```

Snippet code 5. 11. Convolution to convert the concatenating feature into one channel

Then Adversarial loss is computed to evaluate the generated image is real or fake. By recognizing the fake image from genuine samples. The discriminator urges the generator to produce images of better quality and semantic coherence between text and image.

5.4.1. Summary of Generator and Discriminator

All the details part of the generator (UPBlocks) and the discriminator(DownBlocks) are covered in the upper section. This sub-section tries to show the number of channels and the size of the images at each stage of the generator (UPBlocks) and the discriminator(DownBlocks).

Table 5.1. demonstrate the general input and output of the UPBlocks

Generator UPBlocks	Number of channels				Size of image
	UPBlocks		Attention		
	Input	Output	Input	Output	
Input	256	-	-	-	4×4
UPBlocks 0	256	256	256	256	4×4
UPBlocks 1	256	256	256	256	8×8
UPBlocks 2	256	256	256	256	16×16
UPBlocks 3	256	256	256	256	32×32
UPBlocks 4	256	128	128	128	64×64
UPBlocks 5	128	64	64	64	128×128
UPBlocks 6	64	32	-	-	256×256
Convolution layer (32,3, filter-3, stride-1, padding-1)	32	3	-	-	256×256

As shown in table 5.1. each of UPBlocks input and outputs is described. first, the input 256×4×4 input to the first UPBlocks. After that, a series of UPBlocks is applied and the correspondence output is presented.

Table 5.1 explains each input and output of the generator in each UPBlocks and attention model, the first UPBlock 0 takes 256 channel and 4×4 size images (256×4×4) produces 256 channels and 4×4 image size. Then the first attention model takes two inputs which are word vector and the output of the first UPBlocks 0 then the output of the attention model is 256×4×4 concatenating to the first output of the UPBlocks then the output is input to the next UPBlocks 1. UPBlocks 1 accepts the 256 channel and 4×4 size images as input and produces an output 256 channel and 8×8 size images. The second attention model accepts 256 channel and 8×8 size images and produces 256 channel and 8×8 then the output will input to the next UPBlock2. UPBlocks 2 takes 256 channels and 8×8 as input and produces 256 channels and 16×16 image size. The third attention model accepts 256 channel and 16×16 size images and produces 256 channel and 16×16 then the output will be input to the next UPBlock3.

UPBlocks 3 takes 256 channels and 16×16 as input and produces 256 channels and 32×32 image size. The third attention model accepts 256 channel and 32×32 size images and produces 256 channel and 32×32 then the output will be input to the next UPBlock4. UPBlocks 4 takes 256 channels and 32×32 as input and produces 128 channels and 64×64 image size. The third attention model accepts 128 channel and 64×64 size images and produces 128 channel and 64×64 then the output will be input to the next UPBlock5. UPBlocks 5 takes 128 channels and 64×64 as input and produces 64 channels and 128×128 image size. The third attention model accepts 64 channel and 128×128 size images and produces 64 channel and 128×128 then the output will be input to the next UPBlock6.

UPBlocks 6 takes 64 channels and 128×128 as input and produces 32 channels and 256×256 image size. The third attention model accepts 32 channel and 256×256 size images and produces 32 channel and 256×256 then the output will be input to the next UPBlock7. The last convolution converts the 32 channel and 256×256 image size into 3 channel and 256×256 image size.

Table 5.2. demonstrate the general input and output of the DownBlocks

Discrmnator DownBlocks	Number of channels		Size of image
	Input	Output	
Input image	3	-	256×256

Convolution layer (3,32, filter -3,stride-1,padding-1)	3	32	256×256
DownBlocks 1	32	64	128×128
DownBlocks 2	64	128	64×64
DownBlocks 3	128	256	32×32
DownBlocks 4	256	512	16×16
DownBlocks 5	512	512	8×8
DownBlocks 6	512	512	4×4
Convolution layer (512+256, 64, filter-3, stride-1, padding-1) concatination of image channale and sentence channel	512+256	64	4×4
Convolution layer (64,1, filter-4, stride-1, padding-0)	64	1	1×1

As shown in table 5.2. each of DownBlocks input and outputs is described. first, the convolution layer converts into 3×256×256 image into 32×256×256. Then this 32×256×256 will be input to the first DownBlocks. After that, a series of DownBlocks are applied.

5.5. Spatial And Channel-wise Attention Implementation

As discussed in the previous section, spatial attention is used to focus the relevant part of the text description to synthesized the fine-grain detail of an image, this spatial attention takes two inputs the image feature at a certain stage and the word feature.

```
class SpatialAttention(nn.Module):
    def __init__(self, idf, cdf, what):
        super(SpatialAttention, self).__init__()
        self.conv_context4 = conv1x14(cdf, idf)
        self.conv_context8 = conv1x18(cdf, idf)
        self.conv_context16 = conv1x116(cdf, idf)
        self.conv_context32 = conv1x132(cdf, idf)
        self.conv_context64 = conv1x164(cdf, idf)
        self.conv_context128 = conv1x1128(cdf, idf)
```

Snippet code 5. 12. Spatial attention code

To compute the word-level vector feature the word features should convert to common semantic space.

```
def conv1x1(in_planes, out_planes):
    "1x1 convolution with padding"
    return nn.Conv2d(in_planes, out_planes, kernel_size=1, stride=1, padding=0, bias=False)
```

Snippet code 5. 13. Convolution to convert the channel

channel-wise attention is also used to focus on the important channel for synthesized an image. It takes two inputs the image feature which is the output of the spatial attention and the word feature.

```
class ChannelAttention(nn.Module):
    def __init__(self, idf, cdf, what):
        super(ChannelAttention, self).__init__()
        self.conv_context000 = conv1x1(cdf, 4*4)
        self.conv_context00 = conv1x1(cdf, 8*8)
        self.conv_context0 = conv1x1(cdf, 16*16)
        self.conv_context1 = conv1x1(cdf, 32*32)
        self.conv_context2 = conv1x1(cdf, 64*64)
        self.conv_context3 = conv1x1(cdf, 128*128)
        self.sm = nn.Softmax()
```

Snippet code 5. 14. Channel -wise attention code

5.6. Deep Attentional Multimodal Similarity Model Implementation

This work also introduces the DAMSM loss, as discussed in the early section, it's utilized to figure out how similar the word feature and the produced image is. Before calculating the DAMSM loss the sentence and the word loss are calculated as shown below. The word loss is calculated as follow

```
region_features, cnn_code = image_encoder(fake_imgs)
w_loss0, w_loss1, _ = words_loss(region_features, words_embs,
                                 match_labels, cap_lens,
```

```
                                    class_ids, batch_size)
w_loss = (w_loss0 + w_loss1) * cfg.TRAIN.SMOOTH.LAMBDA
```

Snippet code 5. 15. Computing the word loss

The sentence loss is calculated as follow

```
s_loss0, s_loss1 = sent_loss(cnn_code, sent_emb,
                                match_labels, class_ids, batch_size)
s_loss = (s_loss0 + s_loss1) * cfg.TRAIN.SMOOTH.LAMBDA
```

Snippet code 5. 16. Computing sentence loss

When the total sentence loss is computed it multiply by the factor of $lambda = 1.0$

The DAMSM loss is computed by taking the sentence loss and the text loss as input.

```
def DAMSM_loss(image_encoder, fake_imgs, real_labels,
               words_embs, sent_emb, match_labels,
               cap_lens, class_ids):
```

Snippet code 5. 17. DAMSM loss function

Add the two losses: sentence loss and word loss to get DAMSM loss

```
DAMSM = w_loss + s_loss
return DAMSM

 DAMSM = 0.05 * DAMSM_loss(image_encoder, fake, real_labels,

words_embs,sent_emb, match_labels,
                           cap_lens, class_ids)
errG_total = errG + DAMSM
```

Snippet code 5. 18. Adding the DAMSM loss to generator loss

5.7. Discriminator Network Implementation

This work is used additional spectral normalization in the discriminator part for stable taring and to avoid modal collapse problems. This spectral normalization is added after each convolution layer in the discriminator.

```
self.conv_r = nn.Sequential(
            #spectral_norm
       spectral_norm(nn.Conv2d(fin, fout, 4, 2, 1, bias=False)),
       nn.LeakyReLU(0.2, inplace=True),
            #spectral_norm
      spectral_norm(nn.Conv2d(fout, fout, 3, 1, 1, bias=False)),
      nn.LeakyReLU(0.2, inplace=True),)
```

Snippet code 5. 19. Implementation of spectral normalization

5.8. Experiment Class

To improve the text to image translation different hypotheses should be rise. eight different classes of experiments were conducted and two models were implemented as a baseline as shown in table 5.3.

CG: - The cycle text to image translation with BERT is the baseline paper of this work. This baseline model is the combination of AttnGAN and CycleGAN and this model is implemented in the same environment.

DF: - Deep fusion Generative Adversarial Networks (DFGAN) is a base for this proposed model that employs single pair of generators and discriminators to generate a better-quality image. This model is implemented to evaluate this work.

All the above experiments are the baseline experiments that are conducted to validate the new experiments conducted below.

DF+C: - DFGAN with cycle consistency loss. DF+C experiment introduced a cycle consistency loss (caption loss) that was used to calculate the difference between the original input text and the reconstructed text description. The reconstructed text is extracted from the generated images using the CNN model which is inception v3 and this extracted image feature pass through the RNN

models to get the reconstructed text. This loss was added to the generator loss that was used to maintaining semantic consistency between the input text and the generated images. The training tuning parameters in this experiment is set to the optimized using Adam with $\beta_1 = 0.0$ and $\beta_2 = 0.9$, the learning rate is set to 0.0001 for the generator and 0.0004 for the discriminator and the Batch size is 12, lambda value is 1.0.

DF+B: - DFGAN with BERT, this experiment raises the question of what if the text encoder is improved or substitute by another transformer text encoder model? To answer this question or to prove this hypothesis the bidirectional encoder representational transformer (BERT) is introduced it is one of the transformers approaches of text encoders that are used to encode the given text description effectively. This experiment substitutes the previous text encoder found in DFGAN through BERT, then this model is trained by setting up the training tuning parameters, In this experiment, Adam optimizer was used with $\beta_1 = 0.0$ and $\beta_2 = 0.9$ and the learning rate is set to 0.0001 for the generator and 0.0004 for the discriminator, and the Batch size is 12.

DF+SA: - This experiment raises the question of what if the attention mechanisms are employed to the model for a better-quality image generation? To prove this hypothesis spatial attention was introduced. Spatial attention is combined with DFGAN. Spatial attention is a kind of attention that focuses on just correlating words with sub-region of image and ignores channel information. This spatial attention was added in all blocks 1, 2,3,4,5, and 6 in the generator to focus on fine-grain details during the generation of images. This experiment was trained by setting up the training tuning parameters of the optimized using Adam with $\beta_1 = 0.0$ and $\beta_2 = 0.9$, the learning rate is set to 0.0001 for the generator and 0.0004 for the discriminator, and the Batch size is 12.

DF+SA+D: - This experiment adapts the DF+SA experiment. This experiment raises the question what if a certain loss function is added to the generator to keep the text image semantic consistency? To prove this hypothesis this experiment introduces DAMSM loss. DAMSM loss introduced better semantic consistency between the image feature and text feature which is the word and sentence feature. This DAMSM loss is explained in detail in chapter four section 4.6. DAMSM loss is added to the generator loss to learn the generator to synthesized the realistic image and to maintain the semantic consistency of the text-image domain. During training this experiment the hyperparameters of the DAMSM loss are fixed to $\gamma_1 = 5$, $\gamma_2 = 5$ and $\gamma_3 = 10$. By adding this loss function the model was trained. The training tuning parameters in this

experiment the optimized using Adam with $\beta_1 = 0.0$ and $\beta_2 = 0.9$, the learning rate is set to 0.0001 for the generator and 0.0004 for the discriminator, and the Batch size is 12.

DF+SA+CA: - This experiment extends the DF+SA experiment**.** By observing from the previous experiment, the attention mechanism has a great influence on the generation of the image so this experiment introduces an additional attention mechanism which is channel-wise attention. This experiment is being carried out to see what would happen if spatial and channel-wise attention were combined to improve the generated image. channel-wise attention is used to give more attention to the most important part of the channel for image generation. Each attention was added to the generator model after blocks 1,2,3,4,5 and 6. This experiment is trained by keeping the training tuning parameters optimized using Adam with $\beta_1 = 0.0$ and $\beta_2 = 0.9$, the learning rate is set to 0.0001 for the generator and 0.0004 for the discriminator, and the Batch size is 12.

DF+SA+CA+D: - this experiment adapts the DF+SA+CA experiment. This experiment raises the question what if a certain loss function is added to the generator to keep the text-image semantic consistency? To prove this hypothesis, this experiment introduces DAMSM loss. DAMSM loss introduced better semantic coherence between the image feature and text feature which is the word and sentence feature. This DAMSM loss is explained in detail in chapter four section 4.6. DAMSM loss is added to the generator loss to learn the generator to synthesized the realistic image and to handle the semantic coherence of the text-image domain. During training this experiment the hyperparameters of the DAMSM loss are fixed to $\gamma_1 = 5$, $\gamma_2 = 5$ and $\gamma_3 = 10$. By adding this loss function the model was trained. The training tuning parameters in this experiment the optimized using Adam with $\beta_1 = 0.0$ and $\beta_2 = 0.9$, the learning rate is set to 0.0001 for the generator and 0.0004 for the discriminator, and the Batch size is 12.

Training Details

Why Adam optimizer is used in this experiment? Adam joins the great behaviors of the AdaGrad and RMSProp calculations to provide an optimization set of rules which can address sparse gradients on noisy problems (Kingma & Ba, 2015). The other reason is, it is appropriate for issues that are huge as far as data or parameters and it requires little memory. Due to those reasons, an Adam optimizer was used in all experiments.

Why the learning rate is set to 0.0001 for Generator and 0.0004 for Discriminator? Those learning rates were set according to two time-scale update rules (TTUR) (Heusel et al., 2017) to ensure the convergence of GAN training.

Why DAMSM loss hyperparameters were fixed to $\gamma_1 = 5$, $\gamma_2 = 5$ and $\gamma_3 = 10$. ? these experiments used those values by reviews papers, most of the papers recommend this value (Tsue et al., 2020)(Xu et al., 2017a).

Table 5. 3. Lists of experimental classes

Notation	**Experiment**	**Datasets**	**Model used**
CC	CycleGAN baseline	CUB-200 dataset	Cycle-GAN
DF	DFGAN	CUB-200 dataset	DFGAN
DF + C	DFGAN with CycleGAN generator trained with caption loss	CUB-200 dataset	DFGAN with caption loss
DF + B	DFGAN with BERT Text feature extract with BERT	CUB-200 dataset	DFGAN with BERT
DF+SA	DFGAN with spatial attention add at the generator to focus on fine-grain details during generation at blocks 1,2,3,4,5 and 6.	CUB-200 dataset	DFGAN with spatial Attention
DF+SA+D	DFGAN with spatial attention add at the generator to focus on fine-grain details during generation at blocks 1,2,3,4,5 and 6 and DAMSM loss	CUB-200 dataset	DFGAN with spatial Attention plus DAMSM loss
DF+SA+CA	DFGAN with spatial and channel-wise attention add at generator at blocks 1,2,3,4,5 and 6.	CUB-200 dataset	DFGAN with spatial and channel-wise Attention
DF+SA+CA+D	DFGAN with spatial and channel-wise attention add at generator at block 1,2,3,4,5 and 6 and DAMSM loss is added	CUB-200 dataset	DFGAN with spatial and channel-wise Attention plus DAMSM loss

CHAPTER SIX

6. RESULTS AND DISCUSSIONS

6.1. Chapter Overview

In this chapter, this work follows up the proposed methods stated in the previous chapters. The results of the experimental stages are reported in this chapter. The proposed methods' outcomes are compared with other state-of-the-art methods such as conditional GAN, StackGAN-v1, StackGAN-v2, CycleGAN, and DFGAN through different evaluation metrics such as inception score and human evaluation. This thesis tries to prove the two hypotheses: "*adding attention mechanism constraints would improve the quality of the generated image"*, the second is "adding *DAMSM loss constraint would improve the semantic consistency between the fake image and the textual description.*"

6.2. Text-to-Image Translation

The text to image translation is one of the GAN applications that translates the given textual description domain into an image domain. This study was conducted around six experiments to prove the above two hypotheses. Let's look at the steps in how the proposed model came about.

To obtain the proposed model, six experiments were carried out using different components and different methods. The DF experiment is the basic work of DFGAN, and DFGAN was implemented as it is. The DF experiment gave better results, but the quality was lower images, and also there are semantic consistency issues. It trains up to 301 epochs, but after the 234 epochs during training, it started to generate a black image. To overcome these problems, the first experiment was carried out, namely DF + C. DF + C, the experiment introduced the cycle consistency loss or caption loss, which is used to maintain the semantic consistency between text and image pairs. DF + C gives better results than the original DF experiment. This DF + C experiment tries to maintain the semantic gap, but still needs improvement, and there are also problems in generating high-quality images. To alleviate these problems, a DF+B experiment was carried out. DF + B introduces bi-directional encoder representation from a transformer (BERT) used to enhance the text encoding part. Understanding the text description has a positive impact

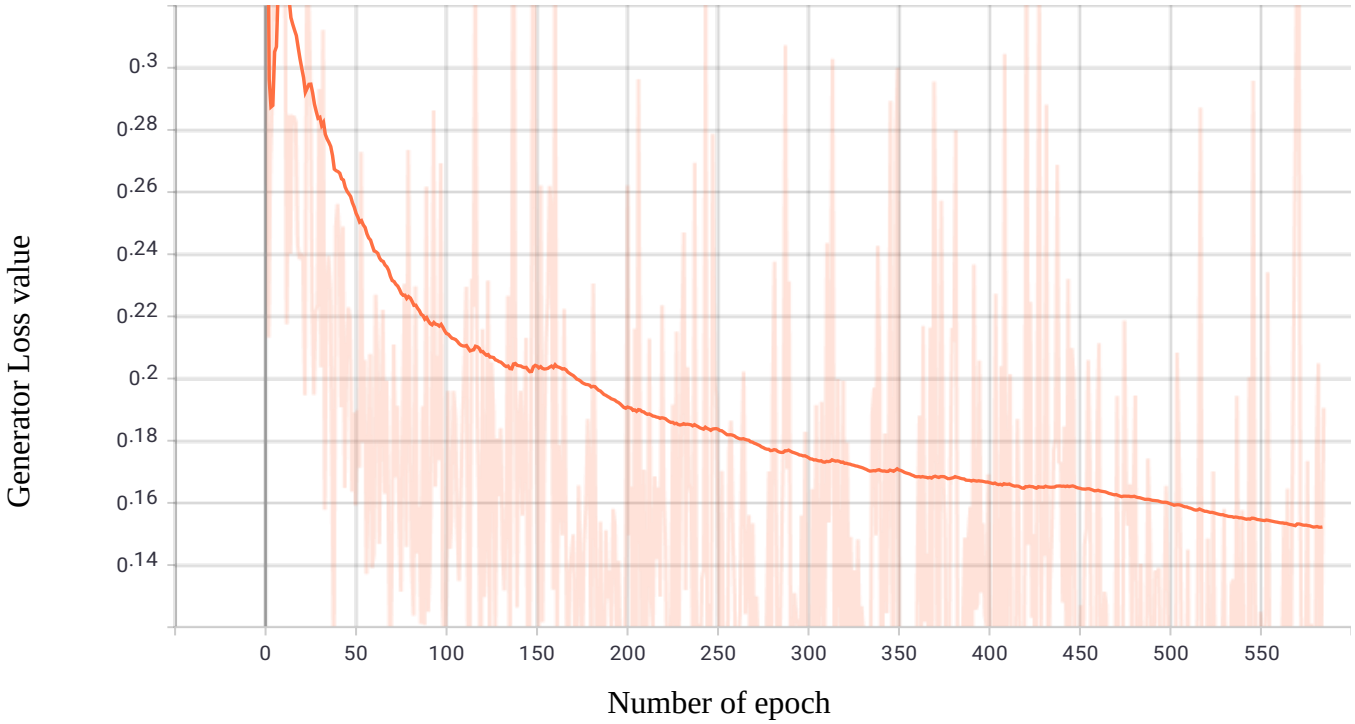
on the creation of images. DF + B gives the best results, but the image quality is still poor, which has also a semantic gap problem. To overcome these problems, DF+SA was carried out.

The previous experiment generated an image only from a sentence level. Still, this kind of trend stacking the whole sentence description in a single vector leads to the lack of the word level importance, to address those issues DF+SA experiment was conducted. DF+SA introduces spatial attention to synthesizes the image in word-level and sentence-level vectors by focusing on the most critical part of the text descriptions. This experiment synthesized a better picture than the previous experiments, but the DF+SA experiment result image has an artifact on the generated image, and still, semantic gab was an issue. To mitigate the above experiment problems, DF+SA+D was conducted by extending the DF+SA experiment. DF+SA+D introduces additional DAMSM loss to solve the semantic issue between the generated image and the text description. DF+SA+D generates a better lookup image, but the rendered image has an artifact, and quality was also an issue. To mitigate this problem, another experiment was conducted DF+SA+CA by adapted the DF+SA experiment.

DF+SA+CA introduces channel-wise attention which is used to focus on only the most relevant channels during the generation of an image. The DF+SA+CA gives a better high-quality image as compared to the previous experiment but still, DF+SA+CA has a semantic gab issue. To solve this problem DF+SA+CA+D experiment was conducted by extending the previous DF+SA+CA experiment. DF+SA+CA+D introducing the DAMSM loss to maintain the semantic gap between the text-image pair. This experiment gives a high-quality image and keeps the semantic gap between the text and the generated image compared to the two baseline works and the other experiments.

All those experiments are conducted with the same dataset in the same environment. Let's see the component of the proposed model, which is the DF+SA+CA+D experiment. DF+SA+CA+D experiment was trained till 500 epochs and each epoch has 736 iterations; the training time of the experiments is five days is taken to reach 500 epochs this is because of the GPU performance. The proposed model training hyperparameters were described in chapter 5 in section 5.8. To stabilize the proposed model's training and to avoid the most common problems of the GAN, which are

vanishing gradient and modal collapse problems, different solutions were used as described in chapter 4. The ideal principle of GAN training is balancing the network of generator and discriminator. So as shown below the discriminator loss and the generator loss are decreasing smoothly. This shows us the two-loss compute one another. The training loss function of the proposed model is demonstrated in the below figures.



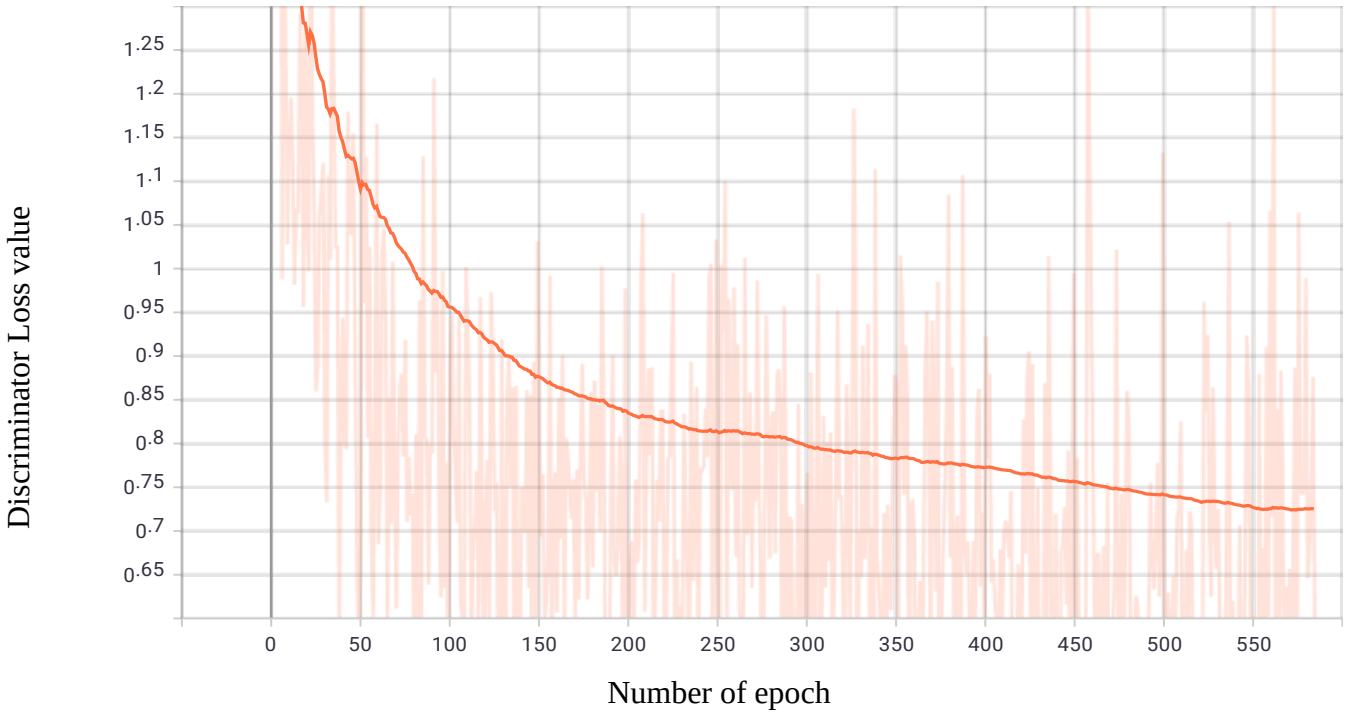


Figure 6. 1. Shows the training loss of the DF+SA+CA+D model

As discussed above, different experiments were conducted, those experiments should be described in qualitative and quantitative measuring metrics compared with the baselines works tested on CUB-200 datasets. Therefore, from different kinds of measuring parameters, this study used the inception score (IS) and human evaluation.

Human evaluation is one of the measuring a parameter of this study. Two questions were prepared to get this measurement parameter. The first question, labeled as Q1, inquires as to which of the generated images is a more natural or realistic image. The second question, labeled as Q2, asks which image is more semantically consistent with the written text description provided. The two questions are distinct in that Q1 inquiries about the appearance of the generated image regardless of its quality, the second Q2 inquiries about the semantic consistency of the generated image with the provided text description. In the next section, let's see the evaluation metrics result of a different experiment.

6.3. Evaluation Metrics

There are different methods to assess the generated images and model, from those inception scores and human evaluation measures were employed. Let look at the inception score evaluation matric result.

6.3.1. Inception Score

These assessment measures were utilized in the majority of prior publications, as well as in this study. Table 6.1 shows the different experiments and their inception score. This score takes all test images and measures how the quality of the image and how the images are diversified. The higher inception score indicates the better quality of the image, and diversified images are generated. Let look at the inception score of different experiments.

Table 6. 1. shows the inception score comparison between all experiments.

Number	**Experiment**	**Model used**	**Inception score**
1	conditional GAN	Conditional GAN	3.62 ±0.07
2	StackGAN-V1	StackGAN-v1	3.70 ±0.04
3	StackGAN-V2	StackGAN-v2	3.84 ±0.06
4	CG	Cycle-GAN	3.70±0.045
5	DF	DFGAN	3.45±0.065
6	DF + C	DFGAN with caption loss	3.60±0.03
7	DF + B	DFGAN with BERT	3.59±0.09
8	DF + SA	DFGAN with spatial Attention	3.61±0.64
9	DF + SA+D	DFGAN with spatial attention plus DAMSM loss	**4.03 ± 0.03**
10	DF + SA + CA	DFGAN with spatial and channel-wise Attention	**4.17± 0.06**
11	DF + SA + CA +D	DFGAN with spatial and channel-wise Attention plus DAMSM loss	**4.31±0.14**

From above table 6.1, The bold values of the inception show the best result in the experiments.

In the above table 6.1 conditional GAN, StackGAN-v1, and StackGAN-v2 the inception score of these models was taken from their original papers because the proposed model outperforms those models without implementation of their paper. As shown in table 6.1 the inception score of the proposed is higher as compared to the other models which indicate the proposed model can synthesis high-quality images and enhance the semantic consistent between the text description

and generated image. let's see another representation of table 6.1 in a graph with epochs versus inception scores.

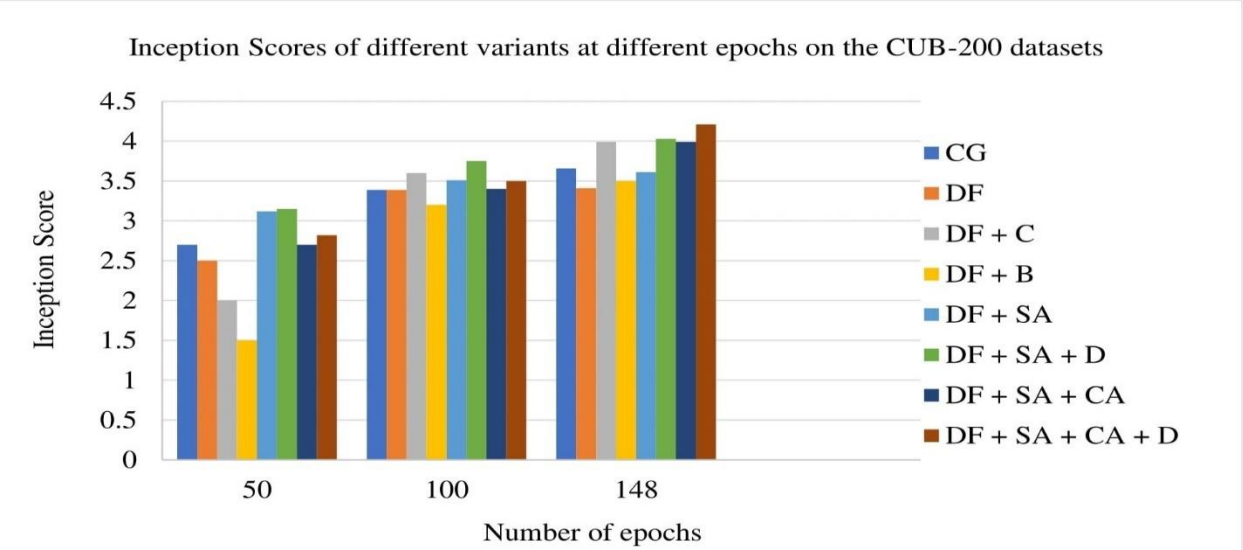


Figure 6. 2. Demonstrate different experiments with their inception score

This figure illustrates different experiments and baseline papers' inception scores at different epochs. This graph illustrates each model and its corresponding inception score vs epoch. The average epoch value was selected to show the comparison of the different experiments, this average epoche was chosen because some methods generate a black image after a certain epoch.

Next, let’s see how each model or experiment synthesized an image from the given text description. As shown in table 6.2, the first row represents the text description, the second row represents the ground truth image, the third row represents the result of the CG experiments, the Fourth row represents the result of the DF experiments, the fifth row represents the result of the DF+C experiments, the sixth row represents the result of the DF+SA+CA+D experiments.

Table 6. 2. demonstrate text to image translation results

Text description	A small bird with a red head, breast, and belly, and black wing.	a bird with black head, white throat, nape and breast, black ring around the throat and black wings and back	This bird has a wing that is brown and has a white belly.

Ground truth			
CG			
DF			
DF + C			

DF + SA + CA + D			

As shown in table 6.2. the proposed model shows a high-quality image and a better semantic coherence between the synthesized image and the text description. First, let's see the first column comparison. In this column, the CG experiments synthesized an image from the given text description and gave a better image, but the image quality is low and lacks semantic consistency. Let's look at the DF, DF experiments synthesized an image from the given text description and gives a better image. DF experiment achieves a better result as compared to the CG experiment, but it generates less quality image and semantic consistency is still an issue. In addition, during generation, the part of birds might be deformed or missed.

The DF+C experiments synthesized an image from the given text description, gave a better image, and showed how its generated image is some closer to the ground truth. But it has a quality issue. DF + SA +CA + D experiment effectively synthesized an image from the given text description and provided a better high-quality image. Therefore, among all the above experiments the DF + SA +CA + D experiment synthesized an image effectively from the text description (i.e. it maintain the semantic consistency between text-image pairs) and generate a high-quality image by preserving the contents of the image. Next, let’s look at what was human evaluation result of the proposed modal compare with the baseline work of DF experiment.

6.3.2. Human Evaluation Results for Q1

This section goes deep into Q1's human evaluation result. To do so, individuals who were interested filled out Google form questionnaires. This questionary asked the user to answer which

generated image is a more realistic or natural image. let's see the sample question and how those volunteers answer the question.[2]

1) carefully look at the following images, choose one of them which more realistic image or more natural(የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ፣ ከመካከላቸው አንዱን የበለጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።) *

Mark only one oval.

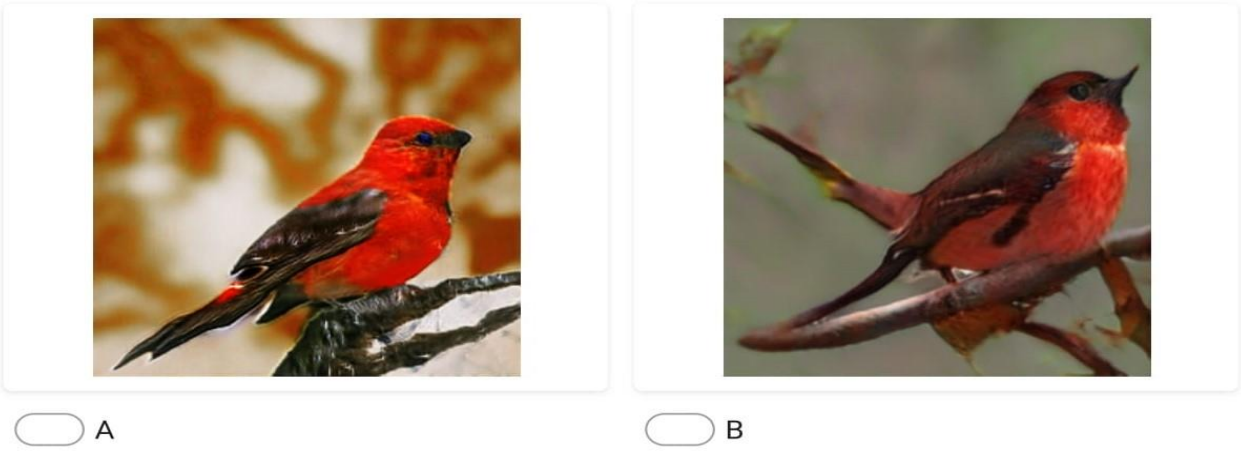


Figure 6. 3. Sample question from Q1

This was a sample question from Q1. This inquiry was replied to by a volunteer individual and score the accompanying outcome. The first image was the consequence of the proposed model and the subsequent image was the baseline work DFGAN. Among 30 users 73.3% answer the first image and 26.7% answer the subsequent image as shown in figure 6.4. this demonstrates that the proposed model creates a more realistic image and sensible than the baseline work.

[2] Human evaluation form for Q1 found at: https://docs.google.com/forms/d/e/1FAIpQLSeNLmP8CTBhsk21gDPMvlNDDc821ZmFFiN-XqmIbCZZHeZPA/viewform

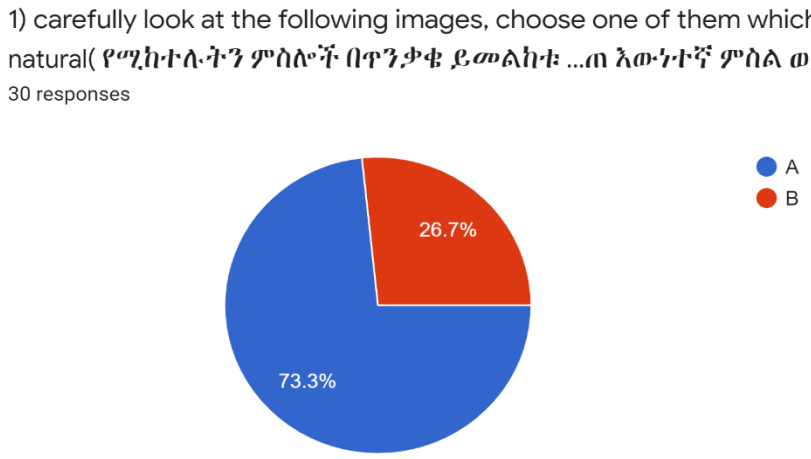


Figure 6. 4. Show the result of the sample Q1[3]

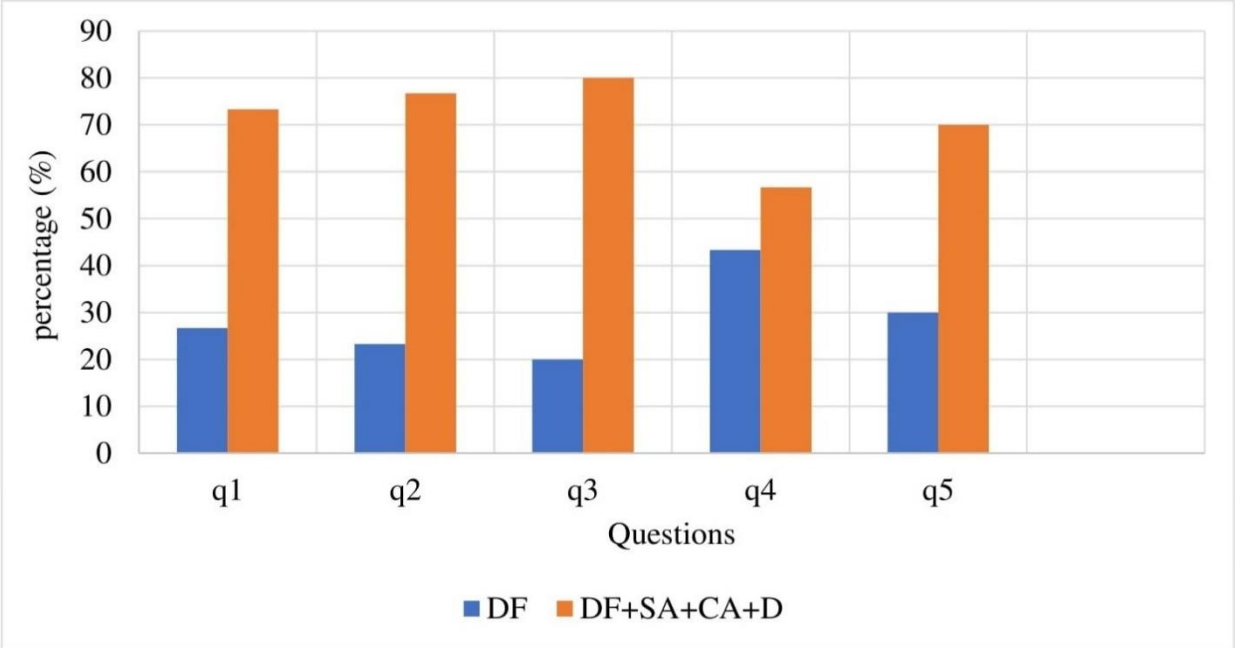


Figure 6. 5. Show all five questions in Q1 and their result in percentage

The graph demonstrates all questions and the correspondence user response in percentage. The graph shows that the proposed model scores a higher value as compared to the baseline work.

Therefore, this indicates the proposed model synthesized a more realistic image than the baseline work. The details of human evaluation are discussed in the appendix section of a Q1 user study.

6.3.3. Human Evaluation Results for Q2

This section goes deep into Q2's human evaluation result. To do so, individuals who were interested filled out Google form questionnaires. This questionary asked the user to answer a question that does the generated image is more semantically consistent for the given text description? let's see the sample question and how those volunteers answer the question based on the given text description.[4]

Figure 6. 6. Show the sample question in Q2

This was a sample question from Q2. the users are expected to answer the question based on which image was more synthesized from the text description. The first image was the proposed model result and the second image was the result of the baseline work DFGAN. From 30 users 83.3% of the user answer the first image and 16.7% answer the second image as shown in figure 6.7. this result demonstrates the proposed model was more synthesized from the given text description.

[4] Human evaluation form for Q2 found at:
https://docs.google.com/forms/d/e/1FAIpQLSfiLF4AvANBfP_22cu30cYxPgatohjsYjOnDIV95h3bPxC-3A/viewform

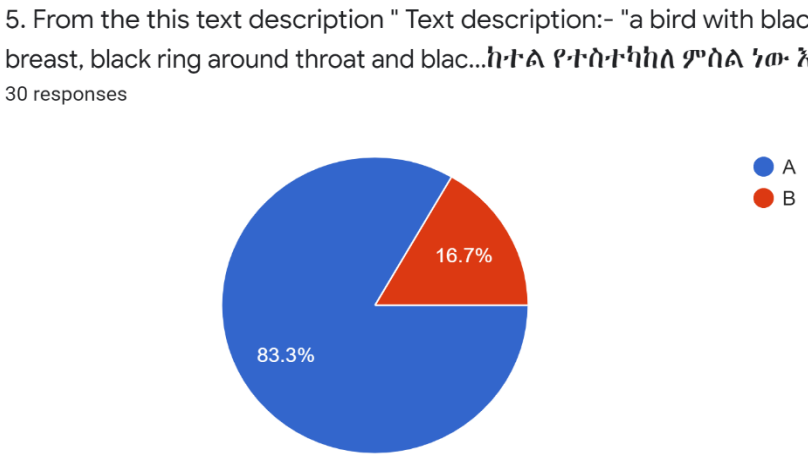


Figure 6. 7. Shows the result of the Q2

The details of human evaluation are discussed in the appendix section of a Q2 user study.

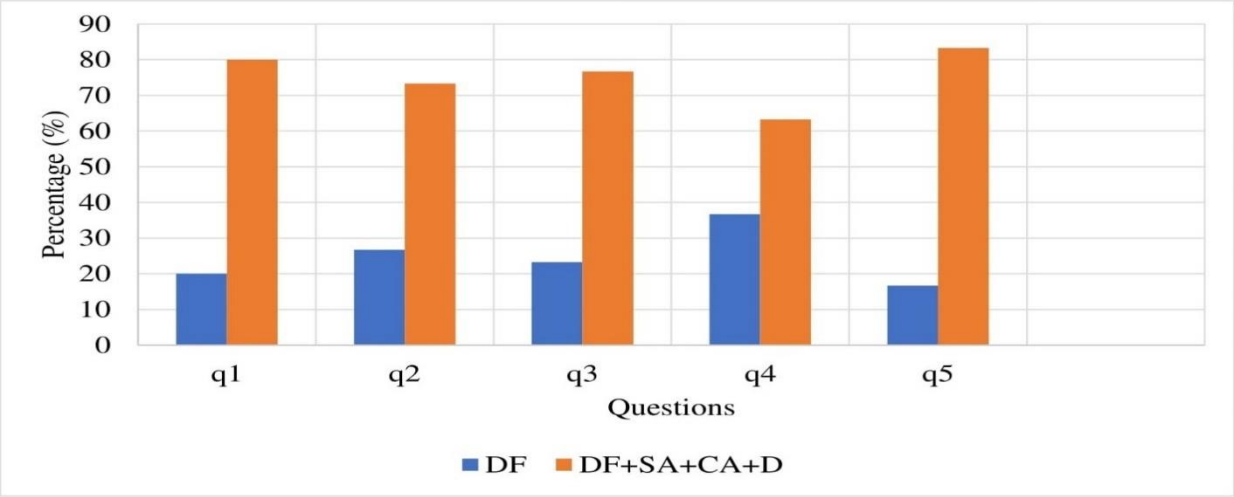


Figure 6. 8. Demonstrate all questions in Q2 and their percentage

The graph demonstrates all questions and the correspondence user response in percentage. The graph shows that the proposed model scores a higher value as compared to the baseline work. Therefore, this indicates the proposed model enhances the semantic consistency between the generated image and the text information than the baseline work.

6.4. Results Discussion on Text-to-Image Translation

This proposed model is compared with several states of the art methods such as conditional GAN, StackGAN-v1, StackGAN-v2, and CycleGAN. Those methods used a stack of GAN and achieved a remarkable result in text-to-image translation but the proposed model used a single pair of generators and discriminators to generate a high-quality image. Let's discuss the qualitative and the quantitative results in detail.

6.4.1. Quantitative Results

Several experiments were conducted to come up with the final proposed model. The first experiment was DF + C, which introduces caption loss(cycle loss) which used to enhance the semantic consistency of text-image by calculating the original input text description and the reconstructed text from the generated image, has the lowest inception score which is 3.60±0.03 as compared to CG experiments, but has high inception score as compared to DF experiment, DF + B experiment also has the lowest inception score which is 3.59±0.09 as compared to CG experiments but has the highest inception score as compared to DF experiment. DF + B experiment introduces BERT, which is one of the transformer methods that is used to enhance the text encoding method. Those two experiments only used a sentence-level vector to synthesized images, but this lacks fine-grained details of word-level information. To mitigate these problems some experiments were conducted as described below. The upcoming experiments have synthesized an image from both the word-level vector which is used to generate fine-grain detail of an image and the sentence-level vector.

This study evaluates the effectiveness of spatial attention, channel-wise attention, and DAMSM loss in different experiments. First, let look at the DF+SA experiment that introduced the spatial attention mechanism to focus on the most relevant words in the text description. Compared with the baseline work, DF + SA improves the inception score from 3.45±0.065 to 3.61±0.64 by adding spatial attention in six blocks of a generator. The next experiment was DF + SA + D, which extends the DF+SA experiment. DF + SA + D was introduced DAMSM loss to improve the generated image and text description semantic consistency. DF + SA + D improves the inception score from 3.45±0.065 to 4.03 ± 0.03 by adding DAMSM loss.

The next experiment was DF + SA + CA, which extends the DF+SA experiment. DF + SA + CA introduced channel-wise attention used to focus the most relevant channel. Due to this reason, the

quality of the image is enhanced. DF + SA + CA improves the inception score from 3.45±0.065 to 4.17± 0.06 by adding both spatial and channel-wise attention in a six-block of a generator. The last experiment was DF+SA+CA+D, which extends the DF+SA+CA experiment. DF+SA+CA+D introduced the DAMSM loss to enhance the semantic consistency between the text description and the generated image. DF+SA+CA+D improves the inception score from 3.45±0.065 to 4.31±0.14 by adding channel-wise attention in six-block.

The quantitative evaluation result shows the effectiveness of the proposed model, which can generate high-quality images and maintain the semantic consistency between the text description and the generated images. The proposed model improves the baseline work CycleGAN from 3.70±0.045 to 4.31±0.14 and also enhances the baseline work of DFGAN from 3.45±0.065 to 4.31±0.14. This indicates that the identified components such as spatial attention, channel-wise attention, and DAMSM loss have a great influence on the improvement of the quality of the generated image and on enhancing the semantic consistency between the text-image domain.

As shown in table 6.1, the proposed model achieves the highest inception score when compared with other states of the art methods include conditional GAN, StackGAN-v1, and StackGAN-v2. The proposed model improves the inception score from 3.45±0.065 to 4.31±0.14. The proposed model's quantitative result indicates that it can generate a more realistic image with a better text-image semantic consistency.

Table 6. 3. Summary of the effectiveness of components

Components					
	DFGAN	Spatial attention	Channel-wise attention	DAMSM Loss	IS ⬆
1	✓	-	-	-	3.45±0.065
2	✓	✓	-	-	3.61±0.64
3	✓	✓	✓	-	4.17±0.06
4	✓	✓	✓	✓	**4.31±0.14**

Table 6.3, demonstrates the summary report of the effectiveness of each component and indicates how each component improves the proposed model by controlling each component.

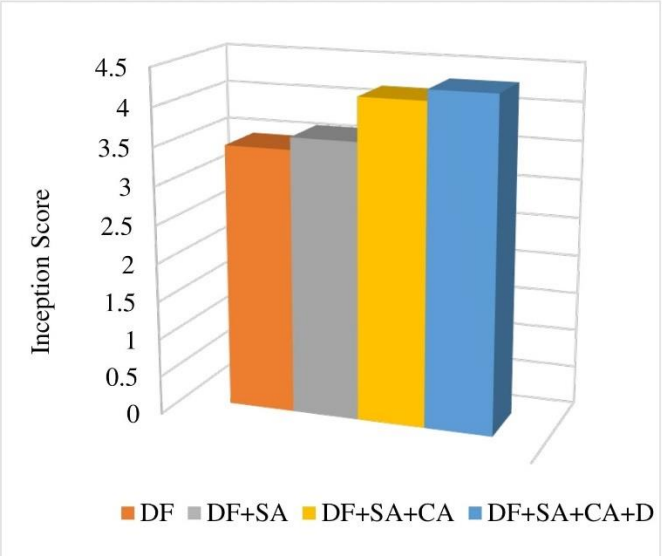


Figure 6. 9. inception score of different models

Figure 6.9, demonstrates how each of the novel components effectively improves models.

6.4.2. Qualitative Results

The human evaluation metrics were used in two forms Q1 and Q2 to assess the proposed model and the baseline works image how the image is realistic and the semantic consistency between the text description and the generated image. So first let's discuss the first human evaluation question Q1 results.

Discussion on Human evaluation Results Q1

In this Q1 around five questions were given to the users and each question contains the proposed model image and the baseline work image. Then the users have expected to respond to which image was realistic or natural. Figure 6.5 shows the comparison of the proposed model and the baseline work for each question. The average calculation was used for each model as follow

$$Average\ Model = \frac{1}{q_t}\sum_{k=1}^{5} Q_k \qquad eq.(6.1)$$

Where $q_t = 5$ is the total question and Q_k represents each question percentage.

The average percentage of the value of the DF baseline work and the proposed model is represented in table 6.4.

Table 6.4. shows the average value of each model

Models	Q1 in average percentage (%)
DFGAN	24.68
proposed model	75.32

From table 6.4, the human evaluation analysis of the proposed model averagely increases by 50.64 percent according to as compare to the baseline work. Therefore, This indicates the proposed model has a high-quality image as compared to the DF baseline work.

Discussion on Human evaluation Results Q2

As shown in figure 6.8, each question was answered by the user and the result was presented in percentage. This human evaluation tries to measure how the models generated images were semantically consistent with the text description. The average percentage was calculated for each model according to the user response. The average was calculated using equation 6.1.

Table 6.5. shows the average value of each model

Models	Q2 in average percentage (%)
DFGAN	28.66
proposed model	71.34

From table 6.5, for this, human evaluation analysis, the proposed model averagely increases by 42.68 percent compared with baseline work. Therefore, the result shows the proposed model maintains the semantic coherence of the created image and the text information

6.4.3. Research Question Discussion

This research answers each research question that was raised in the first chapter. This study answers each question as follow:

Answer for Q1: The better constraints were identified for enhancing the image quality and improving the semantic consistency of text-image pairs such as spatial and channel-wise attention were identified for enhancing the image quality and DAMSM loss was identified for maintaining the semantic gap between the generated image and the textual description.

Answer for Q2: Spatial and channel-wise attention was added to the model, the experimental result shows a better result regarding the image quality. The result of the inception score and the human evaluation shows the effectiveness of those attentions for enhancing the quality of the image. To show the effect of each attention separate experiments were conducted so based on the inception score methods as shown in table 6.3, the spatial attention improved the result from 3.45±0.065 to 3.61±0.64. The channel-wise attention improves the results from 3.61±0.64 to 4.17±0.06. Based on the first human evaluation form (Q1) as shown in table 6.4, the quality of the proposed model exceeds the DF experiments by 50.64. From this answer, this study generalized that the identified attention mechanisms have a great influence on quality enhancement.

Answer for Q3: DAMSM loss was added to the proposed model to saw the effect of this loss, the experimental result shows a better result regarding enhancing the semantic gap between the generated image and the textual description. Based on the inception analysis as shown in table 6.3, the DAMSM loss improves the result from 4.17±0.06 to 4.31±0.14. Based on the second human evaluation (Q2) as shown in table 6.5, the proposed model exceeds the DF experiments by 42.68. From this answer, this study generalized that the identified DAMSM loss shows, DAMSM loss has a positive impact on the enhancement of sematic consistency between the text and the fake image.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

One of the uses of Generative Adversarial Networks is text to images translation. The text domain and image domains are different that makes the translation difficult. Different approaches were used to tackle these problems. But those approaches used a stack of GAN to generated high-quality images and used another network to ensure that the obtained image is semantically consistent with the text description. Those approaches have problems when the first image is poorly synthesized, subsequent generators will struggle to improve it to acceptable image quality. As a result, it has an impact on the generative model's convergence and stability, as well as the quality of early generating images. Other methodologies utilize a single generator and discriminator, but those approaches used sentence-level information to synthesized an image, so stacking whole information in a single vector, leads to a lack the relevance of word-level information. Therefore, those approaches have the issue of generating high-quality images and semantic consistency between the synthesized image and the text information. To overcome those problems this study proposed a single pair of generators and discriminator and certain constraints.

The purpose of this work was to improve the semantic coherence of the synthesized image and the text information and to enhance the quality of the rendered image by adding certain constraints in the generator and discriminator network. This study introduces the spatial and channel-wise attention mechanisms and deep attention multimodal similarity model loss and spectral normalization to the baseline work. Indeed, those constraints make the proposed model very mindful of generating high-quality images and maintain the semantic consistency between the text description and the generated image.

The semantic gap between the generated images and the written description is one of the most difficult aspects of text-to-image translation, so this thesis introduces a DAMSM loss constraint to minimize the semantic gap between the text description and the generated image. DAMSM loss is utilized encoder-decoder principle. The encoder is used to extract the text description into word and sentence level vectors. The Decoder takes the generated image as input and extracts the feature of the generated image in local and global features of the image. The DAMSM loss is used to

compute the semantic distance between the generated image and the text description at word level and sentence level. This loss function helps the proposed model to maintain the text-image semantic gap.

Another challenge of text-to-image translation is generating a realistic high-quality image, so this study introduces a spatial and channel-wise attention mechanism to generate a high-quality image. the spatial attention is used to focus the relevant word information during the generation of the image. channel-wise attention is used to further improve the image quality created, by focusing on the most important relevant channel information of the image by ignoring irrelevant channels. Spatial and channel-wise attention mechanism helps the proposed model to generate the realistic and high-quality image.

Compare the proposed model with the previous state of art methods like conditional GAN, satackGAN-v1and StackGAN-v2, the qualitative and quantitative experimental results show, Those works have the lowest inception score. Comparing with the baseline works CG and DF, the qualitative and quantitative experimental results demonstrate the effectiveness of the proposed model. Based on the inception score analysis the proposed model improves the baseline work CG from 3.70±0.045 to 4.31±0.14 and also improves the baseline work of DF from 3.45±0.065 to 4.31±0.14. The average human evaluation of the proposed model improves the DF by 46.66 percent. This work concludes that adding spatial attention and channel-wise attention constraints to text-to-image translation does improve the quality of the generated image. It has been shown by the inception score and the human evaluation metrics of experimental results. deep attentional multimodal similarity model loss constraints also have a positive impact on improving the semantic consistency between the generated image the text description at word and sentence levels. The spectral normalization is also used in stabilizing the training of the discriminator.

7.2. Future Work

This thesis doesn't come up with limitations. As an observer in the experiments, the model depends on the performance of the text encoder and the generator network, and the discriminator to synthesize a high-quality image. To further improve the generated image quality, and the semantic consistency of the text-image pair which is the text encoder, the generator, and the discriminator should be improved. In this thesis works the bi-LSTM text encoder was used to encode the text description into word-level and sentence-level vectors. Understanding the text description through

effective techniques helps the generator to generate a better image, so this thesis recommends that better transfer text encoder techniques are like the Robustly optimized BERT approach (Roberta), which is the retraining of BERT with improved training methodology.

This thesis used inception v3 for image encoding that was used to extract the feature of the generated image. Understanding the feature of the generating image is an important thing for further computation, Thesis recommends that using better feature extraction models than inception v3 in order to enhance the performance of the model. At the generator stage, this work is trying to generate high-quality images and semantically consistent images by introducing an attention mechanism but for further improvement of the quality of the generated image, this thesis recommends that utilizing a memory mechanism and segmentation mask used to enhance the quality of images. The above-recommended models work effectively if the powerful GPU is available.

This thesis has a limitation of generating the image background to solve this issue, this study recommends the multi conditional Gan concepts by extract the feature of the background image separately and feed it to the generator as one condition which helps to consider the image background during the generation of the image.

SPECIAL ACKNOWLEDGMENTS

This research project is funded by Adama Science and Technology University under the grant number:

ASTU/SM-R/236/21

Adama, Ethiopia

REFERENCES

A Review of Generative Adversarial Networks — part 1 | by Roy Ganz | Analytics Vidhya | Medium. (n.d.). Retrieved August 19, 2021, from https://medium.com/analytics-vidhya/a-review-of-generative-adversarial-networks-part-1-a3e5757a3dc2

Ahmadi, S. (2018). Attention-based Encoder-Decoder Networks for Spelling and Grammatical Error Correction. *ArXiv*.

Automatic Image Captioning Using Deep Learning. (n.d.). Retrieved August 19, 2021, from https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/?__cf_chl_captcha_tk__=pmd_spQvSXMbP0_nWBa6yIzL1RQ5gate75cpdP90L9r0bWs-1629399820-0-gqNtZGzNAxCjcnBszQ-R

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Barrios, S., Buldain, D., Comech, M. P., Gilbert, I., & Orue, I. (2019). Partial discharge classification using deep learning methods - Survey of recent progress. *Energies*, *12*(13), 1–16. https://doi.org/10.3390/en12132485

Bekesiene, S., Smaliukiene, R., & Vaicaitiene, R. (2021). Using artificial neural networks in predicting the level of stress among military conscripts. *Mathematics*, *9*(6). https://doi.org/10.3390/math9060626

Bi-LSTM. What is a neural network? Just like our… | by Raghav Aggarwal | Medium. (n.d.). Retrieved August 19, 2021, from https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0

Bodnar, C. (2018). Text to Image Synthesis Using Generative Adversarial Networks. *ArXiv*. https://doi.org/10.13140/RG.2.2.35817.39523

Cai, Y., Wang, X., Yu, Z., Li, F., Xu, P., Li, Y., & Li, L. (2019). Dualattn-GAN: Text to Image Synthesis with Dual Attentional Generative Adversarial Network. *IEEE Access*, *7*, 183706–183716. https://doi.org/10.1109/ACCESS.2019.2958864

Chandra, R., Goyal, S., & Gupta, R. (2021). Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction. *IEEE Access*, *9*(May), 83105–83123. https://doi.org/10.1109/ACCESS.2021.3085085

Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., & Chua, T. S. (2017). SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 6298–6306. https://doi.org/10.1109/CVPR.2017.667

Cheng, J., Wu, F., Tian, Y., Wang, L., & Tao, D. (2020). Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *d*, 10908–10917. https://doi.org/10.1109/CVPR42600.2020.01092

Christie's just sold its first piece of art generated by AI. - Vox. (n.d.). Retrieved August 8, 2021, from https://www.vox.com/the-goods/2018/10/29/18038946/art-algorithm

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., & Courville, A. (2017). Modulating early visual processing by language. *Advances in Neural Information Processing Systems*, *2017-Decem*, 6595–6605.

Dong, H., Zhang, J., Mcilwraith, D., & Guo, Y. (2017). I2T2I: LEARNING TEXT TO IMAGE SYNTHESIS WITH TEXTUAL DATA AUGMENTATION Hao Dong, Jingqing Zhang, Douglas McIlwraith, Yike Guo Data Science Institute, Imperial College London. *IEEE Conference Proceeding*. https://ieeexplore.ieee.org/abstract/document/8296635/

El-Nouby, A., Sharma, S., Schulz, H., Hjelm, R. D., Asri, L. El, Kahou, S. E., Bengio, Y., & Taylor, G. (2019). Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 10303–10311. https://doi.org/10.1109/ICCV.2019.01040

For, R. (2005). E Rror R Esilience V Ideo T Ranscoding for. *Ieee Wireless Communications*, *August*, 14–21.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gorti, S. K., & Ma, J. (2018). Text-to-image-to-text translation using cycle consistent adversarial networks. *ArXiv*.

Graves, A. (n.d.). *Framewise Phoneme Classification with Bidirectional LSTM Networks*.

He, X., Peng, Y., & Zhao, J. (2019). Fast Fine-Grained Image Classification via Weakly Supervised

Discriminative Localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(5), 1394–1407. https://doi.org/10.1109/TCSVT.2018.2834480

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 6627–6638.

Hinz, T., Heinrich, S., & Wermter, S. (2019a). Semantic object accuracy for generative text-to-image synthesis. *ArXiv*, *14*(8). https://doi.org/10.1109/tpami.2020.3021209

Hinz, T., Heinrich, S., & Wermter, S. (2019b). Semantic object accuracy for generative text-to-image synthesis. *ArXiv*, *14*(8), 1–22. https://doi.org/10.1109/tpami.2020.3021209

Hochreiter, S. (1997). *Long Short-Term Memory*. *1780*, 1735–1780.

Huiskes, M. J., Thomee, B., & Lew, M. S. (2010). New trends and ideas in visual concept detection: The MIR flickr retrieval evaluation initiative. *MIR 2010 - Proceedings of the 2010 ACM SIGMM International Conference on Multimedia Information Retrieval*, 527–536. https://doi.org/10.1145/1743384.1743475

Jain, P., & Jayaswal, T. (2020). Generative adversarial training and its utilization for text to image generation: A survey and analysis. *Journal of Critical Reviews*, *7*(8), 1455–1463. https://doi.org/10.31838/jcr.07.08.290

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Lecun, Y., Bottou, L., Bengio, Y., & Ha, P. (1998). LeNet. *Proceedings of the IEEE, November*, 1–46.

Li, B., Qi, X., Lukasiewicz, T., & Torr, P. H. S. (2019). *Controllable Text-to-Image Generation*. *NeurIPS*.

Li, L., Sun, Y., Hu, F., Zhou, T., Xi, X., & Ren, J. (2020). Text to Realistic Image Generation with Attentional Concatenation Generative Adversarial Networks. *Discrete Dynamics in Nature and Society*, *2020*. https://doi.org/10.1155/2020/6452536

Mikolov, T. (2014). *Distributed Representations of Words and Phrases and their Compositionality arXiv : 1310 . 4546v1 [cs . CL] 16 Oct 2013*. *October*.

Monica, K., & Rao, D. R. (2020). Text to Image Translation using Cycle GAN. *International Journal of Engineering and Advanced Technology*, *9*(4), 1294–1297. https://doi.org/10.35940/ijeat.d8703.049420

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 689–696.

Pedamonti, D. (2018). *Comparison of non-linear activation functions for deep neural networks on MNIST classification task*. *3*. http://arxiv.org/abs/1804.02763

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe : Global Vectors for Word Representation*. 1532–1543.

Qiao, T., Zhang, J., Xu, D., & Tao, D. (2019a). Learn, imagine and create: Text-to-image generation from prior knowledge. *Advances in Neural Information Processing Systems*, *32*(NeurIPS), 1–11.

Qiao, T., Zhang, J., Xu, D., & Tao, D. (2019b). MirrorGAN: Learning text-to-image generation by redescription. *ArXiv*.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. *33rd International Conference on Machine Learning, ICML 2016*, *3*, 1681–1690.

Salehi, P., Chalechale, A., & Taghizadeh, M. (2020). Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments. *ArXiv*.

Sangeetha, V., & Prasad, K. J. R. (2006). Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2- carbaldehydes. *Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry*, *45*(8), 1951–1954. https://doi.org/10.1002/chin.200650130

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Sharma, S., Suhubdy, D., Michalski, V., Kahou, S. E., & Bengio, Y. (2018). ChatPainter: Improving text to image generation using dialogue. *ArXiv*.

Sherstinsky, A. (2020). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. *404*(March), 1–43.

Sohn, K., Shang, W., & Lee, H. (n.d.). *Improved Multimodal Deep Learning with Variation of Information*. 1–9.

Srivastava, N., & Salakhutdinov, R. (2014). Multimodal learning with Deep Boltzmann Machines. *Journal of Machine Learning Research*, *15*, 2949–2980.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *4*(January), 3104–3112.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Tao, M., Tang, H., Wu, S., Sebe, N., Jing, X.-Y., Wu, F., & Bao, B. (2020). *DF-GAN: Deep Fusion Generative Adversarial Networks for Text-to-Image Synthesis*. *1*, 1–13. http://arxiv.org/abs/2008.05865

Tsue, T., Li, J., & Sen, S. (2020). Cycle Text-to-Image GAN with BERT. *ArXiv*.

Van Den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *33rd International Conference on Machine Learning, ICML 2016*, *4*, 2611–2620.

Welling, M. (n.d.). *Auto-Encoding Variational Bayes arXiv : 1312 . 6114v10 [stat . ML] 1 May 2014*. *Ml*, 1–14.

What is Microsoft Visio® | Lucidchart. (n.d.). Retrieved August 8, 2021, from https://www.lucidchart.com/pages/what-is-microsoft-visio

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & He, X. (n.d.). *AttnGAN : Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks*. 1316–1324.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & He, X. (2017a). AttnGAN : Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. *ArXiv*, 1316–1324.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & He, X. (2017b). AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. *ArXiv*, 1316–1324.

Yang, X. (2020). An Overview of the Attention Mechanisms in Computer Vision. *Journal of Physics: Conference Series*, *1693*(1). https://doi.org/10.1088/1742-6596/1693/1/012173

Zakraoui, J., Saleh, M., & Al Ja'am, J. (2019). Text-to-picture tools, systems, and approaches: a survey. *Multimedia Tools and Applications*, *78*(16), 22833–22859. https://doi.org/10.1007/s11042-019-7541-4

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *ArXiv*, 5907–5915.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2019). StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(8), 1947–1962. https://doi.org/10.1109/TPAMI.2018.2856256

Zhu, M., Pan, P., Chen, W., & Yang, Y. (2019). DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 5795–5803. https://doi.org/10.1109/CVPR.2019.00595

Zhu, Y., Zhang, L., Fan, J., & Han, S. (2007). Neural basis of cultural influence on self-representation. *NeuroImage*, *34*(3), 1310–1316. https://doi.org/10.1016/j.neuroimage.2006.08.047

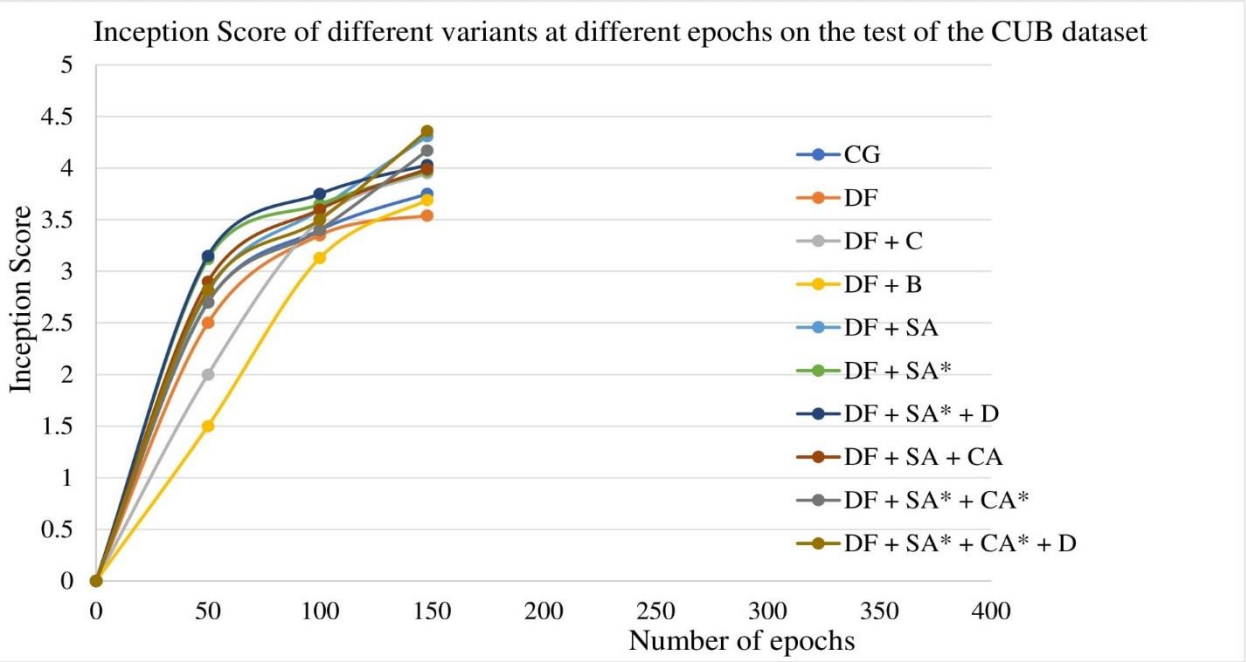
Zia, T., Arif, S., Murtaza, S., & Ullah, M. A. (2020). Text-To-Image Generation With Attention Based Recurrent Neural Networks. *ArXiv*.

APPENDIXES

Appendix A: Ablation Study

This ablation study has been done to show the effectiveness of spatial attention, channel-wise attention, and the DAMSM loss and come up with the number of attention mechanisms added in the generator block. The ablation study tries to answers two questions. The first question was, in which combination of those components yields a high-quality image and enhances the semantic consistency between the generated image and the text description? The second question was, what is the effective number of attention mechanisms that work to enhance the quality of the generated image? let's look at the explanation of the above question.

Around eight experiments were conducted with different combinations of each component. the experiment is DF + SA is used spatial attention and add it in three blocks of the UPBlock and it improves the DF experiment from 3.45±0.065 to 3.97±0.32. DF +SA* was conducted inspired by the result of the DF+SA and in this experiment, the spatial attention was added in 6 black this improves the DF experiment from 3.45±0.065 to 3.61±0.64. DF+SA+CA introduced channel-Wise attention by extending DF+SA and channel-wise attention was added in three blocks only DF+SA+CA and improves the DF experiment from 3.45±0.065 to 3.99 ± 0.03. DF+SA*+CA* was done by extending the DF+SA+CA experiment and the channel-wise attention was added in six UPBlocks and improves the DF experiment from 3.45±0.065 to 4.17± 0.06. DF+SA*+CA*+D was done by extending the DF+SA*+CA* and DAMSM loss was added and improves the DF experiment from 3.45±0.065 to 4.31±0.14. Finally, this study shows the effectiveness of the spatial attention, channel-wise attention, and DAMSM loss, and adding those components in six blocks of the generator gives a better quality of the image and maintains the semantic consistency between the generated image and the text description in both sentences and word-level rather than adding in three blocks of the generator. Both attention mechanisms work better when both were added in all six blocks rather than adding in the three blcoks.



The above figure shows different experiments with different models

Appendix B: Result on Different epochs

Training at 50 epochs

Training at 154 epochs

Training at 404 epochs

Training at 493 epochs

Appendix C: Result on Different Models

Text Description	This small bird has greenish feathers over most of its body, a small beak, black wing bars, and black tarsus and feet.	A colorful bird with a red head, face, and breast, and grey, black and tan secondaries.	This is a small bird with a very long tail and colors ranging from grey to white.	This is a small grey colored bird with a long tail
Ground Truth				
CG				


DF
DF+C
DF+SA+CA+D

Appendix D: Sample Code

Train Generator Network

```
class NetG(nn.Module):
    def __init__(self, ngf=64, nz=100):
        super(NetG, self).__init__()
        self.ngf = ngf
        self.fc = nn.Linear(nz, ngf*8*4*4)
        self.block0 = G_Block(ngf * 8, ngf * 8)#4x4
        self.block1 = G_Block(ngf * 8, ngf * 8)#4x4
        self.block2 = G_Block(ngf * 8, ngf * 8)#8x8
        self.block3 = G_Block(ngf * 8, ngf * 8)#16x16
        self.block4 = G_Block(ngf * 8, ngf * 4)#32x32
        self.block5 = G_Block(ngf * 4, ngf * 2)#64x64
        self.block6 = G_Block(ngf * 2, ngf * 1)#128x128

        self.conv_img = nn.Sequential(
            nn.LeakyReLU(0.2,inplace=True),
            nn.Conv2d(ngf, 3, 3, 1, 1),
            nn.Tanh(),
        )
        self.att0 = SPATIAL_NET(ngf, 256, 0)
        self.channel_att0 = CHANNEL_NET(ngf, 256, 0)

        self.att1 = SPATIAL_NET(ngf, 256, 1)
        self.channel_att1 = CHANNEL_NET(ngf, 256, 1)

        self.att2 = SPATIAL_NET(ngf, 256, 2)
        self.channel_att2 = CHANNEL_NET(ngf, 256, 2)

        self.att3 = SPATIAL_NET(ngf, 256, 3)
        self.channel_att3 = CHANNEL_NET(ngf, 256, 3)

        self.att4 = SPATIAL_NET(ngf, 256, 4)
        self.channel_att4 = CHANNEL_NET(ngf, 256, 4)
```

```
        self.att5 = SPATIAL_NET(ngf, 256, 5)
        self.channel_att5 = CHANNEL_NET(ngf, 256, 5)

        self.attChannelerb0 = nn.Conv2d(512, 256, kernel_size=1, strid
e=1,
                     padding=0, bias=False)
        self.attChannelerb1 = nn.Conv2d(512, 256, kernel_size=1, strid
e=1,
                     padding=0, bias=False)
        self.attChannelerb2 = nn.Conv2d(512, 256, kernel_size=1, strid
e=1,
                     padding=0, bias=False)
        self.attChannelerb3 = nn.Conv2d(512, 256, kernel_size=1, strid
e=1,
                     padding=0, bias=False)
        self.attChannelerb4 = nn.Conv2d(256, 128, kernel_size=1, strid
e=1,
                     padding=0, bias=False)
        self.attChannelerb5 = nn.Conv2d(128, 64, kernel_size=1, stride
=1,
                     padding=0, bias=False)

    def forward(self, x, c, word_embs, mask):
        out = self.fc(x)
        out = out.view(x.size(0), 8*self.ngf, 4, 4)
        out = self.block0(out,c)

        self.att0.applyMask(mask)
        c_code, att = self.att0(out, word_embs)
        c_code_channel, att_channel = self.channel_att0(c_code, word_e
mbs, out.size(2), out.size(3))

class affine(nn.Module):
```

```
def __init__(self, num_features):
    super(affine, self).__init__()

    self.fc_gamma = nn.Sequential(OrderedDict([
        ('linear1',nn.Linear(256, 256)),
        ('relu1',nn.ReLU(inplace=True)),
        ('linear2',nn.Linear(256, num_features)),
        ]))
    self.fc_beta = nn.Sequential(OrderedDict([
        ('linear1',nn.Linear(256, 256)),
        ('relu1',nn.ReLU(inplace=True)),
        ('linear2',nn.Linear(256, num_features)),
        ]))
    self._initialize()

def _initialize(self):
    nn.init.zeros_(self.fc_gamma.linear2.weight.data)
    nn.init.ones_(self.fc_gamma.linear2.bias.data)
    nn.init.zeros_(self.fc_beta.linear2.weight.data)
    nn.init.zeros_(self.fc_beta.linear2.bias.data)

def forward(self, x, y=None):

    weight = self.fc_gamma(y)
    bias = self.fc_beta(y)

    if weight.dim() == 1:
        weight = weight.unsqueeze(0)
    if bias.dim() == 1:
        bias = bias.unsqueeze(0)

    size = x.size()
    weight = weight.unsqueeze(-1).unsqueeze(-1).expand(size)
```

```
        bias = bias.unsqueeze(-1).unsqueeze(-1).expand(size)
        return weight * x + bias
```

Train discriminator Network

```
class NetD(nn.Module):
    def __init__(self, ndf):
        super(NetD, self).__init__()

        self.conv_img = nn.Conv2d(3, ndf, 3, 1, 1)#128
        self.block0 = resD(ndf * 1, ndf * 2)#64
        self.block1 = resD(ndf * 2, ndf * 4)#32
        self.block2 = resD(ndf * 4, ndf * 8)#16
        self.block3 = resD(ndf * 8, ndf * 16)#8
        self.block4 = resD(ndf * 16, ndf * 16)#4
        self.block5 = resD(ndf * 16, ndf * 16)#4

        self.COND_DNET = D_GET_LOGITS(ndf)

    def forward(self,x):

        out = self.conv_img(x)
        #spectral normaliztion

        out = self.block0(out)
        out = self.block1(out)
        out = self.block2(out)
        out = self.block3(out)
        out = self.block4(out)
        out = self.block5(out)

        return out

class resD(nn.Module):
```

```
    def __init__(self, fin, fout, downsample=True):
        super().__init__()
        self.downsample = downsample
        self.learned_shortcut = (fin != fout)
        self.conv_r = nn.Sequential(

            spectral_norm(nn.Conv2d(fin, fout, 4, 2, 1, bias=False)),
            #spectral normaliztion
            nn.LeakyReLU(0.2, inplace=True),
            spectral_norm(nn.Conv2d(fout, fout, 3, 1, 1, bias=False)),
            #spectral normaliztion
            nn.LeakyReLU(0.2, inplace=True),
        )

        self.conv_s = nn.Conv2d(fin,fout, 1, stride=1, padding=0)
        #self.conv_s = spectral_norm(nn.Conv2d(fin,fout, 1, stride=1,
padding=0))
        self.gamma = nn.Parameter(torch.zeros(1))

    def forward(self, x, c=None):
        return self.shortcut(x)+self.gamma*self.residual(x)

    def shortcut(self, x):
        if self.learned_shortcut:
            x = self.conv_s(x)
            #spectral normaliztion
        if self.downsample:
            return F.avg_pool2d(x, 2) #GovalAvgPooling
        return x

    def residual(self, x):
        return self.conv_r(x) #spectral normaliztion
```

Spatial attention implementation

```
class SpatialAttention(nn.Module):
    def __init__(self, idf, cdf, what):
        super(SpatialAttention, self).__init__()
        self.conv_context4 = conv1x14(cdf, idf)
        self.conv_context8 = conv1x18(cdf, idf)
        self.conv_context16 = conv1x116(cdf, idf)
        self.conv_context32 = conv1x132(cdf, idf)
        self.conv_context64 = conv1x164(cdf, idf)
        self.conv_context128 = conv1x1128(cdf, idf)
        # self.conv_context3 = conv1x1(cdf, 128*128)
        self.sm = nn.Softmax()
        self.mask = None
        self.idf = idf
        self.reso = what
    def applyMask(self, mask):
        self.mask = mask  # batch x sourceL

    def forward(self, input, context):
        """
            input: batch x idf x ih x iw (queryL=ihxiw)
            context: batch x cdf x sourceL
        """
        idf, ih, iw = input.size(1), input.size(2), input.size(3)
        queryL = ih * iw
        batch_size, sourceL = context.size(0), context.size(2)

        # --> batch x queryL x idf
        target = input.view(batch_size, -1, queryL)

        targetT = torch.transpose(target, 1, 2).contiguous()
```

```
        sourceT = input.view(batch_size, -1, idf)
        sourceT = context.unsqueeze(3)

        if(self.reso == 0):
            sourceT = self.conv_context4(sourceT).squeeze(3)
        if(self.reso == 1):
            sourceT = self.conv_context8(sourceT).squeeze(3)
        if(self.reso == 2):
            sourceT = self.conv_context16(sourceT).squeeze(3)
        if(self.reso == 3):
            sourceT = self.conv_context32(sourceT).squeeze(3)
        if(self.reso == 4):
            sourceT = self.conv_context64(sourceT).squeeze(3)
        if(self.reso == 5):
            sourceT = self.conv_context128(sourceT).squeeze(3)
        attn = torch.bmm(targetT, sourceT)
        # --> batch*queryL x sourceL
        attn = attn.view(batch_size*queryL, sourceL)
        if self.mask is not None:
            # batch_size x sourceL --> batch_size*queryL x sourceL
            mask = self.mask.repeat(queryL, 1)
            attn.data.masked_fill_(mask.data, -float('inf'))

        # make the softmax on the dimension 1
        attn = self.sm(attn)
        # --> batch x queryL x sourceL
        attn = attn.view(batch_size, queryL, sourceL)
        # --> batch x sourceL x queryL
        attn = torch.transpose(attn, 1, 2).contiguous()
        # (batch x idf x sourceL)(batch x sourceL x queryL)
        # --> batch x idf x queryL
        weightedContext = torch.bmm(sourceT, attn)
        attn = attn.view(batch_size, -1, ih, iw)

        return weightedContext, attn
```

Channel-wise Attention implementation

```
class ChannelAttention(nn.Module):
    def __init__(self, idf, cdf, what):
        super(ChannelAttention, self).__init__()
        self.conv_context000 = conv1x1(cdf, 4*4)
        self.conv_context00 = conv1x1(cdf, 8*8)
        self.conv_context0 = conv1x1(cdf, 16*16)
        self.conv_context1 = conv1x1(cdf, 32*32)
        self.conv_context2 = conv1x1(cdf, 64*64)
        self.conv_context3 = conv1x1(cdf, 128*128)
        self.sm = nn.Softmax()
        self.idf = idf
        self.cdf = cdf
    def forward(self, weightedContext, context, ih, iw):
        batch_size, sourceL = context.size(0), context.size(2)
        sourceC = context.unsqueeze(3)
        if (ih == 4):
            sourceC = self.conv_context000(sourceC).squeeze(3)
        if (ih == 8):
            sourceC = self.conv_context00(sourceC).squeeze(3)
        if (ih == 16):
            sourceC = self.conv_context0(sourceC).squeeze(3)
        if (ih == 32):
            sourceC = self.conv_context1(sourceC).squeeze(3)
        if (ih == 64):
            sourceC = self.conv_context2(sourceC).squeeze(3)
        if (ih == 128):
            sourceC = self.conv_context3(sourceC).squeeze(3)
        attn_c = torch.bmm(weightedContext, sourceC)
        attn_c = attn_c.view(batch_size * weightedContext.size(1), sou
rceL)

        attn_c = self.sm(attn_c)
```

```
        attn_c = attn_c.view(batch_size, weightedContext.size(1), sour
ceL)
        attn_c = torch.transpose(attn_c, 1, 2).contiguous()

        weightedContext_c = torch.bmm(sourceC, attn_c)
        weightedContext_c = torch.transpose(weightedContext_c, 1, 2).c
ontiguous()
        weightedContext_c = weightedContext_c.view(batch_size, -
1, ih, iw)



        return weightedContext_c, attn_c
```

Appendix E: Human evaluation study sample google form

The first human evaluation form labeled as Q1

Human Evaluation User Study

Thank you for participating Human Evaluation User Study.

* Required

please see the images carefully and fill quick survey answer the question accordingly.
(እባክዎን ምስሎቹን በጥንቃቄ ይመልከቱ እና ፈጣን የዳሰሳ ጥናት ይሙሉ ለዚህ ጥያቄ መልስ ይስጡ።)

1. please write your E-mail (እባክዎን ኢሜልዎን ይፃፉ) *

2. 1) carefully look at the following images, choose one of them which more realistic image or more natural(የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ፣ ከመካከላቸው አንዱን የበለጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።) *

   Mark only one oval.

A

B

3. 2) carefully look at the following images, choose one of them which more realistic image or more natural(**የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ፣ ከመካከላቸው አንዱን የበለጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።**) *

   Mark only one oval.

A

B

4. 3) carefully look at the following images, choose one of them which more realistic image or more natural(**የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ፣ ከመካከላቸው አንዱን የበለጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።**) *

   Mark only one oval.

A

B

The second human evaluation form labeled as Q2

Thank you for participating Human Evaluation User Study.

Thank your willingness for participating Human Evaluation User Study.

* Required

please read the text quoted text descriptions and fill this quick survey based on given the text description answer the question accordingly. (እባክዎን የተጠቀሱትን የጽሑፍ መግለጫዎች ያንብቡ እና የጽሑፍ መግለጫው ለጥያቄው መልስ በመስጠት መሠረት ይህን ፈጣን የዳሰሳ ጥናት ይሙሉ ።)

1. please write your E-mail (እባክዎን ኢሜልዎን ይፃፉ) *

2. 1. From the this text description " Text description:- "a small bird with a red head, breast, and belly and black wings.", which one of the following image is more semantically consistent image and looks more real?(ከላይ ከተጠቀሰው የጽሑፍ መግለጫ ውስጥ ከሚከተሉት ምስሎች መካከል የትኛው በቅደም-ተከተል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል?) *

   Mark only one oval.

A

B

3. 2. From the this text description " Text description:- "this bird has wings that are grey with a short black bill.", which one of the following image is more semantically consistent image and looks more real?(**ከላይ ከተጠቀሰው የጽሑፍ መግለጫ ውስጥ ከሚከተሉት ምስሎች መካከል የትኛው በቅደም-ተከተል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል**?) *

   Mark only one oval.

A

B

4. 3. From the this text description " Text description:- "this bird has wings that are brown and has a white belly.", which one of the following image is more semantically consistent image and looks more real?(**ከላይ ከተጠቀሰው የጽሑፍ መግለጫ ውስጥ ከሚከተሉት ምስሎች መካከል የትኛው በቅደም-ተከተል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል**?) *

   Mark only one oval.

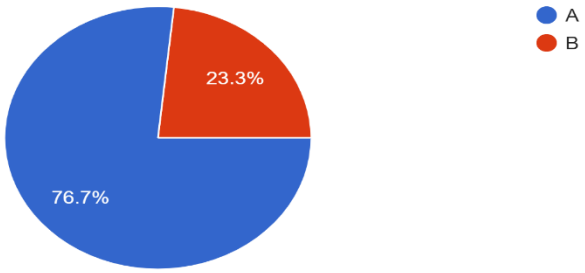
A

B

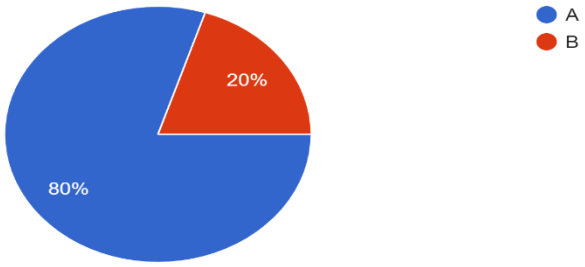
Appendix F: Human evaluation study Result

The result of the first question Q1

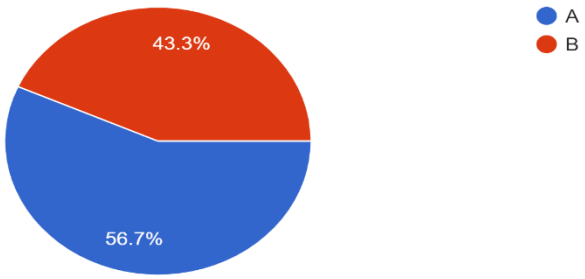
2) carefully look at the following images, choose one of them which more realistic image or more natural(የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ...ጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።)
30 responses



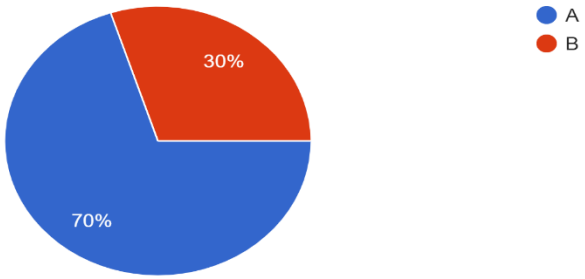
3) carefully look at the following images, choose one of them which more realistic image or more natural(የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ...ጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።)
30 responses



4) carefully look at the following images, choose one of them which more realistic image or more natural(የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ...ጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።)
30 responses

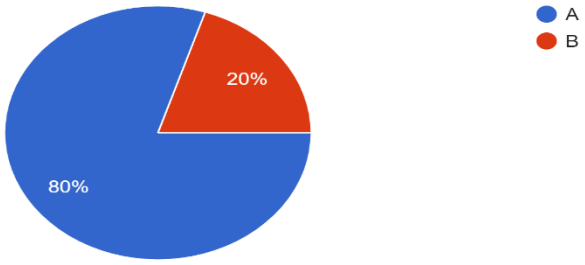


5) carefully look at the following images, choose one of them which more realistic image or more natural(የሚከተሉትን ምስሎች በጥንቃቄ ይመልከቱ ...ጠ እውነተኛ ምስል ወይም የበለጠ ተፈጥሯዊ ይምረጡ።)
30 responses

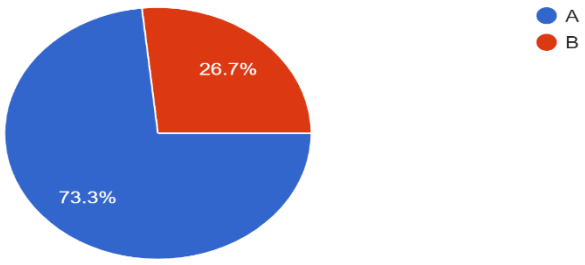


The result of the second question Q2

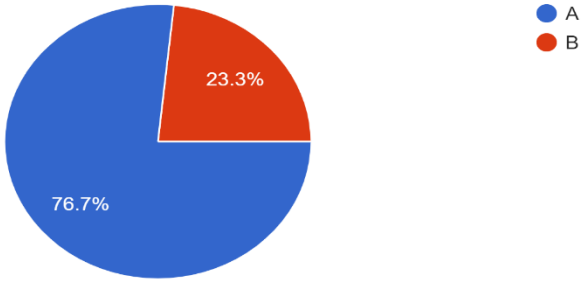
1. From the this text description " Text description:- "a small bird with a red head, breast, and belly and black wings.", which one of the follo...ከተል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል?)
30 responses



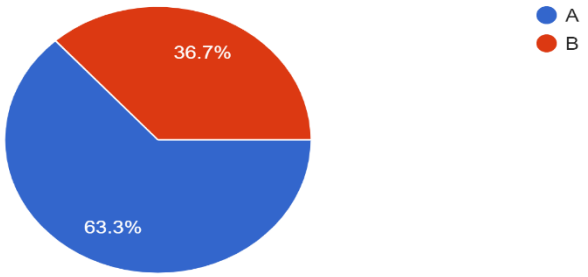
2. From the this text description " Text description:- "this bird has wings that are grey with a short black bill.", which one of the following i...ተከተል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል?)
30 responses



3. From the this text description " Text description:- "this bird has wings that are brown and has a white belly.", which one of the following ...ከተ‹ል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል?)
30 responses



4. From the this text description " Text description:- "a bird with a small pointed bill, brown eyebrow, and brown and white spots cover...ል የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል?)
30 responses



5. From the this text description " Text description:- "a bird with black head, white throat, nape and breast, black ring around throat and blac...ከላይ የተስተካከለ ምስል ነው እናም የበለጠ እውነተኛ ይመስላል?)

30 responses