DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

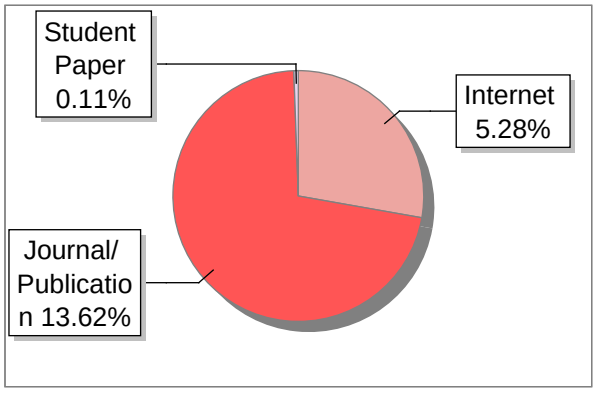
Author Name	Md madam students
Title	Project
Paper/Submission ID	776572
Submission Date	2023-06-17 08:00:22
Total Pages	41
Document type	Thesis

Result Information

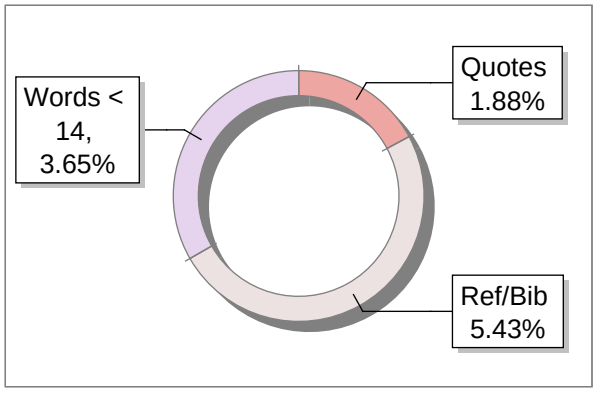
Similarity **19 %**

1 10 20 30 40 50 60 70 80 90

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

19	68	B
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	arxiv.org	1	Publication
2	arxiv.org	1	Publication
3	mdpi.com	1	Internet Data
4	mdpi.com	1	Internet Data
5	Synthesis of Prostate MR Images for Classification Using Capsule Network-Based G by Yu-2020	1	Publication
6	arxiv.org	1	Publication
7	arxiv.org	1	Publication
8	arxiv.org	1	Publication
9	crunchprep.com	1	Internet Data
10	arxiv.org	1	Publication
11	www.atlantis-press.com	<1	Internet Data
12	arxiv.org	<1	Publication
13	arxiv.org	<1	Publication
14	arxiv.org	<1	Publication

15	article.ijssn.org	<1	Publication
16	arxiv.org	<1	Publication
17	Explainable AI Interpreting, Explaining and Visualizing Deep Learning by Wojciec-2019	<1	Publication
18	IEEE 2018 IEEECVF Conference on Computer Vision and Pattern Recogni	<1	Publication
19	IEEE 2020 11th International Conference on Computing, Communication and Networ	<1	Publication
20	GANs enabled super-resolution reconstruction of wind field by Tran-2020	<1	Publication
21	openscholar.dut.ac.za	<1	Publication
22	arxiv.org	<1	Publication
23	arxiv.org	<1	Publication
24	Deep Learning Based Inter-subject Continuous Decoding of Motor Imagery for Pract by Roy-2020	<1	Publication
25	github.com	<1	Internet Data
26	IEEE 2020 11th International Conference on Computing, Communication and Networ	<1	Publication
27	moam.info	<1	Internet Data
28	Deep Sketch-guided Cartoon Video Inbetweening by Li-2021	<1	Publication
29	religiondocbox.com	<1	Internet Data
30	moam.info	<1	Internet Data

31	Polticas para avaliao da qualidade do Ensino Superior no Brasil um balano c by Dias-2006	<1	Publication
32	arxiv.org	<1	Publication
33	A measurement of the extent of market imperfections between markets and applicat by Hsu-2010	<1	Publication
34	Three-dimensional tumor models Promoting breakthroughs in nanotheranostics tran by Mapanao-2020	<1	Publication
35	2 An IT Enterprise Architecture Process Model by Grossman-1999	<1	Publication
36	www.spandidos-publications.com	<1	Publication
37	arxiv.org	<1	Publication
38	A night-time outdoor data set for low-light enhancement by Zhou-2020	<1	Publication
39	dokumen.pub	<1	Internet Data
40	pdfs.semanticscholar.org	<1	Internet Data
41	Thesis Submitted to Shodhganga Repository	<1	Publication
42	documents.mx	<1	Internet Data
43	journalofbigdata.springeropen.com	<1	Internet Data
44	aa.tamu.edu	<1	Publication
45	Explainable AI Interpreting, Explaining and Visualizing Deep Learning by Wojciec-2019	<1	Publication
46	arxiv.org	<1	Publication
47	arxiv.org	<1	Publication
48	mdpi.com	<1	Internet Data

49	ajmse.leena-luna.co.jp	<1	Internet Data
50	Diffusion network embedding, by Shi, Yong Lei, Min- 2019	<1	Publication
51	arxiv.org	<1	Publication
52	arxiv.org	<1	Publication
53	A novel super-resolution CT image reconstruction via semi-supervised generative by Jiang-2020	<1	Publication
54	coek.info	<1	Internet Data
55	moam.info	<1	Internet Data
56	moam.info	<1	Internet Data
57	Quantitative and chemically intuitive evaluation of the nature of ML bonds in by Brooker-2020	<1	Publication
58	arxiv.org	<1	Publication
59	asbmr.onlinelibrary.wiley.com	<1	Internet Data
60	docplayer.net	<1	Internet Data
61	docplayer.net	<1	Internet Data
62	documents.mx	<1	Internet Data
63	Human-robot collaboration within a virtual space using behavioral task morphologies	<1	Student Paper
64	jefjournal.org.za	<1	Publication
65	mrsej.ksu.ru	<1	Publication
66	paperswithcode.com	<1	Internet Data

67	plosjournal.deepdyve.com	<1	Internet Data
68	Thesis Submitted to Shodhganga, shodhganga.inflibnet.ac.in	<1	Publication

CHAPTER - 1

INTRODUCTION

1.1 Overview

In the fields of computer vision and natural language processing, text-to-image creation creates an image from a textual description. The aim of text-to-picture generation is to produce an image that visually appeals to the eye and captures the essence of a textual description. Generative adversarial networks (GANs), variational autoencoders (VAEs), and reinforcement learning are some of the methods used in text-to-image generation. Every strategy has pros and cons, and the best course of action relies on the particular demands of the work at hand.

A well-liked method for tackling this problem is to use generative adversarial networks (GANs), in which the discriminator and generator neural networks compete with one another. While the discriminator assesses the created image's realism, the generator generates an image based on the textual description. Through this process, the discriminator learns to differentiate between real and fake images while the generator is educated to create increasingly realistic images. Text-to-image generation has a wide range of applications, including advertising, e-commerce, gaming, concept visualization, automated image generation, generative art, AI-powered design, human-computer interaction, and image dataset augmentation. A Text to Image Synthesis system that can generate excellent images from textual descriptions is the end product.

1.2 Motivation

Multimodal learning, to put it simply, is learning anything while using more than one sense (visual, aural, and kinesthetic). Because how information is conveyed to humans can have a significant impact on how much of it they understand. According to studies, the majority of people learn best when different learning modalities are combined and applied simultaneously. It can be difficult to structure information on a subject simply by reading a lengthy paragraph; but, if the information is presented in the form of an image, it is much simpler to understand. Images are more appealing than text. Information can be conveyed more clearly using visual aids. The capacity to produce high-quality images that closely resemble real photos has allowed Generative Adversarial Networks (GANs) to provide ground-breaking outcomes.

The initial GAN, however, lacked the ability to edit the images it was trained to produce and produce images that adhered to a set of requirements. Text-to-Image generation (TTI) is one application that was created and implemented as a conditional GAN model to help overcome this obstacle. Its vast range of applications includes photo-searching, photo-editing, art creation, computer-aided design, image reconstruction, captioning, and portrait drawing. It has a substantial impact on many academic fields.

1.3 Objectives

The goals or learning path for creating a TTI (Text to Image Synthesis) model using the GAN architecture can be the following actions: -

- Generative Adversarial Networks (GANs): An Overview To lay the groundwork for the text to image synthesis model, learn about the design and operation of GANs.
- Data Gathering: To train the model, gather a sizable and varied collection of photographs and the textual descriptions that go with them.
- To prepare the textual descriptions and photos for usage in the model, preprocess them.
- Designing the generator and discriminator networks with the text to image synthesis task's unique requirements in mind is known as "model design."
- To learn the mapping between textual descriptions and photos, train the model using the dataset that has been gathered.
- Evaluation: Evaluate the performance of the model using metrics such as image quality, visual similarity, and diversity of generated images.
- Optimization: Improve the model performance through optimization techniques such as hyperparameter tuning and fine-tuning.
- Deployment: Deploy the final model in a user-friendly API or web application to make it accessible to end-users.
- Maintenance: Continuously monitor the model performance and update it to maintain its accuracy and relevance over time.

1.4 Scope

The Generative Adversarial Networks (GANs)-based Text to Image Synthesis system will be a cutting-edge method for transforming textual descriptions into visually cohesive images. An image representing the description will be produced by the system using a written description as input. High-quality, realistic photographs that have a visually pleasing structure and insightful substance will be produced. The following elements will be included in the system's scope:

- Data Collection
- Model Design
- Model Training
- Model Evaluation
- Deployment
- Maintenance

The generated images will be of high quality and realism, with a visually coherent structure and meaningful content and some of its lucrative aspects are

- Advanced Architecture: The system utilizes the latest advancements in GANs to generate high-quality and realistic images.
- User-friendly Interface: The system will be deployed as a user-friendly API or web application, making it accessible and easy to use for end-users.
- Customizable Output: The system can be fine-tuned to produce images with specific styles or characteristics.
- Continuous Improvement: The system will be regularly monitored and updated to maintain its accuracy and relevance over time.

1.5 Existing System

GANs (Generative Adversarial Networks) are used in a number of existing text to picture creation systems, including:

1. One of the first systems to convert text into images was StackGAN, which creates high-resolution images from text descriptions using a multi-stage GAN architecture.
2. The AttnGAN system adds an attention mechanism to the GAN design, enabling the image generator to concentrate on particular areas of the image as it generates it.
3. MirrorGAN employs a distinct attention strategy that is applied to both the generator and discriminator in the GAN architecture, although it is built on a similar principle as AttnGAN.
4. DALLE 2: A new text-to-image generating system from OpenAI that can produce a variety of excellent images from textual inputs.
5. Image-to-Image Translation with Conditional Adversarial Networks (pix2pix): This is a more general framework for image generation that can be adapted for text-to-image generation by conditioning the generator on text descriptions.

These are just a few examples of existing systems for text-to-image generation using GANs. The choice of the system to use depends on the specific requirements of the task and the computational resources available. And in conjunction some of the demerits of these systems are as the following,

1. First off, it still struggles to produce graphics with logical text. For instance, when given the instruction "A sign that says deep learning," it generates the images shown below with gibberish.
2. DALL·E 2 also has inherent biases due to the skewed nature of data collected from the internet. It has gender-biased occupation representations.
3. DALL.E 2 has difficulty connecting properties to objects. When prompted to create an image of "a red cube on top of a blue cube," it can be difficult to identify which cube should be red and which should be blue.

1.6 Proposed System

Using a text encoder, we encrypt the text query. The description embedding is concatenated to the noise vector after being initially compressed using a fully-connected layer to a minimal dimension. The deconvolutional layer, where up sampling occurs, receives it next.

The discriminator takes into account both the generated image and the written description. It then evaluates how well the created image matches the real image, computes the loss function, and modifies the weights as necessary.

Various word embedding methods, including GloVe (Global Vectors for Word Representation), a trained word embedding methodology, and BERT, will be used to compare the results (Bidirectional Encoder Representations from Transformers)

CHAPTER - 2

PROBLEM STATEMENT

2.1 Problem Statement

Generative Adversarial Network (GAN), which consists of a generator and a discriminator, is the deep learning method we utilized. For the creation of text to images, we also want to use Tensorflow, Numpy, NLTK, and Tensorlayer. Comparatively speaking, Tensorflow compiles more quickly than other deep learning libraries. We have utilized the NLTK (Natural Language Toolkit) tokenizer for text division, which is the process of breaking up larger text into smaller pieces like words. It enables the computer to examine, prepare, and comprehend the written text that the user inputs. Tensor layer, a library built on top of TensorFlow, will be used during model training to construct different layers in the network for the generator and discriminator, such as input layer, convolutional 2D layer, dense layer, etc. The Python Pickle module is used to serialize data and transform objects into bytes so that they can be conveniently stored in files or transported.

CHAPTER - 3

DETAILED SURVEY

3.1 Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network

[1]"Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network" is a research paper that proposes a novel approach to text-to-image synthesis using a hierarchical GAN architecture. The authors aim to overcome the limitations of traditional GANs, which often generate images with low resolution and lack of fine-grained details. The proposed approach, referred to as Hierarchically-nested Adversarial Network (H-GAN), utilizes a multi-level GAN architecture to generate high-resolution images with fine-grained details.

Traditional GANs frequently produce images with low resolution and a lack of fine-grained details, which the authors want to address. The suggested method, known as Hierarchically-nested Adversarial Network (H-GAN), makes use of a multi-level GAN architecture to produce high-resolution images with minute features. A generator network and a discriminator network make up the two primary parts of the H-GAN. Each level of the generator network's levels generates a picture with a better resolution. A written description is the generator network's input, and a synthesized image is its output. The authors also introduce a novel loss function, referred to as a texture loss, which helps to ensure that the generated images have realistic textures.

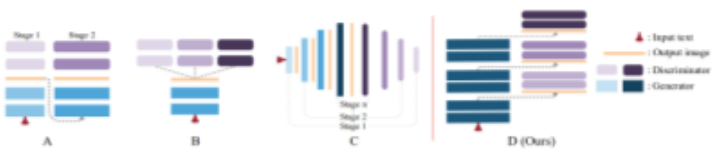


Fig - 1 H-GAN Architecture

The authors show that in terms of image quality, resolution, and fine-grained features, the H-GAN outperforms conventional GANs and cutting-edge text-to-image synthesis models. This research is significant because it advances the state of the art in text-to-image synthesis and has potential applications in a variety of fields, including digital marketing, education, and entertainment.

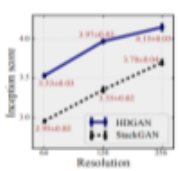


Fig - 2 H-GAN Results

3.2 Instance-level Image Translation by Deep Attention Generative Adversarial Networks

[2]The authors of this research suggested a DA-GAN model. It introduces a unique method for translating images using a deep attention Generative Adversarial Network and offers the first solution by dividing the task of translating samples from two independent sets into translating instances in a highly-structured latent space (DAGAN).

Picture translation is the process of changing an image from one domain to another, for as going from a grayscale image to a colour image or from a sketch to a realistic image.

The authors want to get around typical GANs' drawbacks, which include frequently hazy images and a lack of fine-grained features.

The suggested method, known as DAGAN, makes use of a deep attention mechanism to make sure that the images produced contain high resolution and fine-grained features. The attention loss, a brand-new loss function that the authors introduce, aids in preserving the source images' fine-grained information in the generated images.

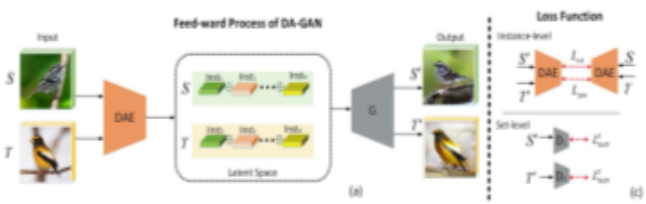


Fig - 3 DA-GAN Architecture

The main contributions can be summarized into the following stages -

- To considerably improve controllability and enable utilisation at both the instance-level and set-level, they divided the task into instance-level picture translation.
- And are the first to include the attention mechanism into generative adversarial networks, to the best of our knowledge.
- A unique framework called DA-GAN is presented, and it provides results that are both aesthetically pleasing and useful for a wide range of jobs.

Method	Accuracy
DA-GAN	94.3 %
DA-GAN w/o *DAE*	90.2 %
DA-GAN w/o const	79.8%
DA-GAN w/o Sym	90.6%
DA-GAN w/o D_2	88.2%

Method	Accuracy
SA[6]	59.32 %
DANN[8]	73.85 %
DTN [34]	84.44%
UNIT [21]	90.53 %
DA-GAN	**93.60** %

Fig - 4 DA-GAN Results

3.3 Fine-Grained Text to Image Generation with Attentional Generative Adv Networks

[3]The authors of this research put forth the Attentional Generative Adversarial Network (AttnGAN), which enables multi-stage, attention-driven refinement for precise text-to-image generation. The AttnGAN can synthesise fine-grained information at particular portions of the image by focusing on the relevant words in the natural language description.

The generative network may produce different picture subregions based on the phrases that are most relevant to those subregions using an attention model. The DAMSM learns two neural networks that map sentence words and image subregions to a shared semantic space in order to calculate a fine-grained loss for picture generation. With this technique, word-level similarity between the text and the image is measured.

The fine-grained image-text matching loss is computed by the deep attentional multimodal similarity model.

To train the AttnGAN generator, the deep attentional multimodal similarity model computes the fine-grained image-text matching loss and comes up with a 4.36 inception score. AttnGAN uses a multi-stage process to produce photos of high quality.

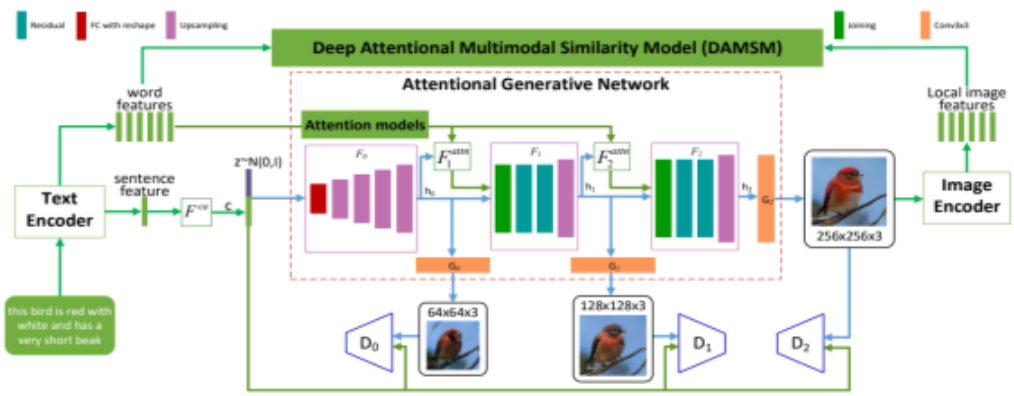


Fig -5 AttnGAN Architecture

3.4 Text-to-face generation using styleGAN2

[4]The author proposed a method that uses StyleGAN2 for image generation and BERT for text encoding to produce high-quality images.

The ability of BERT can perform state-of-the-art tasks in Natural Language Processing was the main deciding factor in our choice of BERT as our language encoder. When performing language processing tasks, BERT employs bi-directional training of transformers, which gives the model a deeper comprehension of the language context and flow than single-directional models. Therefore, unlike other context-free word embeddings like GloVe or Word2Vec, BERT embeddings are constructed taking into account the context in which the words are used.

The StyleGAN2 generator was chosen because, with the addition of features like weight demodulation, path length regularisation, and avoidance of progressive development, it produced cutting-edge results in face generation. These qualities have made it possible to get around problems with the StyleGAN design, namely the production of phase artefacts and the droplet effect.

The peak-signal-to-noise ratio (PSNR), root mean squared error (RMSE), and pixel loss based on mean squared error (MSE) are the three most often employed loss functions in image enhancement procedures (PSNR). When compared to per-pixel loss, feature loss or perceptual loss is a better metric since it allows for the reconstruction of finer information in images.

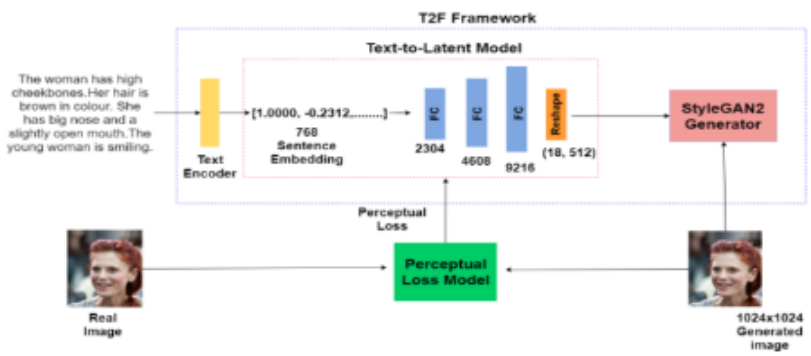


Figure 1. Proposed Model Architecture

Fig - 6 STYLEGAN Model Architecture

3.5 Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks

[5]For creating photorealistic visuals from text descriptions, the authors suggested a unique Stacked Generative Adversarial Networks. It breaks down the challenging problem of producing high-resolution photographs into smaller, easier-to-manage subproblems.

There are two components to the architecture:

The Stage-I GAN is the initial stage of the StackGAN architecture and it creates low-resolution images from the textual description using a Generative Adversarial Network. The second step of the StackGAN architecture, Stage-II GAN, is used to enhance the low-resolution images produced by Stage-I GAN. Another Generative Adversarial Network, known as a Stage-II GAN, uses the low-resolution photos as input and generates high-resolution photorealistic images.

The CUB dataset, which contains 11,788 photos of 200 different bird species, is one of the datasets used in this model.

On the CUB dataset, StackGAN improves its inception score by 28.47% as compared to GAN-INT-CLS.

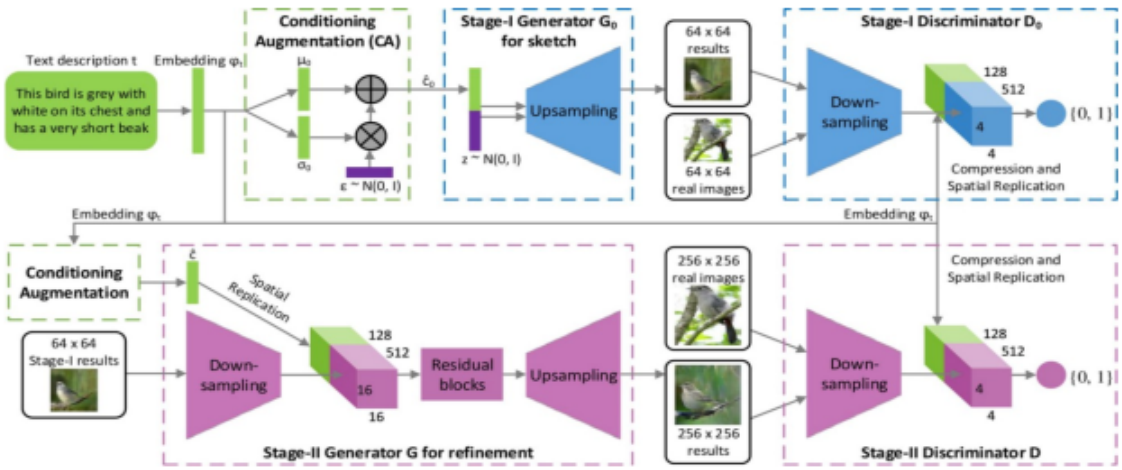


Fig - 7 Proposed STACKGAN Model Architecture

3.6 Object-driven Text-to-Image Synthesis via Adversarial Training

[6]The authors suggested object-driven attentive generative adversarial networks (Obj-GANs), which allow object-centered text-to-image synthesis for complex scenarios. To generate intricate graphics from word descriptions, an object-driven attentive generative network (Obj-GAN) is advised. Two novel components are offered, namely the object-driven attentive generating network and the object-wise discriminator. In order to enable picture formation based on the semantic layout generated in the first phase during the image generation stage, the object-driven attentive generator and object-wise discriminator are built.

The Obj-GAN is first given the language, which produces a series of objects with provided bounding boxes (with class labels) and forms. In the second stage, the shape generator defines the shape of each object within its enclosing box. The picture maker produces The picture generator creates a better quality image by paying attention to the most pertinent words and pre-generated class labels in various places.

The COCO dataset is used for analysis. It has 80 object classes, and there are 5 text descriptions and 80 object-wise annotations (such as bounding boxes and shapes) for each image. For the training and test phases, we use the official 2014 train (over 80K photos) and validation (over 40K images) splits. It generates a distance from conception score of 25.83 and an inception score of 27.37.

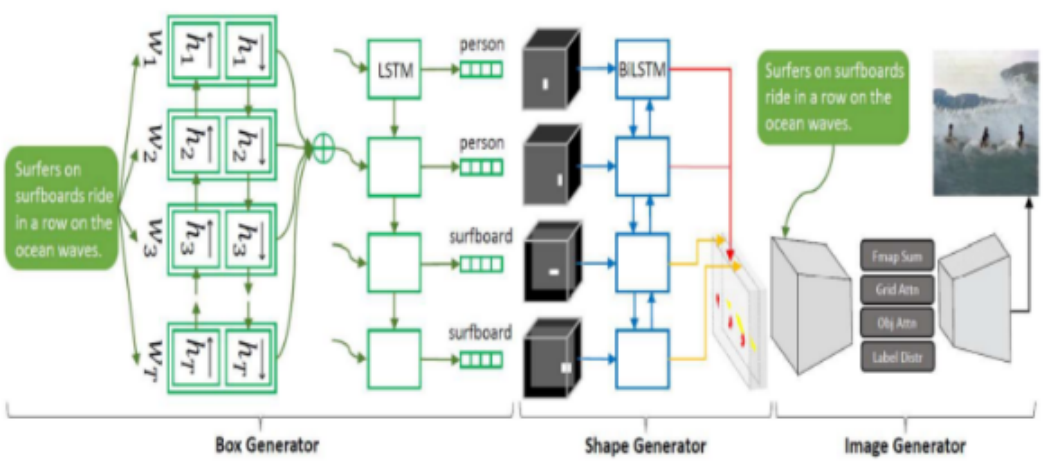


Fig - 8 Obj- GAN Model Architecture

3.7 DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis

[7]A brand-new Dynamic Memory Generative Adversarial Network was suggested by the authors (DM-GAN). It was done to address two issues. First, the quality of the initial photos has a significant impact on the generation output. If the starting photos are not of high quality, the image refinement procedure cannot produce them. Second, the information showing the substance of the image varies depending on the word in the input sentence. The refinement procedure is ineffective in current models because they use the same word representations across several image processing stages. The value of each word for refining should be determined by taking into account the visual information.

The author suggested integrating a memory system to address the first issue. for the key-value memory structure framework to be implemented in the GAN. Fuzzy image features from the first image are treated as requests to read features from the memory module. Memory reads are used to improve the original images. To solve the second issue, we add a memory writing gate that dynamically selects words that are relevant to the produced image. As a result, the produced image and the text description are tightly associated. As a result, based on the initial image and text information, the memory component is dynamically written to and retrieved each time a picture is enhanced. A response gate is also used to adaptively receive data from the picture and memory. them.

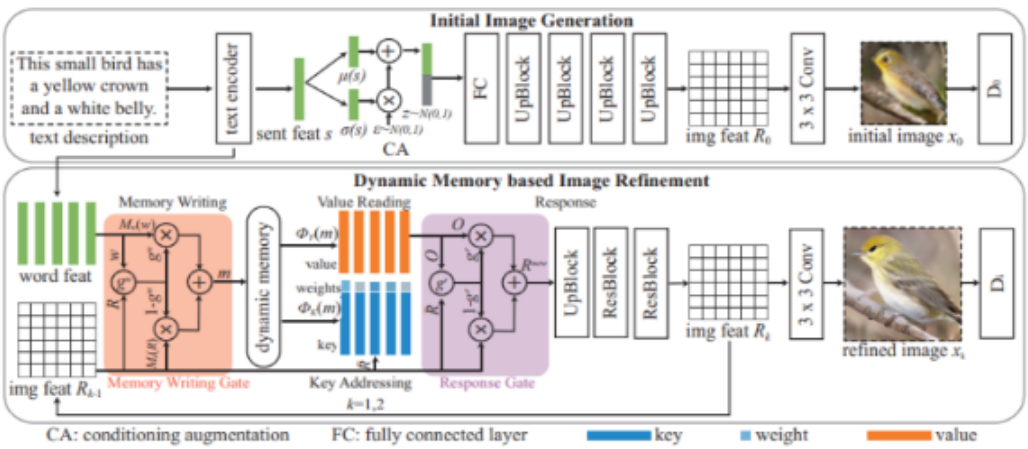


Figure 2. The DM-GAN architecture for text-to-image synthesis. Our DM-GAN model first generates an initial image, and then refines the initial image to generate a high-quality one.

Fig - 9 DM-GAN Model Architecture

3.8 Texture GAN : Controlling Deep Image Synthesis with Texture Patches

[8]The writers suggested The first deep image generation technique that lets users modify object texture is TextureGAN. The network realistically adds one or more example textures on the indicated objects after a user "drags" them onto sketched versions of such objects.

From input sketches with textures overlay, a conditional generative network known as TextureGAN learns to produce realistic images. The generator in Texture GAN consists of two parts: a structural generator and a texture generator. The structural generator generates the basic structure of the image, while the texture generator controls the texture. The texture generator takes as input a set of texture patches and applies them to the generated image.
The model was evaluated on a number of datasets, including DTD, MNIST, and CIFAR-10, and showed that their method can generate high-quality images with consistent textures.

Fig - 10 Texture display

3.9 Semantics Disentangling for Text-to-Image Generation

[9]The authors aim to overcome the limitations of traditional GANs, which often produce images with low resolution and lack of fine-grained details. Additionally, traditional GANs often have difficulty generating images that match the textual description. The proposed technique, known as SD-GAN, makes use of a cutting-edge architecture to separate the semantic data from the textual description. The decoupled semantic information is then used to govern the image generation. The semantic loss is a brand-new loss function that the authors introduce to help guarantee that the generated images correspond to the textual description.

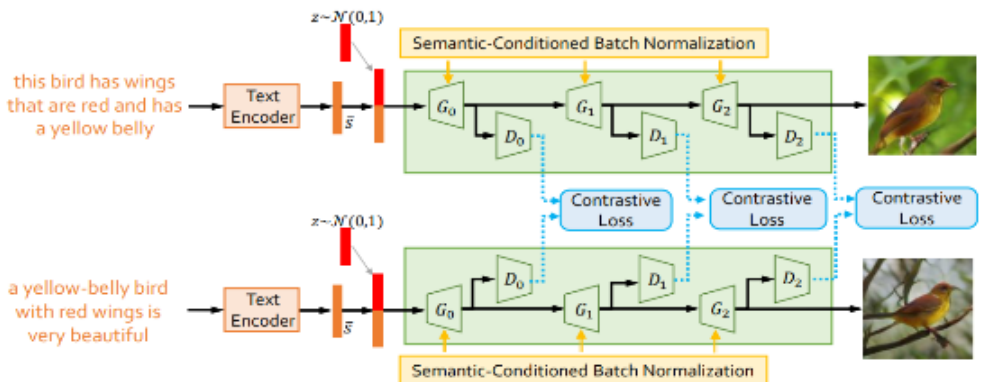


Fig - 11 The architecture of SD-GAN

The authors conduct extensive experiments to evaluate the performance of the SD-GAN. The experiments show that the SD-GAN outperforms traditional GANs in terms of image quality, resolution, and fine-grained details. The authors also compare the SD-GAN to several state-of-the-art text-to-image synthesis models and demonstrate that the SD-GAN produces images with higher quality and resolution.

Fig - 12 SD-GAN Results

3.10 Realistic image synthesis with stacked generative adversarial networks

[10]The proposed approach, referred to as SGAN, utilizes a stacked architecture to generate high-resolution images with fine-grained details. The authors introduce a novel loss function, referred to as the gradient penalty, which helps to ensure that the generated images are realistic and have the same distribution as the real images. The gradient penalty is calculated by comparing the gradients of the generated images and the real images.

The results demonstrate that in terms of image quality, resolution, and fine-grained details, STACKGAN-V2 surpasses conventional GANs and other cutting-edge image synthesis models. The authors also show how STACKGAN-V2 can produce graphics with a resolution of up to 256x256 pixels from textual descriptions.

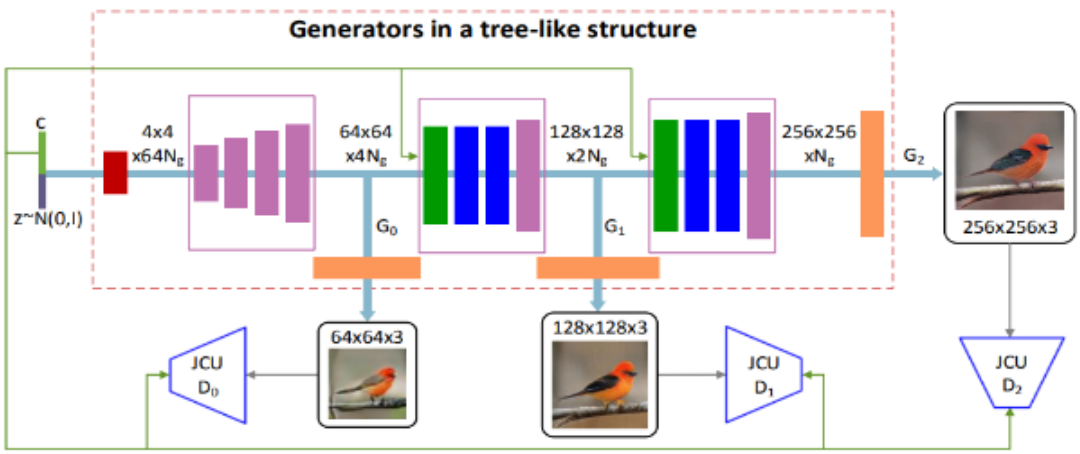


Fig - 13 STACKGAN-V2

The two-stage architecture used by STACKGAN-V2 produces low-resolution images in the first stage, which are subsequently enhanced to yield high-resolution images with fine-grained details in the second stage.

Fig - 14 STACKGAN-V2 Results

3.11 Semantic Image Synthesis via Adversarial Learning

[11]The author has proposed a model based on a conditional GAN framework that conditions on both images and text descriptions. An encoder-decoder design for generator G is used. The encoders are used to encrypt the text descriptions and source images. After that, the decoder creates images from texts and features representations of images. The differentiating task is carried out by the discriminator D using text semantic features. A convolutional neural network (CNN) is used as the encoder to convert source images with a size of 64 by 64 into spatial feature representations with a dimension of 16 by 16 by 512. In all convolutional layers, ReLU activation is applied. With the exception of the first convolutional layer, batch normalization is applied to all other layers.

There are various residual blocks that make up the residual transformation unit. The component then encodes the combined representations of the image and text features. This item is included in our model for two different reasons. First, it makes the identify function simple for the generator network to learn, ensuring that the output image keeps the same structure as the original image. Second, it enables a more thorough encoding procedure for the model, which in turn improves the encoding of the combined image and text representations.Multiple upsampling stages in the decoder convert the latent feature representations into synthetic data.

We first use convolutional layers to downsample the images into feature representations of dimension 4 X 4 X 512 for the discriminator. The final probabilities are then generated by concatenating the picture representation with the text embeddings, followed by the application of two convolutional layers.

3.12 DU-VLG: Unifying Vision-and-Language Generation via Dual Sequence to Sequence Pre-training

[12]The authors suggest DU-VLG, a deep learning-based framework that combines the advantages of models for generating language from vision and vision from language.

A vision-to-language generator and a language-to-vision generator are the two primary parts of the system. The language-to-vision generator is trained to create a picture from a text description, while the vision-to-language generator is trained to create a text description from an image. Both generators are pre-trained using a substantial number of image-text combinations before being fine-tuned for the intended objective.

The authors describe a brand-new dual sequence-to-sequence pre-training method that enables the complementary training of the two generators. The pre-training procedure aims to enhance the quality of the text and images generated, as well as to urge the two generators to collaborate to provide high-quality image-text pairs.

The proposed framework is assessed by the authors using a number of benchmark datasets, and their results are contrasted with those of current state-of-the-art techniques. They discover that their method exceeds the current approaches in terms of the output of high-quality image-text pairs as well as the quality of the image and text.

As a result, "DU-VLG: Unifying Vision-and-Language Generation through Dual Sequence-to-Sequence Pre-training" proposes a fresh method for integrating the two. By conducting thorough testing on benchmark datasets, the authors present a framework that incorporates the advantages of both vision-to-language and language-to-vision generation models. The suggested method advances the state-of-the-art in the area of vision-and-language generation and offers an exciting new line of inquiry.

3.13 Least squares generative adversarial networks

[13]A novel method for training generative adversarial networks is suggested in the publication "Least Squares Generative Adversarial Networks (LS-GANs)" (GANs). GANs are a sort of deep learning model that have been effective at producing high-quality images, but they can be challenging to train because the training process is unstable.

The least squares loss function, as opposed to the conventional cross-entropy loss function, is suggested to be used during training by the paper's authors. There is evidence that the least squares loss function performs better in several regression tasks and is more stable than the cross-entropy loss function.

The generator network in LS-GANs is trained to produce data that minimizes the least squares loss between the generated data and the real data. By optimizing the least squares loss, the discriminator network is trained to discriminate between the created data and the real data.

LS-GANs outperform typical GANs in terms of image quality, stability during training, and convergence time, according to experiments on a variety of datasets. Additionally, LS-GANs perform well in a number of applications, such as super-resolution, style transfer, and picture production.

In conclusion, "Least Squares Generative Adversarial Networks" introduces a novel method for training GANs that substitutes the least squares loss function for the conventional cross-entropy loss function. The researchers demonstrate that LS-GANs perform better than conventional GANs in terms of image quality, stability throughout training, and convergence speed, making them a promising replacement for deep learning-based image production.

3.14 Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks

[14]An approach based on deep learning is presented in the paper "Auto-Painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks." The authors suggest a system dubbed Auto-Painter that turns sketches into full-color cartoon graphics using conditional Generative Adversarial Networks (cGANs).

Both a discriminator network and a generator network make up the cGAN architecture of Auto-Painter. The discriminator network is trained to discern between the generated images and real photos, while the generator network is trained to produce cartoon images that correspond to the input sketch.

The generator network is first fine-tuned on a smaller dataset of sketches and the related cartoon images. The authors pre-train the generator network using a large dataset of real cartoon images. The end result is a generator network that can produce cartoon images of a high caliber that are accurate to the input sketches.

The authors compare Auto-Painter to various state-of-the-art techniques in terms of image quality and precision in matching the input sketches by evaluating it on a variety of sketch datasets. They also demonstrate that the system can produce cartoon graphics in a variety of styles, such as anime and Disney-style cartoons, and that it can handle a broad variety of sketch types, including hand-drawn sketches and scanned sketches.

The paper "Auto-Painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks" concludes by presenting a deep learning-based method for producing cartoon images of high quality from sketches. The authors illustrate how their Auto-Painter system, which handles a broad variety of sketch styles and produces cartoon images in various styles, surpasses existing systems in terms of image quality and accuracy when matching the input sketches.

3.15 Conditional image synthesis with auxiliary classifier gans

[15]A novel method for conditional image synthesis—a job in which an image is produced based on a given input, such as a class label or a textual description—is put forth in the paper "Conditional Image Synthesis using Auxiliary Classifier GANs (AC-GANs)". The authors provide an innovative form of conditional image synthesis-capable Generative Adversarial Network (GAN) known as an Auxiliary Classifier GAN (AC-GAN).

The authors demonstrate how adding the class label prediction job to the discriminator network can enhance the output quality and increase the generator network's resistance to input variance. Additionally, they demonstrate how AC-GANs perform better than conventional GANs in a number of benchmark datasets and applications, including picture creation, image-to-image translation, and super-resolution.

In conclusion, the paper "Conditional Image Synthesis with Auxiliary Classifier GANs" introduces a novel method for conditional image synthesis based on an AC-GAN. The authors demonstrate how adding the class label prediction job to the discriminator network can enhance the output quality and increase the generator network's resistance to input variance.The process of creating images and classifying them becomes conditional on the class label when the class is added as an input, thus the name. The result is a more reliable training process and a generator model that can be applied to produce images of a certain type, such as class labels.

SURVEY SUMMARY TABLE

SL NO	Title of Paper	Problem Addressed	Author's Approach/Method	Results
1	Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network [1]	Render fine-grained image details for high resolution images	Hierarchically-nested discriminators and visual-semantic similarity measure	The results show consistency in picture structures, colour, and styling.
2	Instance-level Image Translation by Deep Attention Generative Adversarial Network [2]	Inability to learn instance-level correspondences and aligned semantic parts	The process of converting samples into instances in a latent space is broken down	Semantic relation for visually appealing results.
3	Fine-Grained Text to Image Generation with Attentional Generative Adv Networks [3]	Lack of important fine grained information at the word level, giving low quality	Attention-driven, multi-stage refinement and DAMSM	Generate high quality image through a multi-stage process
4	Text-to-face generation using styleGAN2 [4]	To generate photo-realistic images of the human face that are aligned with the input descriptions.	To use BERT as the text encoder and StyleGAN2 architecture to produce high-quality facial images.	Face Semantic Distance:0.9224 Face Semantic Similarity:56.96
5	Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks [5]	To generate realistic images from text description	The author used stackGAN to produce high quality images	Inception score:27.37 FID:25.83 for COCO dataset

6	Object-driven Text-to-Image Synthesis via Adversarial Training [6]	Using a written description to create realistic visuals	(Obj-GANs) that enable complicated scene object-centered text-to-image synthesis.	Inception score:27.37 FID:25.83 for COCO dataset
7	DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis [7]	Generating realistic images from text descriptions and improving the content of fuzzily generated images.	Dynamic Memory Generative Adversarial Network is used to address the problem	On the CUB dataset and the COCO dataset, the DM-GAN increases R-precision by 4.49% and 3.09%, respectively.
8	Texture GAN : Controlling Deep Image Synthesis with Texture Patches [8]	To control the texture of the image from the user	The authors proposed TextureGAN, which allows the user to control the texture of the image that would be generated	Produced high-quality images with accurate and consistent textures
9	Semantics Disentangling for Text-to-Image Generation [9]	Inability to infer into the semantics of the text to depict accurately	SDGAN separates the semantic commons from the text consistency and keeps the semantic richness.	Our SD-GAN achieves the inception score of 4.67 .09 on the CUB dataset.
10	Realistic image synthesis with stacked generative adversarial networks [10]	Using a written description to create realistic visuals	Stacked Generative Adversarial Networks were designed to produce photostylized, high-resolution images.	Fixes flaws in the low-resolution Stage-I image
11	Semantic Image Synthesis via Adversarial Learning [11]	Identify image and provide suitable text descriptions	The author combines adversarial loss and semantic loss to improve the quality of the generated images.	Results show that the model outperforms other models

12	DU-VLG: Unifying Vision-and-Language Generation via Dual Sequence-to-Sequence Pre-training [12]	Add appropriate caption to the given image	The authors proposed DU-VLG,which includes vision-to-language model as well as language to vision model and they are made to function simultaneously.	Results show that the model outperforms other models such as AC-GAN
13	Least squares generative adversarial networks [13]	To improve the training stability of the GAN	The author proposes to use least squared loss function in place of standard cross-entropy as it has a better stability	Results show that training the GAN using this method produces better image in terms of quality
14	Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. [14]	To generate colored cartoon images from black and white sketches	The authors suggest using conditional Generative Adversarial Networks in a system they term Auto-Painter (cGANs)	The authors show that their system, Auto-Painter, outperforms existing methods in terms of image quality and accuracy in matching the input sketches
15	Conditional Image Synthesis with Auxiliary Classifier GANs [15]	To generate image from text description	The author makes use of a type of GAN called Auxiliary Classifier-GAN	The author shows that incorporating the class label prediction task into the discriminator network can improve the quality of the generated images

CHAPTER - 5

SYSTEM REQUIREMENT SPECIFICATION

5.1 Functional Requirements

Here are some functional requirements for the implementation of text to image synthesis in a Python environment :

- Data processing: The system must be able to preprocess the text and image data, including tokenization, padding, and normalization.
- Text encoding: The system must have a mechanism for encoding the textual descriptions into a suitable format for use as input to the generative network.
- Generative network architecture: The system must implement a Generative Adversarial Network (GAN) or a variant of GAN, such as STACKGAN or AttnGAN, for generating images from textual descriptions.
- Training: The system must have a mechanism for training the generative network on the preprocessed data, including loss functions and optimization algorithms.
- Model checkpointing: The system must have the capability to save and load trained models for later use.
- Evaluation: The system must have a mechanism for evaluating the performance of the trained model, including metrics such as image quality, resolution, and fine-grained details.
- Scalability: The system must be scalable and able to handle large amounts of text and image data.
- Compatibility: The system must be compatible with Jupyter notebooks and the Python programming language.
- Documentation: The system must have clear and comprehensive documentation, including explanations of the algorithms and processes used.
- Robustness: The system should be robust and handle edge cases and errors gracefully.

5.2 Non-Functional Requirements

Here are some non-functional requirements in the IEEE ERS format for the implementation of the text to image synthesis application in a Python environment:

Usability:

1.1 The programme must have an intuitive user interface.

1.2. The programme must give users short and clear error messages to help them fix any problems.

1.3 The application must give developers thorough instructions and examples to use.

Performance:

2.1. The programme must be responsive quickly when producing images from text inputs.

2.2. The programme needs to be able to manage a lot of requests coming in at once from many users.

2.3. The application needs to have a fast throughput rate and minimal latency.

Security:

3.1. To safeguard sensitive data, the application must provide secure processes for authentication and authorisation.

3.2. To protect data confidentiality, the application must use encryption during data transmission.

3.3. The application needs to be created with security in mind and adhere to accepted security practices and standards.

Scalability:

4.1. The programme needs to be scalable and capable of handling an expanding data volume over time.

4.2. The programme needs to be able to accommodate a growing user base without performance suffering.

Compatibility:

5.1. The application must be compatible with a variety of platforms and devices.

5.2. The application must be compatible with a variety of web browsers.

Reliability:

6.1. The programme must function properly and be accessible around-the-clock.

6.2. The application must have failover measures in place to guarantee service continuity in the event of a malfunction.

6.3. The programme must include built-in monitoring and logging to help with quick issue discovery and repair.

5.3 Hardware Requirements

Hardware requirements for text-to-image synthesis can include:

1. Processor: A high-performance processor is needed to handle the complex computations involved in generating images.
2. Memory: A sufficient amount of RAM is needed to store the data and intermediate results during image generation.
3. Storage: Adequate storage is needed to store the generated images and other data.
4. Graphics Processing Unit (GPU): A powerful GPU can significantly improve the performance of the system and speed up image generation.
5. Network: A fast and reliable network connection is necessary for the system to access training data and other resources.
6. Power Supply: A reliable power supply is necessary to ensure that the system remains operational.
7. Cooling: Adequate cooling is necessary to prevent overheating and ensure the stability of the system.
8. Operating System: The system should be compatible with a suitable operating system,

such as Windows, Linux, or macOS.

9. Input/Output (I/O) Devices: The system should have appropriate I/O devices, such as a keyboard and mouse, for user interaction.

10. Monitor: A monitor is necessary for displaying the generated images and for interaction with the system.

5.4 Software Requirements

Software requirements for text-to-image synthesis can include:

1. Deep Learning Framework: A deep learning framework, such as TensorFlow, PyTorch, or Caffe, is needed to develop and train the model.

2. Image Processing Library: An image processing library, such as OpenCV, Pillow, or scikit-image, is needed to process and manipulate images.

3. Natural Language Processing Library: A natural language processing library, such as NLTK, spaCy, or Gensim, is needed to process and analyze textual input.

4. Python: Python is a commonly used programming language for text-to-image synthesis and provides a rich set of libraries for data processing and machine learning.

5. Source Code Management: A source code management tool, such as Git, is necessary for version control and collaboration.

6. Development Environment: A development environment, such as Jupyter Notebook or an Integrated Development Environment (IDE), is necessary for developing and testing the system.

CHAPTER - 6

SYSTEM DESIGN

6.1 System Design

6.1.1 System Architecture

A generator and a discriminator are its two major parts.

1. Generator: The generator is in charge of producing fresh, artificial visuals. It converts input random noise into a synthetic image that mimics a real anime or cartoon image by adding input random noise.

2. Discriminator: it is in charge of assessing if a picture is created or genuine. It accepts actual and artificial photos and returns a likelihood that the image is real.

The generator tries to trick the discriminator, while the discriminator tries to accurately determine whether a picture is genuine or produced. These two components are taught in an adversarial way. The training procedure is repeated until the discriminator can accurately determine whether a picture is genuine or produced, and the generator is able to produce images that are indistinguishable from actual photos. Depending on the particular needs of the application, the system architecture may also include additional components in addition to these two, such as a data preprocessor, a data augmentation module, and a visualization module. The entire architecture should be made to be adaptable and expandable, enabling simple adjustment as needed

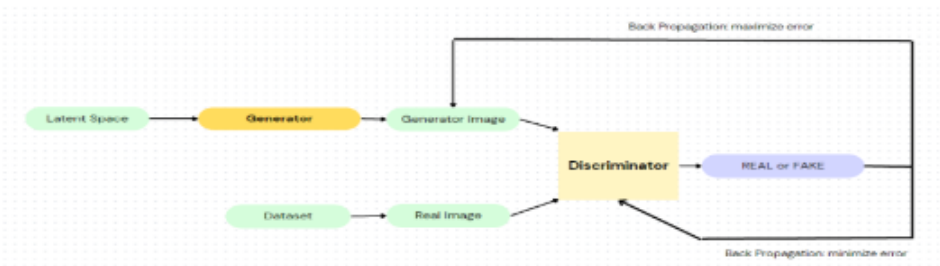


Fig - 15 System Architecture

6.1.2 Module Design

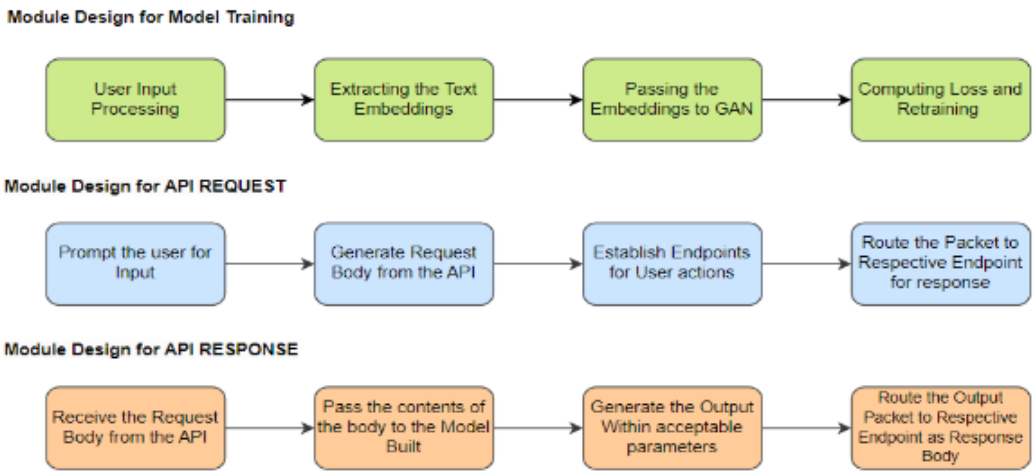


Fig - 16 Modules Design

The module design for the application can be outlined into the following key aspects in terms of their functionality -

- Data Preprocessor
- Generator
- Discriminator
- Loss Function
- Optimizer
- Training Loop
- UI (User Interface API)
- Database or Local Storage blocks

6.2 Detailed Design

6.2.1 Activity Diagram

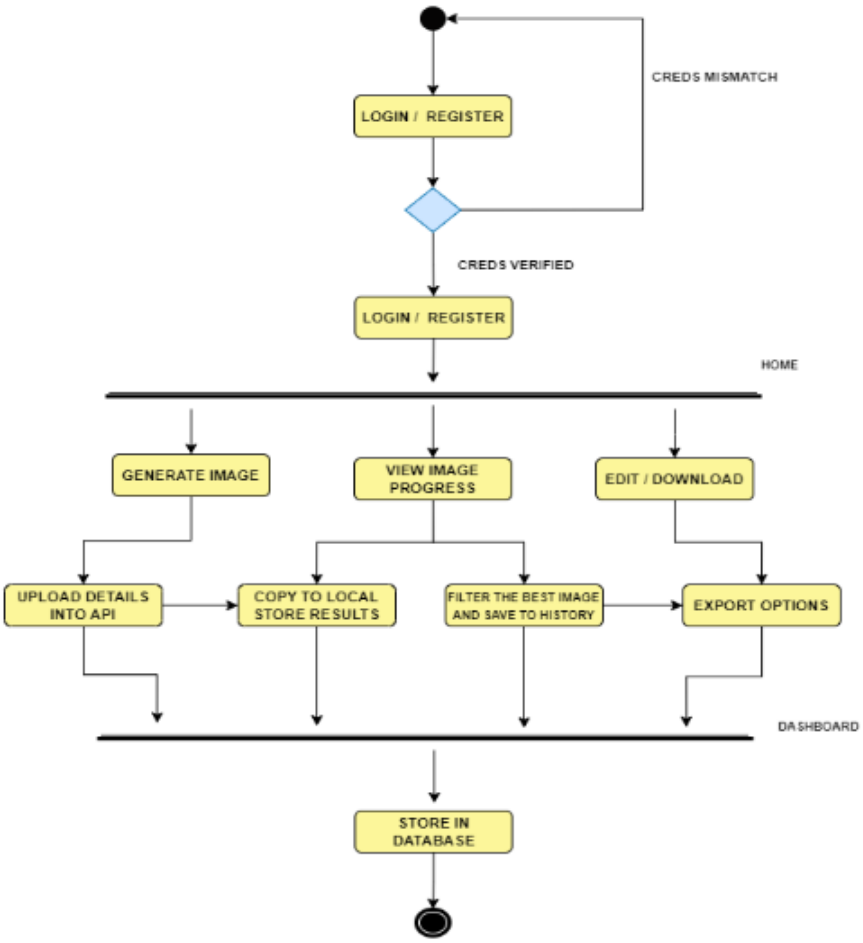


Fig - 17 Activity Diagram

6.2.2 Use CaseDiagram

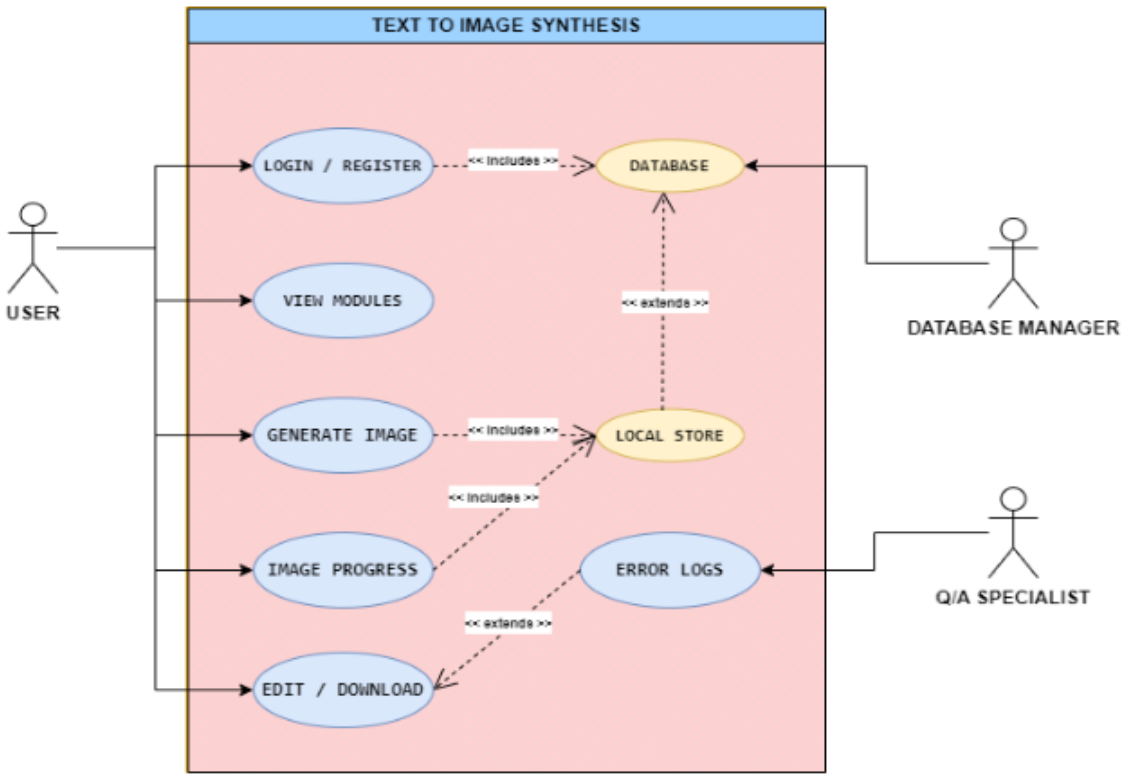


Fig - 18 Use Case Diagram

6.2.3 Scenarios

Normal scenarios are instances where the desired image is accurately and simply described in the text, without any ambiguity or exceptionality. In these circumstances, the model may produce a believable visual that matches the textual description.

The term "exceptional scenarios" describes situations where the textual description is convoluted, unclear, or contradictory, making it challenging for the model to provide a cogent image. These situations might entail explanations that leave room for several possible interpretations, peculiar objects or settings, or contradictory facts. In certain circumstances, the model may produce an image that partially matches the textual description. Models frequently need more advanced algorithms, larger amounts of training data, and a deeper comprehension of the intended meaning underlying the textual description to handle uncommon cases.

CHAPTER - 7

APPLICATION

Text-to-image synthesis has a wide range of applications, including:

1. Creating visuals for promotional products like product photos, billboards, and pamphlets.
2. E-commerce: creating product listings images for websites or mobile applications.
3. Gaming: creating graphics, such as characters, environments, and objects, for video games.
4. Concept visualization: Creating visual representations of abstract concepts, such as engineering designs or scientific models, to make them easier to comprehend and convey.
5. Automated image generation: By automating the process of creating images, less time and effort is needed.
6. Generative art is the creation of fresh media for the arts, like digital paintings and animations.
7. AI-powered Design: Developing AI-powered design applications that let users produce pictures based on text input.
8. Human-Computer Interaction: Enabling humans to engage with computers in a more instinctive and natural way would improve human-computer interaction.

CHAPTER - 8

IMPLEMENTATION

We have implemented a Text-to-Image synthesizer using various Python and deep learning libraries such as NumPy, PIL, Pandas, TensorFlow, and Keras. The workflow of our implementation can be described in the following steps:

- **Data Pre-processing:** The flowers image dataset, consisting of 8,189 RGB images, is pre-processed. Each image is resized and checked for the RGB color channel. The pre-processed images are saved as NumPy arrays in batches of 100 images to optimize the preprocessing time.
- **Text Processing:**
    - A collection of text files containing captions associated with images is processed. Each text file is read, and the captions are pre-processed by removing spaces and converting them to lowercase.
    - The average embedding for each caption is calculated using pre-trained GloVe embeddings.
    - The resulting caption embeddings are stored in a NumPy array for further use. Progress updates are provided during the processing of text files.
- **Concatenating Image Arrays:** Numpy arrays of images stored in a specified directory are concatenated. The code iterates over a list of image filenames, starting from the second index. Each filename is printed, and the corresponding image numpy array is loaded using np.load().
- **Training Set and Model Building:** A sample set of shuffled captions is created for training purposes. The generator and discriminator models are built.
- **Loss Functions:**
    - The cross_entropy variable is instantiated as an instance of the Binary_Crossentropy class from the tf.keras.losses module.
    - The discriminator_loss function calculates the loss for the discriminator in a GAN. It takes three inputs: real_image_real_text (real image-text pairs), fake_image_real_text (fake image-text pairs), and real_image_fake_text (real images with fake texts). It computes the loss based on the discriminator's predictions and the actual labels.

- The generator_loss function calculates the loss for the generator in a GAN. It takes fake_output as input, representing the discriminator's predictions on the generator's outputs (fake images). It computes the loss based on the discriminator's predictions and a target array of all ones.

- **Training Step:** The train_step function is defined and annotated with @tf.function to optimize its performance by "compiling" it. Within the function, a random seed is generated, and the generator and discriminator models are applied to the inputs. The losses are calculated using the defined loss functions. The gradients of the losses with respect to the trainable variables of the models are computed using tf.GradientTape. The gradients are used to update the trainable variables of the models using the respective optimizers. The generator and discriminator losses are returned for monitoring the training progress.

This summarizes the chronological order of the steps involved in our implementation of the Text-to-Image synthesizer.

CHAPTER - 9

TESTING

The testing summary report for the ImageCraft prompt variation and engineering test suite can be summarized as follows:

- The test suite consists of multiple test cases designed to validate the functionality of the ImageCraft prompt variation and engineering features. Each test case is assigned a unique Test Case and Scenario ID and includes a description of the test scenario, pre-conditions, test steps, test inputs, expected output, actual output, and test case status (pass/fail).
- During testing, all test cases were executed, and the actual outputs were compared against the expected outputs. The test suite covered various aspects of the ImageCraft prompt variation and engineering, including image processing, text generation, and model performance.
- Overall, the test suite yielded positive results with the majority of test cases passing successfully. However, a few test cases identified some minor issues that require further investigation and refinement. These issues will be addressed in future iterations to enhance the reliability and accuracy of the ImageCraft prompt variation and engineering features.

Following is the link to the spreadsheet of the test suite executed - CAPSTONE - TESTING - SUITE

Fig - 19 Test Suites

This pge is extracted due to viral text or high resolution image or graph.

Text to Image Synthesis CHAPTER - 10 RESULTS Upon training the dataset with the aforementioned dataset of 102 Oxford flowers which consists of 8190 images of flowers and their respective captions or image descriptions . By abiding with the workflow discussed previously , some of the key tasks that are undertaken are cleaning and organizing the dataset , employing a high power GPU to compute the training task . Amazon SageMaker Studio was employed as the platform to execute the workload and train the model utilizing GPU acceleration.To add to this we have even made a rough comparison between the the images generated from the base paper model to the implemented one and we have observed great improvement and clarity in details. The results obtained from the model training exhibited remarkable progress as the number of iterations increased. Notably, the clarity of the generated flower attributes and the level of detail steadily improved, following are some of the best results upon 400-500 epochs . Fig - 20 This flower is purple in color with oval shaped petals Fig - 21 The flower is yellow in color with oval shaped petals Fig - 22 Sample output for a red flower with long petals prompt Fig - 23 This is a white flower resembling jasmine Dept of ISE, BMSCE 2022-23 38

CONCLUSION

Recent advancements in generative adversarial networks have made the creation of visuals from text descriptions an intriguing research topic. In terms of visual realism, diversity, and semantic alignment, it is a flexible and straightforward technique for producing conditional images. Two challenges that the area still has to address are the production of high-resolution photographs with many objects and the development of adequate and reliable evaluation standards that correspond with human judgment.

Beginning in January 2021, developments in AI research have given rise to a profusion of deep-learning models that are able to create unique visuals from straightforward text cues, essentially expanding human imagination. Text-to-image technologies have been created by researchers at OpenAI, Google, Facebook, and other firms; these tools have not yet been made available to the general public. Similar models can be found online in the open source community and at smaller businesses like Midjourney. Despite the fact that AI is catching up in a number of fields, production-ready text to image synthesis will probably take a few more years of intensive research.

In conclusion, Text-to-Image Generation using GANs is a rapidly growing field that holds promise for a wide range of applications. The use of GANs has shown to be effective in synthesizing images from textual descriptions, and ongoing research is likely to lead to further improvements in the quality and consistency of the generated images.

REFERENCES

[1] Zizhao Zhang , Yuanpu Xie , Lin Yang, “Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network", https://doi.org/10.48550/arXiv.1802.09178

[2] Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. “Da-gan: Instance-level image translation by deep attention generative adversarial networks” . In CVPR. pages 5657-5666, 2018

[3] Hao Xu , Pengchuan Zhang , Qiuyuan Huang , Han Zhang , Zhe Gan, Xiaolei Huang , Xiaodong He, “AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adv Networks” , Lehigh University,https://doi.org/10.48550/arXiv.1711.10485

[4] D. M. A. Ayanthi and Sarasi Munasinghe, “Text-to-face generation using styleGAN2”, Department of Computer Science, Faculty of Science, University of Ruhuna, Wellamadama, Matara, Sri Lanka, David C.Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022pp. 49-64, 2022. CS & IT - CSCP 2022 May 21~22, 2022, Zurich, Switzerland , https://doi.org/10.48550/arXiv.2205.12512.

[5] Han Zhang,Tao Xu,Hongsheng Li,Shaoting Zhang, “StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks”, Rutger University,2017 IEEE International Conference on Computer Vision, https://doi.org/10.48550/arXiv.1612.03242.

[6] Wenbo Li,Pengchuan Zhang,Lei Zhang,Qiuyuan Huang, “Object-driven Text-to-Image Synthesis via Adversarial Training”, University at Albany, CVPR 2019, https://doi.org/10.48550/arXiv.1902.10740.

[7] Minfeng Zhu, Pingbo Pan, Wei Chen, Yi Yang, “DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis”, Center for Artificial Intelligence, University of Technology Sydney. 2019 IEEE/CVF Conference onComputer Vision and Pattern Recognition (CVPR), https://doi.org/10.48550/arXiv.1904.01310.

[8] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Jingwan Lu, Chen Fang, Fisher Yu “Texture GAN : Controlling Deep Image Synthesis with Texture Patches ”. 2018 IEEE/CVF

Conference on Computer Vision and Pattern Recognition, https://doi.org/10.48550/arXiv.1706.02823.

[9] Guojun Yin , Bin Liu1 , Lu Sheng, Nenghai Yu1 , Xiaogang Wang, Jing Shao "Semantics Disentangling for Text-to-Image Generation", University of Science and Technology of China, Key Laboratory of Electromagnetic Space Information, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), https://doi.org/10.48550/arXiv.1904.01480.

[10] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgant: Realistic image synthesis with stacked generative adversarial networks. TPAMI, 2018.

[11] Hao Dong, Simiao Yu , Chao Wu, Yike Guo, "Semantic Image Synthesis via Adversarial Learning", Imperial College London, Accepted to ICCV 2017 https://doi.org/10.48550/arXiv.1707.06873

[12] Luyang Huang ,Guocheng Niu, Jiachen Liu, Xinyan Xiao and Hua Wu Baidu Inc., Beijing, China, "DU-VLG: Unifying Vision-and-Language Generation via Dual Sequence-to-Sequence Pre-training", to appear at Findings of ACL 2022. https://doi.org/10.48550/arXiv.2203.09052

[13] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. arXiv preprint ArXiv:1611.04076, 2016.

[14] Y. Liu, Z. Qin, Z. Luo, and H. Wang. Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. arXiv preprint arXiv:1705.01908, 2017

[15] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML, pages 2642-2651, 2017