



e-ISSN: 2319-8753 | p-ISSN: 2320-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 9, Issue 12, December 2020

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

**Impact Factor: 7.512**

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)

# Data Exploration with Adversarial Autoencoders using GAN

Aftab Kazi<sup>1</sup>, Alisha Khatani<sup>1</sup>, Mahira Majgaonkar<sup>1</sup>, Dr. Vinayak Ashok Bharadi<sup>2</sup>

U.G. Student, Department of Information Technology, Finolex academy of Management and Technology, Ratnagiri, Maharashtra, India<sup>1</sup>

Associate Professor, Department of Information Technology, Finolex academy of Management and Technology, Ratnagiri, Maharashtra, India<sup>2</sup>

**ABSTRACT:** The Data Exploration with Adversarial Autoencoders - Clustering with AAE text to image generation using generative adversarial networks (GAN) is a deep learning model which can generate images from their corresponding text description. The image retrieval from input descriptions has been one of the most important applications with increasing numbers of videos on the internet. Although the text to image generation is a challenging work since semantic tasks between features of texts and videos exist. Conversion of information between text and image is a problem in artificial intelligence i.e. AI that connects natural language processing. So, we propose a system, that is a new retrieval framework based on the generative adversarial network (GAN) which will improve the performance by simple procedures. In the addition There are lot studies which are basically focused on generation of high-quality image. which semantically aligns with the given text description. Thorough experiments on two public benchmark datasets demonstrate the superiority of GAN over other representative state-of-the-art methods. Addition to the latent vector, the encoder will also generate a *categorical vector* or so-called *one-hot encoded* vector, so a vector with one of its values being 1 and all other values being 0. This vector will also be fed into the decoder for reconstruction so that it will be likely to also carry relevant information about the input data, basically some sort of additional *hint* for the reconstruction. The idea is that this hint will be the same for *similar* input data, and these similarities are what we want the encoder to find and encode into categories.

**KEYWORDS:** Adversarial Autoencoder (AAE), Generative Adversarial Network (GAN), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN).

## I. INTRODUCTION

In today's artificial intelligence world, image generation is evaluated progressively. For generating the images of 3d objects convolution neural networks (CNN) were used. Also, algorithms have been designed that can extract the information from images upon being queried, this visual question answering the machine was a state of art research in the field of image analysis. Conditional image generation could be achieved using pixel CNN architecture. Changing over common language content depictions into pictures is a stunning show of Deep Learning. Content order errands, for example, opinion investigation have been fruitful with Deep Recurrent Neural Networks that can take in discriminative vector portrayals from content. In another space, Deep Convolution GANs can blend pictures, for example, insides of rooms from an arbitrary commotion vector examined from an ordinary appropriation. The focal point is to associate advances in Deep RNN, enlivened by the possibility of Conditional-GANs. Contingent GANs work by contributing a one-hot class mark vector as contribution to the generator and discriminator notwithstanding the arbitrarily inspected commotion vector. These outcomes in higher preparing strength, all the more outwardly engaging outcomes, just as controllable generator yields. The contrast between conventional Conditional-GANs and the Text-to-Image model introduced is in the molding input. Rather than attempting to develop a meager visual ascribe descriptor to condition GANs, the GANs are molded on a book implanting learned with a Deep Neural Network. Notwithstanding building great content embeddings, making an interpretation from content to pictures is exceptionally multi-modular. The term 'multi-modular' is a significant one Revised Manuscript Received on April 25, 2020. \* Correspondence Author Kambhampati.Monica, CSE, V R Siddhartha Engineering College, Vijayawada, India. Email: monica.kambhampati@gmail.com Dr. Duvvada Rajeswara Rao, CSE, V R Siddhartha Engineering College, Vijayawada, India. Email: rajeshpitam@gmail.com to get comfortable with in Deep Learning research. This alludes to the way that there are a wide range of pictures of winged creatures which relate to the content depiction "feathered creature". Another model in discourse is that there are a wide range of accents and so forth that would bring about



various sounds compared to the content "flying creature". Multi-modular learning is additionally present in picture subtitling, (picture to-content). In any case, this is enormously encouraged because of the successive structure of content to such an extent that the model can anticipate the following word adapted on the picture just as the newly anticipated words. Multi-modular learning is generally troublesome, yet is made a lot simpler with the headway of GANs (Generative Adversarial Networks), this system makes a versatile misfortune work which is appropriate for multi-modular undertakings, for example, content-to-picture.

**A. Aim of the project**

The main aim of this project is to generate the image from text using generative adversarial network(GAN), which will help in many image processing sectors. The generation of images from natural language has many possible applications in the future once the technology is ready for commercial applications. that the autoencoder is trained, we can use it to generate class labels for all our data.

**B. Objectives of the Project**

The objectives of the systems development are:

- Attention-driven, multi stage refinement for fine-grained text to image generation using GAN.
- It will provide the deep attentional multi model and a complete semantic layout.
- The synthesization of different subregions of the image by paying attention to the relevant words in the natural language description.
- Data exploration after GAN encoder and clustering with different classes.

**C. Scope of the Project**

Generative Adversarial Networks abbreviated as GANs are evolutionary technology in the field of Deep Learning which can generate images from random noise.

Now, a day's people are very much interested in generation through machines like Image Generation, audio or video generation, etc. And these all are possible in this era of Deep Learning. Because of the advancement of Deep Learning technologies, we can generate anything we want. One good example is that: machines generate faces which do not exist yet this thing helps a lot in today's world. The scope of the proposed system is that it will generate a lot of images which also do not exist.

- image by paying attention to the relevant words in the natural language description.
- Data exploration after GAN encoder and clustering with different classes.

**II. LITERATURE SURVEY**

Title	Authors	Advantage	Disadvantage	Scope
MirrorGAN: Learning Text-to-image Generation by Redescription	Tingting Qiao, Jing Zhang, Dacheng Tao	Mirror GAN provide high-quality and visually realistic images	Guaranteeing semantic alignment of the generated image with the input text remains challenging	Able to generate fine scaled image upto a optimized computational resource
Object-driven Text-to-Image Synthesis via Adversarial Training	Wenbo Li, Pengchuan Zhang, Lei Zhang.	Obj-GANs that allow object-centered text-to-image synthesis for complex scenes, provide a two-step generation process.	While providing a semantic layout it takes the input as a sentence.	Able to refine the synthesized image with the help of object wise discriminator



Semantics Disentangling for Text-to-Image Generation	Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu	It provides a more natural and photo realistic image with high semantic layout	The model which is used here is very complex and poses a challenge at some stage while creating the realistic image.	Here the generated images can keep generation consistency under expression variants
StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks	Han Zhang1, Tao Xu, Hongsheng Li.	Provide a 256x256 high resolution image using stage 1 and stage 2 GAN, and also provide a high level semantic layout.	The major problem is that it works on the whole sentence. and process only the specific words as a global vectors means as a sentence	Able to generate high resolution images(256 x 256) with more photo realistic details
Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network	Z. Zhang, Y. Xie, and L. Yang	A novel end-to-end method that can directly model high-resolution image statistics and generate photographic images.	In most cases, the image features lack in matching with text features	Regularize mid-level representations and assist generator training to capture the complex image statistics.

### III. PROBLEM DEFINITION

#### A. Problem Statement

Text-to-image generation, i.e. generating an image given a text description, is a very challenging task. Translating an image from a given text description is an amazing thing in deep learning. Common text—conversion methods employ Generative Adversarial Networks to generate high-text-images, but the images produced do not always represent the meaning of the phrase provided to the model as input. Using a captioning network to caption generated images, we tackle this problem and exploit the gap between ground truth captions and generated captions to further enhance the network. Text-to-Image synthesis is a difficult problem with plenty of space for progress despite the current state-of-the-art results. Synthesized images from current methods give the described image a rough sketch but do not capture the true essence of what the text describes.

#### B. Existing System

In the existing system, first generate an initial image with rough shape and color, and then refine the initial image to a high-resolution one. The main problem in the existing system is, these methods heavily depend on the quality of initial images. If the initial image is not well initialized, the following processes can hardly refine the image to a satisfactory quality.

#### C. Disadvantages of Existing System

1. Depends on the quality of initial images.
2. Each word contributes a different level of importance.

#### D. Proposed System

A GAN is a generative model that is trained using two neural network models. One model is called the “generator” or “generative network” model that learns to generate new plausible samples. The other model is called the “discriminator” or “discriminative network” and learns to differentiate generated examples from real examples.

In our proposed system, we propose Generative Adversarial Network to generate high-quality images. The proposed method introduces a module to refine the image contents, when the initial images are not well generated. In this system we are using a GAN(generative adversarial network) that can synthesize fine-grained details at different parts of the image by paying attention to the relevant words in the natural language description. The main purpose of this project is to translate text to image using GAN, which produces better results than existing systems.



#### E. Advantages of Proposed System

1. Using this method we produce high quality images.
2. Helpful for photo editing.
3. Also we produce images for emojis.
4. This system is very helpful for different types of applications.

#### F. Methodology

This project we will develop using python. For this system we are using Tensorflow which is an open-source library designed for high performance numerical and scientific computations. It is one of the most widely used libraries for machine learning research. TensorFlow offers both low level and high-level APIs which make development flexible and allow fast iteration. In our project first we will first collect the dataset for the developing system. After that we will train the dataset and generate a module for proposed output. We will give input as an image with text and showing the result image is real or fake.

### IV. HARDWARE & SOFTWARE REQUIREMENTS

#### A. Hardware

1. Processor: Intel Core i5 or more.
2. RAM: 8GB or more.
3. Hard disk: 250 GB or more.

#### B. Software

1. Operating System : Windows 10, 7, 8.
2. Python.
3. Anaconda.
4. Spyder, Jupyter notebook
5. MYSQL

#### C. Technologies Used:

##### 1. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

##### 2. MySQL

MySQL is well known as the world's most widely used open-source database (back-end). It is the most supportive database for PHP as PHP-MySQL is the most frequently used open-source scripting database pair. The user-interface which WAMP, LAMP and XAMPP servers provide for MySQL is easiest and reduces our work to a large extent.

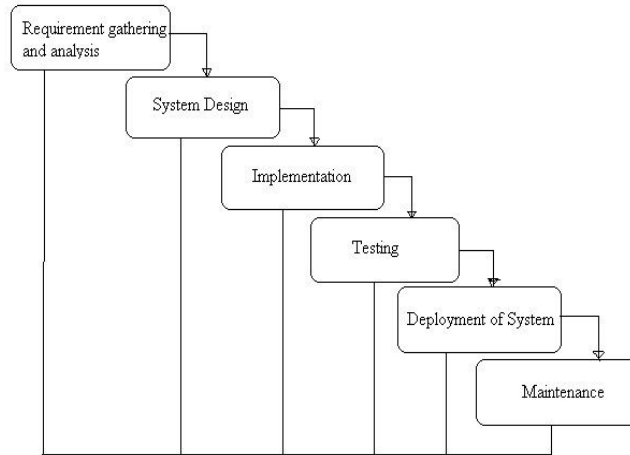
### V. PLANNING & ESTIMATION

#### A. Software development Life Cycle

The entire project spanned for a duration of 6 months. In order to effectively design and develop a cost-effective model the Waterfall model was practiced.



General Overview of "Waterfall Model"



**B. Requirement gathering and Analysis phase**

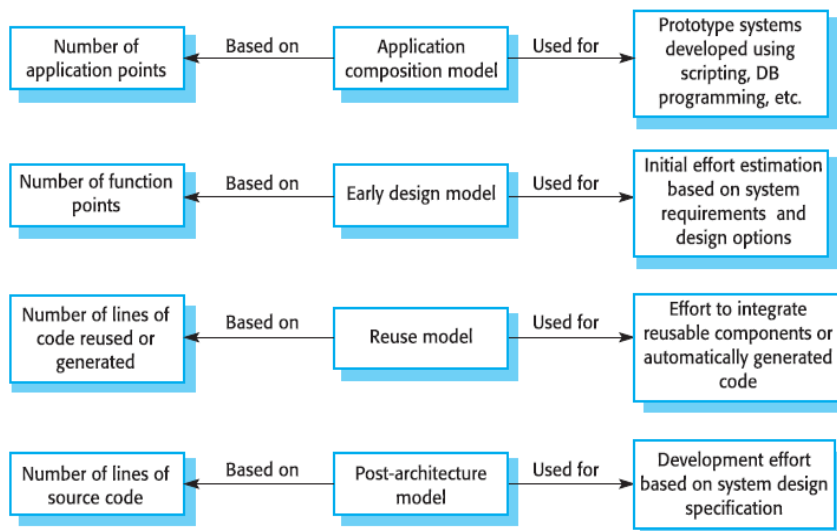
This phase started at the beginning of our project, we had formed groups and modularized the project. Important points of consideration were

1. Define and visualize all the objectives clearly.
2. Gather requirements and evaluate them

Consider the technical requirements needed and then collect technical specifications of various peripheral components (Hardware) required.

3. Analyze the coding languages needed for the project.
4. Define coding strategies.
5. Analyze future risks / problems.
6. Define strategies to avoid these risks else define alternate solutions to these risks.
7. Check financial feasibility.
8. Define Gantt charts and assign time span for each phase.

**C. Cost Estimation**



Cost estimation is done using cocomo model

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
<b>Product attributes</b>						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65



<b>Hardware attributes</b>						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
<b>Personal attributes</b>						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
<b>Project attributes</b>						
Use of software tools	1.24	1.10	1.00	0.91	0.82	
Application of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

The Intermediate Cocomo formula now takes the form:

$$E = a_i (\text{kloc})^{b_i} \cdot \text{EAF}$$

Using above calculation we found that the total time period of the project is around 6 months, the per month cost comes out to be Rs.12, 000/- so the total comes to be Rs.72, 000/-

**D. Feasibility Study**

This system is feasible for all departments like education, industry and clinic etc. and this system will use without experts in that field anyone who has knowledge about using online services which will help to use this system. Any generation of people will use this system in a Personal Computer.

**E. Technical Feasibility**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Is the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within the latest technology. Though the technology may become obsolete after some period of time, due to the fact that never version of the same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java. The project is technically feasible for development.

**F. Economic Feasibility**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the system is economically possible for development.

### G. Behavioural Feasibility

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

### H. Risk Analysis Process

Regardless of the prevention techniques employed, possible threats that could arise inside or outside the organization need to be assessed. Although the exact nature of potential disasters or their resulting consequences are difficult to determine, it is beneficial to perform a comprehensive risk assessment of all threats that can realistically occur to the organization. Regardless of the type of threat, the goals of business recovery planning are to ensure the safety of customers, employees and other personnel during and following a disaster.

The relative probability of a disaster occurring should be determined. Items to consider in determining the probability of a specific disaster should include, but not be limited to: geographic location, topography of the area, proximity to major sources of power, bodies of water and airports, degree of accessibility to facilities within the organization, history of local utility companies in providing uninterrupted services, history of the area's susceptibility to natural threats, proximity to major highways which transport hazardous waste and combustible products. Potential exposures may be classified as natural, technical, or human threats. Examples include:

- Natural Threats: internal flooding, external flooding, internal fire, external fire, seismic activity, high winds, snow and ice storms, volcanic eruption, tornado, hurricane, epidemic, tidal wave, typhoon.
- **Technical Threats:** power failure/fluctuation, heating, ventilation or air conditioning failure, malfunction or failure of CPU, failure of system software, failure of application software, telecommunications failure, gas leaks, communications failure, nuclear fallout.
- Human Threats: robbery, bomb threats, embezzlement, extortion, burglary, vandalism, terrorism, civil disorder, chemical spill, sabotage, explosion, war, biological contamination, radiation contamination, hazardous waste, vehicle crash, airport proximity, work stoppage (Internal/External), computer crime.

All locations and facilities should be included in the risk analysis. Rather than attempting to determine exact probabilities of each disaster, a general relational rating system of high, medium and low can be used initially to identify the probability of the threat occurring. The risk analysis also should determine the impact of each type of potential threat on various functions or departments within the organization. A Risk Analysis Form, found here(PDF Format), can facilitate the process. The functions or departments will vary by type of organization. The planning process should identify and measure the likelihood of all potential risks and the impact on the organization if that threat occurred.

To do this, each department should be analyzed separately. Although the main computer system may be the single greatest risk, it is not the only important concern. Even in the most automated organizations, some departments may not be computerized or automated at all. In fully automated departments, important records remain outside the system, such as legal files, PC data, software stored on diskettes, or supporting documentation for data entry. The impact can be rated as: 0= No impact or interruption in operations, 1= Noticeable impact, interruption in operations for up to 8 hours, 2= Damage to equipment and/or facilities, interruption in operations for 8 - 48 hours, 3= Major damage to the equipment and/or facilities, interruption in operations for more than 48 hours. All main office and/or computer center functions must be relocated. Certain assumptions may be necessary to uniformly apply ratings to each potential threat.

Following are typical assumptions that can be used during the risk assessment process:

1. Although impact ratings could range between 1 and 3 for any facility given a specific set of circumstances, ratings applied should reflect anticipated, likely or expected impact on each area.
2. Each potential threat should be assumed to be "localized" to the facility being rated.
3. Although one potential threat could lead to another potential threat (e.g., a hurricane could spawn tornados), no domino effect should be assumed.
4. If the result of the threat would not warrant movement to an alternate site(s), the impact should be rated no higher than a "2."
5. The risk assessment should be performed by facility. To measure the potential risks, a weighted point rating system can be used.

### I. Functional requirement

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be



calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

Functional requirements are supported by non-functional requirements (also known as *quality requirements*), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>". The plan for implementing *functional* requirements is detailed in the system *design*. The plan for implementing *non-functional* requirements is detailed in the system *architecture*.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the *application architecture* of a system, while non-functional requirements drive the *technical architecture* of a system.

### J. Non-functional requirement

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing *functional* requirements is detailed in the system *design*. The plan for implementing *non-functional* requirements is detailed in the system *architecture*.

In general, functional requirements define what a system is supposed to *do* whereas non-functional requirements define how a system is supposed to *be*. Functional requirements are usually in the form of "system shall do <requirement>", while non-functional requirements are "system shall be <requirement>".

Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

## VI. TESTING

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

### A. Black box testing

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

- **Specification-based testing**

Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks.

- **Advantages and disadvantages**

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not. *But*, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.



## B. White box testing

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

The following types of white box testing exist:

- API testing (application programming interface) - Testing of the application using Public and Private apis
- Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
- Fault injection methods - improving the coverage of a test by introducing faults to test code paths
- Mutation testing methods
- Static testing - White box testing includes all static testing

## C. Code completeness evaluation

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

- *Function coverage*, which reports on functions executed
- *Statement coverage*, which reports on the number of lines executed to complete the test

They both return code coverage metric, measured as a percentage.

## D. Integration testing

Integration testing is any type of software testing, which seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.

## E. Acceptance testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. Before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own HW, is known as user acceptance testing (UAT).

## VII. DESIGN & IMPLEMENTATION

A **flowchart** is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

### Start and end symbols

Represented as circles, ovals or rounded (fillet) rectangles, usually containing the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product".

### Arrows

Showing "flow of control" An arrow coming from one symbol and ending at another symbol represents that control passes to the symbol the arrow points to. The line for the arrow can be solid or dashed. The meaning of the arrow with dashed line may differ from one flowchart to another and can be defined in the legend.

### Generic processing steps

Represented as rectangles Examples: “Add 1 to X”; “replace identified part”; “save changes” or similar.

### Subroutines

Represented as rectangles with double-struck vertical edges; these are used to show complex processing steps which may be detailed in a separate flowchart. Example: PROCESS-FILES. One subroutine may have multiple distinct entry points or exit flows (see coroutine); if so, these are shown as labeled ‘wells’ in the rectangle, and control arrows connect to these ‘wells’.

### Input/output

Represented as a parallelogram Examples: Get X from the user; display X **Prepare conditional** Represented as a hexagon Shows operations which have no effect other than preparing a value for a subsequent conditional or decision step (see below).

### Conditional or decision

Represented as a diamond (rhombus) showing where a decision is necessary, commonly a Yes/No question or True/False test. The conditional symbol is peculiar in that it has two arrows coming out of it, usually from the bottom point and right point, one corresponding to Yes or True, and one corresponding to No or False. (The arrows should always be labeled.) More than two arrows can be used, but this is normally a clear indicator that a complex decision is being taken, in which case it may need to be broken-down further or replaced with the “predefined process” symbol.

### Junction symbol

Generally represented with a black blob, showing where multiple control flows converge in a single exit flow. A junction symbol will have more than one arrow coming into it, but only one going out. In simple cases, one may simply have an arrow point to another arrow instead. These are useful to represent an iterative process (what in Computer Science is called a loop). A loop may, for example, consist of a connector where control first enters, processing steps, a conditional with one arrow exiting the loop, and one going back to the connector. For additional clarity, wherever two lines accidentally cross in the drawing, one of them may be drawn with a small semicircle over the other, showing that no junction is intended.

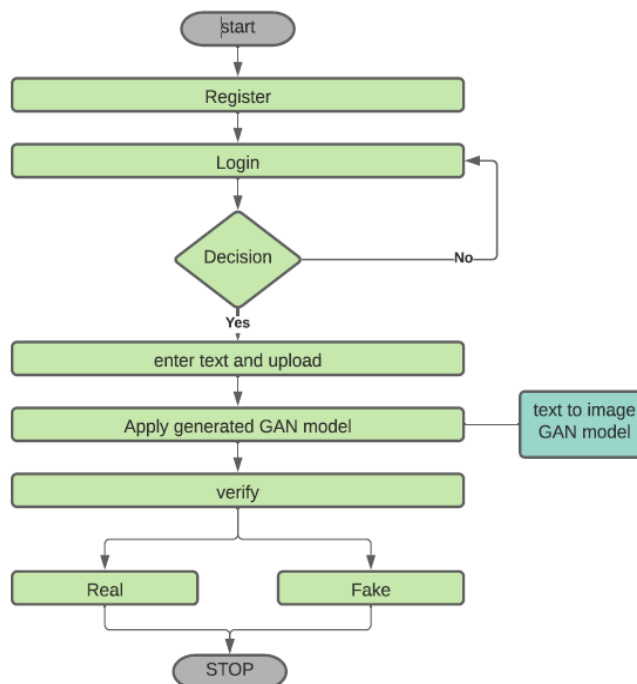
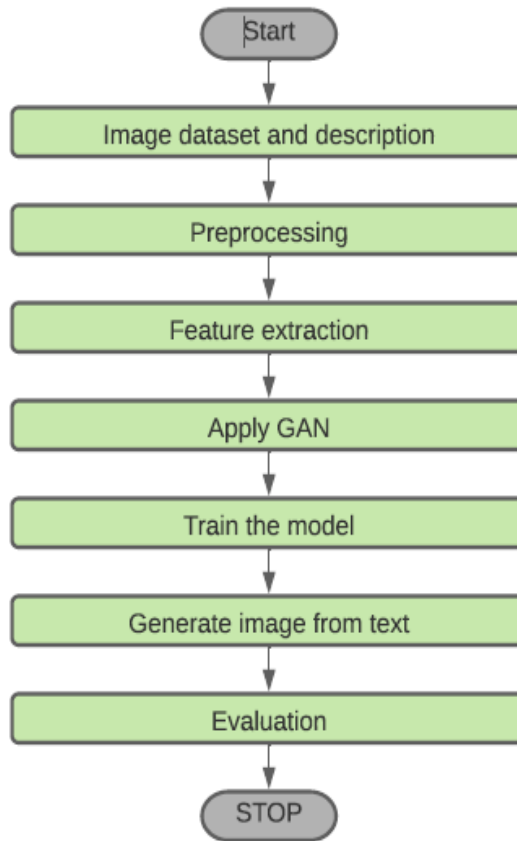
### Labeled connectors

Represented by an identifying label inside a circle. Labeled connectors are used in complex or multi-sheet diagrams to substitute for arrows. For each label, the “outflow” connector must always be unique, but there may be any number of “inflow” connectors. In this case, a junction in control flow is implied.

### Concurrency symbol

Represented by a double transverse line with any number of entry and exit arrows These symbols are used whenever two or more control flows must operate simultaneously. The exit flows are activated concurrently when all of the entry flows have reached the concurrency symbol. A concurrency symbol with a single entry flow is a *fork*; one with a single exit flow is a *join*. It is important to remember to keep these connections logical in order. All processes should flow from top to bottom and left to right.

Flow chart





### System Implementation

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

### Implementation Methods

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system.

The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flaw in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another common method is a direct cut over from the existing manual system to the computerized system. The change may be within a week or within a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization and the personnel will be piloting the system and changes can be made as and when required. But this method is less preferable due to the loss of the entirety of the system.

### Implementation Plan

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps.

- List all files required for implementation.
- Identify all data required to build new files during the implementation.
- List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

### DFD

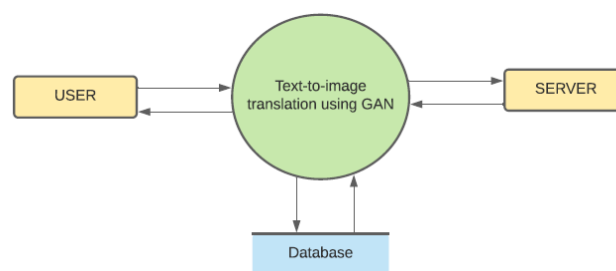
A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then exploded to show more detail of the system being modeled.

### Symbols

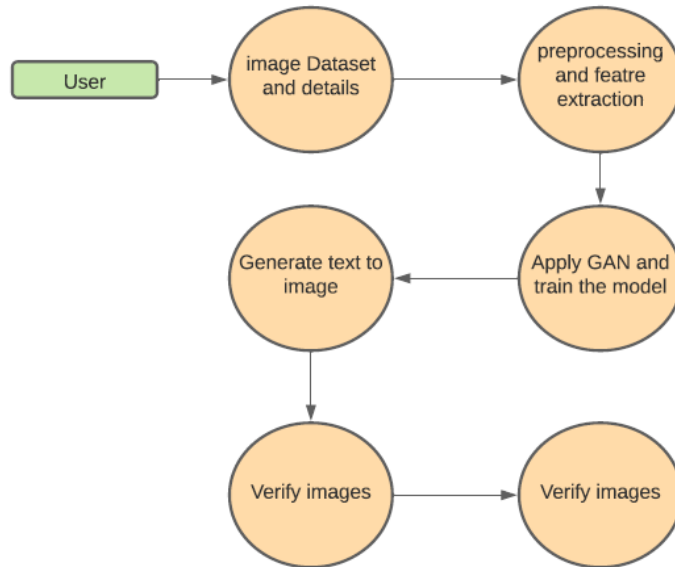
#### The four components of a data flow diagram (DFD) are

- External Entities/Terminators are outside of the system being modeled. Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it.
- Processes modify the inputs in the process of generating the outputs
- Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

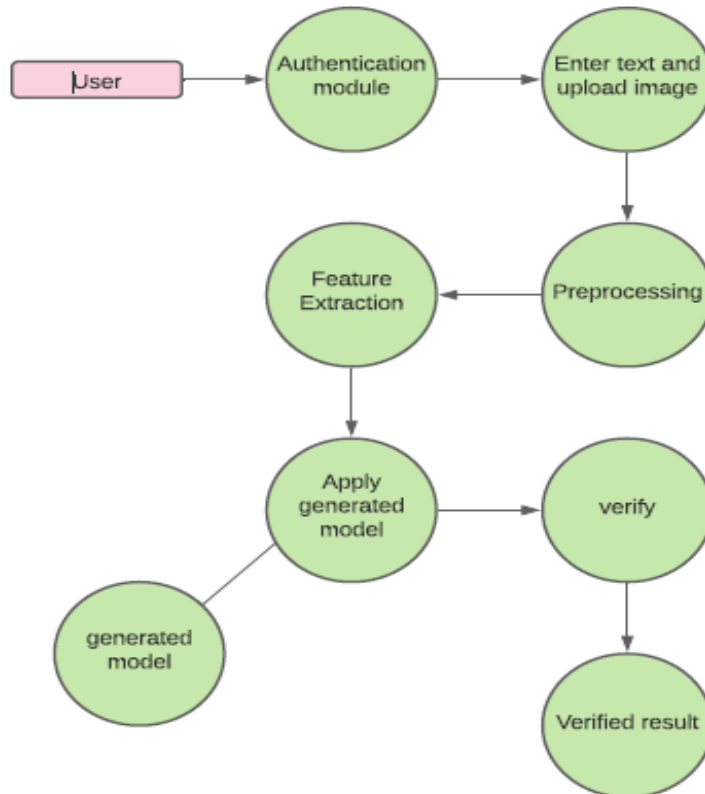
### 0-Level DFD



DFD level 1

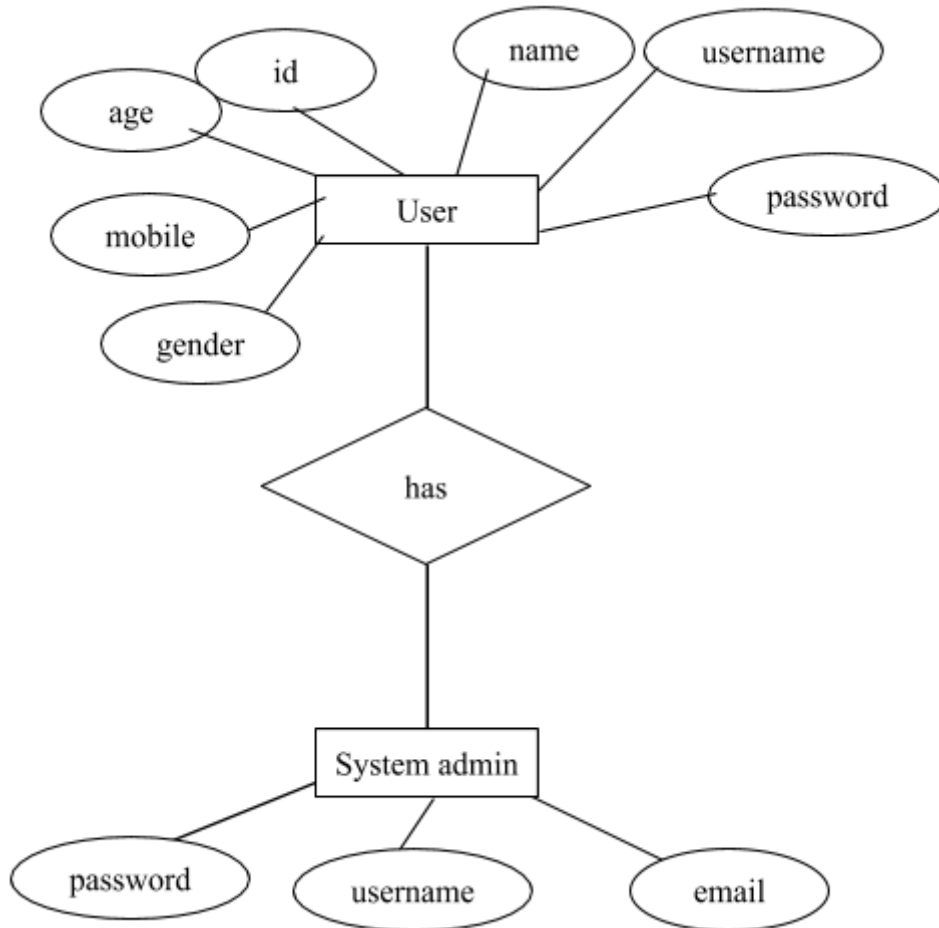


DFD level 2

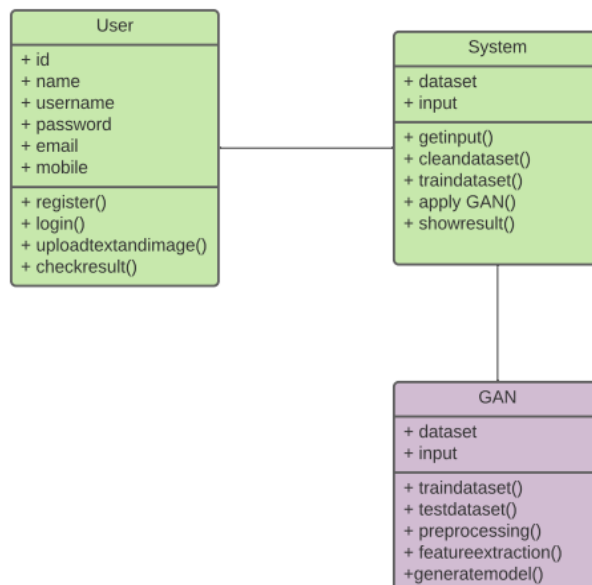




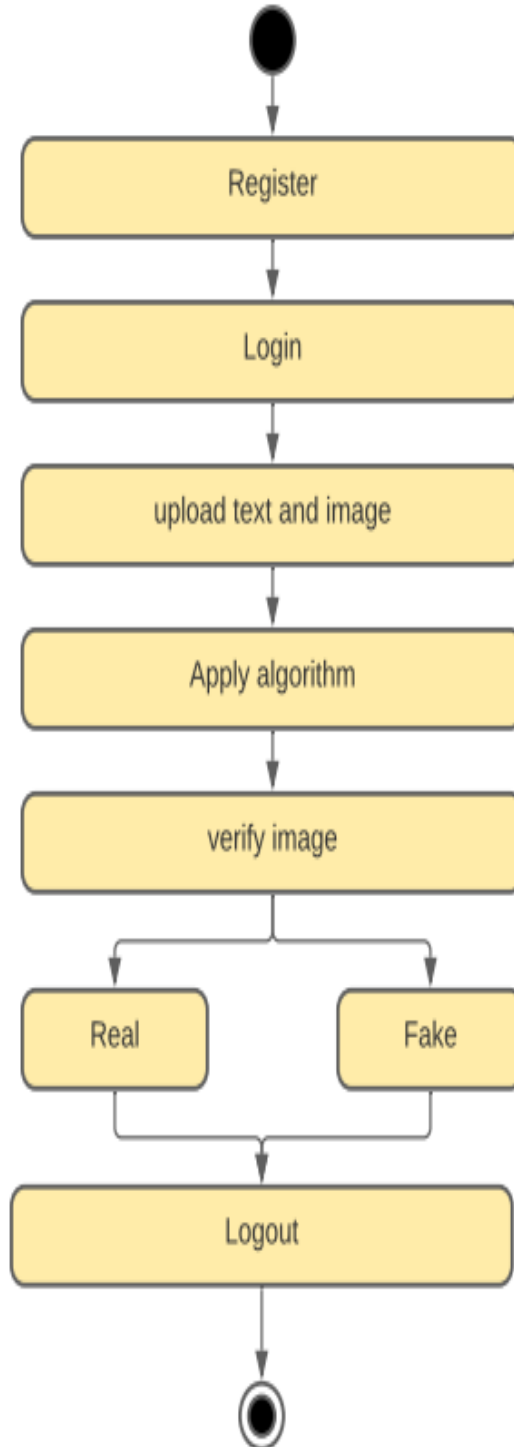
**E-R Diagram**



**Class Diagram**

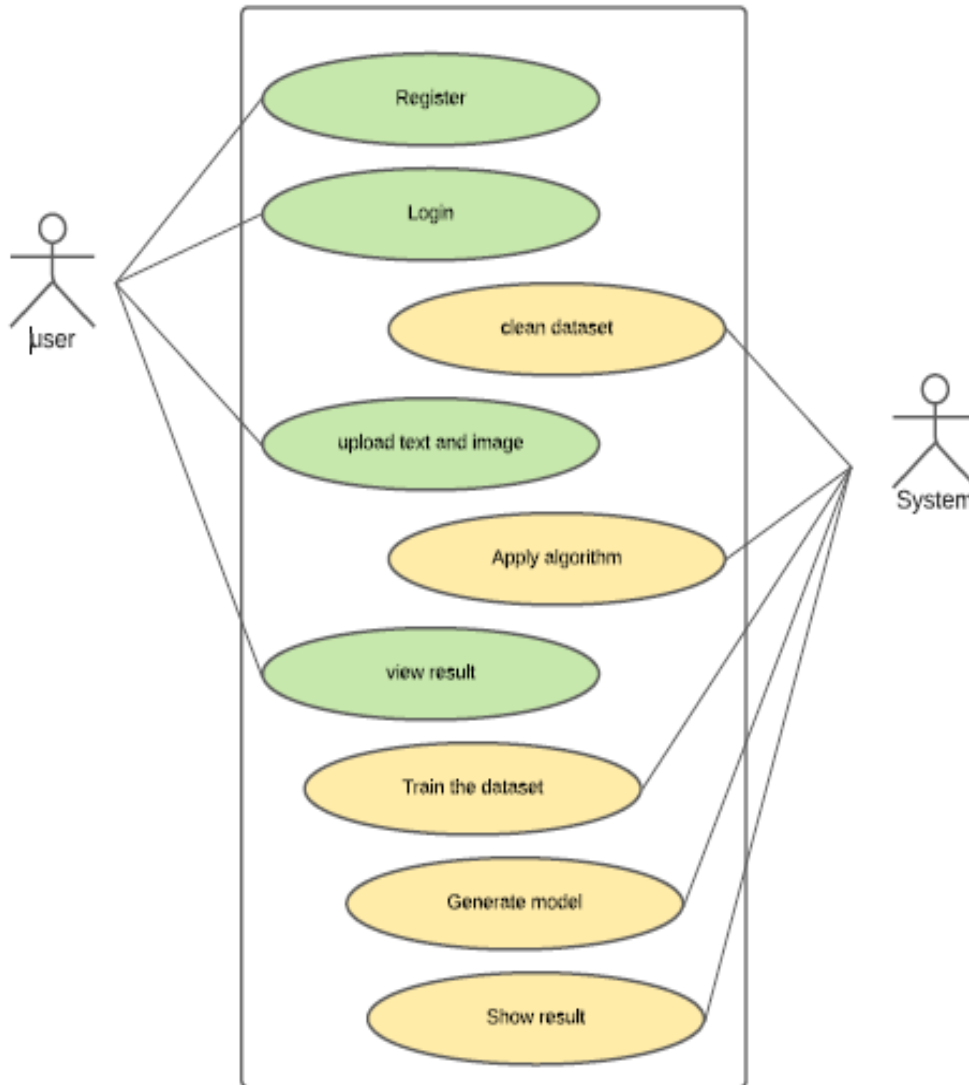


Activity Diagram





**Use Case Diagram**



**VIII. MOTIVATION**

Now, a day's people are very much interested in generation through machines like Image Generation, audio or video generation, etc. And these all are possible in this era of Deep Learning. Because of the advancement of Deep Learning technologies, we can generate anything we want. One good example is that: machines generate faces which do not exist yet this thing helps a lot in today's world. The scope of the proposed system is that it will generate a lot of images which also do not exist.

**IX. FUTURE MODIFICATIONS**

In future it is intended to improve the system developed by increasing the dataset to make it more accurate. update more cleaned dataset and check the accuracy of each algorithm, and use higher accuracy algorithms to improve the accuracy. In future we also make multiple applications like image to image translation, image to emojis, photo editing etc.

**X. CONCLUSION**

In this project we are making the model of text to image translation which will help in multiple applications like photo editing, image to emoji translation. Our Proposed System helps to get fine grained text to image synthesis

and generate high quality image through a multi-stage process. Additionally, we also compute the text-image matching loss for increasing the accuracy of the generator. Compared to existing text to image models, our model generates higher resolution images with more photo-realistic details.

#### REFERENCES

- [1] Dong H, Zhang J, McIlwraith D, Guo Y. I2t2i: Learning text to image synthesis with textual data augmentation. In2017 IEEE International Conference on Image Processing (ICIP) 2017 Sep 17 (pp. 2015- 2019). IEEE.
- [2] Muhammad A, Ahmad F, Martinez-Enriquez AM, Naseer M, Muhammad A, Ashraf M. Image to Multilingual Text Conversion for Literacy Education. In2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) 2018 Dec 17 (pp. 1328-1332). IEEE.
- [3] Zhang C, Peng Y. Stacking vae and gan for context-aware text-toimage generation. In2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM) 2018 Sep 13 (pp. 1-5). IEEE.
- [4] Rampurkar VV, Shah SK, Chhajed GJ, Biswas SK. An approach towards text detection from complex images using morphological techniques. In2018 2nd International. Conference on Inventive Systems and Control (ICISC) 2018 Jan 19 (pp. 969-973). IEEE.
- [5] Yanagi R, Togo R, Ogawa T, Haseyama M. Query is GAN: Scene Retrieval With Attentional Text-to-Image Generative Adversarial Network. IEEE Access. 2019 Oct 14;7:153183-93.
- [6] Xu T, Zhang P, Huang Q, Zhang H, Gan Z, Huang X, He X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 1316-1324).
- [7] Wang X, Gupta A. Generative image modeling using style and structure adversarial networks. InEuropean Conference on Computer Vision 2016 Oct 8 (pp. 318-335). Springer, Cham.
- [8] Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H. Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396. 2016 May 17.
- [9] Cha M, Gwon Y, Kung HT. Adversarial nets with perceptual losses for text-to-image synthesis. In2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP) 2017 Sep 25 (pp. 1-6). IEEE.
- [10] Dai B, Fidler S, Urtasun R, Lin D. Towards diverse and natural image descriptions via a conditional gan. InProceedings of the IEEE International Conference on Computer Vision 2017 (pp. 2970-2979).



**INNO**  **SPACE**  
SJIF Scientific Journal Impact Factor

**Impact Factor:**  
**7.512**

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
**INDIA**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details