A PROJECT REPORT ON

Image Synthesis from text description using GAN

SUBMITTED BY

Rushikesh Bhosle (B150024222)
Saurabh Badgujar (B150024210)
Jagannath Gaidhani (B150024238)

UNDER THE GUIDANCE OF
Prof. Seema Patil

Department Of Computer Engineering
MAEERs MAHARASHTRA INSTITUTE OF TECHNOLOGY
Kothrud, Pune 411 038
2018-2019

Department Of Computer Engineering
MAEERs MAHARASHTRA INSTITUTE OF TECHNOLOGY PUNE

C E R T I F I C A T E

This is to certify that

Rushikesh Bhosle (B150024222)

Saurabh Badgujar (B150024210)

Jagannath Gaidhani (B150024238)

of B. E. Computer successfully completed project in

Image Synthesis from text description using GAN

to my satisfaction and submitted the same during the academic year 2018-2019 towards the partial fulfillment of degree of Bachelor of Engineering in Computer Engineering of Pune University under the Department of Computer Engineering , Maharashtra Institute of Technology, Pune.

Prof. Seema Patil
(Project Guide)

Dr. V.Y.Kulkarni
(Head of Computer Engineering Department)

Place: Pune

Date:

ACKNOWLEDGEMENT

We take this opportunity to express our sincere appreciation for the cooperation given by Dr. V. Y. Kulkarni, Head Of Department (Department of Computer Engineering) and need a special mention for all the motivation and support.

We are deeply indebted to my guide Prof. Seema Patil for completion of our project for which she has guided and helped us going out of the way.

For all efforts behind the Project report, We would also like to express our sincere appreciation to staff of department of Computer Engineering, Maharashtra Institute of Technology Pune, for their extended help and suggestions at every stage.

Rushikesh Bhosle
(Exam Seat No: B150024222)

Jagannath Gaidhani
(Exam Seat No: B150024238)

Saurabh Badgujar
(Exam Seat No: B150024210)

Contents

List of Figures

Abstract

Generating images from natural language is one of the primary applications of recent conditional generative models. Besides testing our ability to model conditional, highly dimensional distributions, text to image synthesis has many exciting and practical applications such as photo editing or computer-aided content creation.In the machine,DC-GAN model architecture has been used for the text to image synthesis.For the feature vector representation deep convolutional and recurrent network is used.The generator and discriminator model are trained on the Oxford-102 flowers dataset and the Caltech CUB-200 birds dataset.

Keywords:Image Synthesis,DC-GAN,generator,discriminator,convolutional and recurrent neural network

Chapter 1

INTRODUCTION

1.1 Motivation

Generating photo realistic image is a challenging problem in computer vision which is opposite of image description problem and has diverse practical applications. With recent advances in deep learning, machine learning algorithms have evolved to such an extent that they can compete with humans in some tasks, such as image classification, image description. But we still cannot say that those algorithms have true intelligence since knowing how to do something does not necessarily mean understanding something, and it is critical for a truly intelligent agent to understand its tasks. Problems like photo realistic image synthesis using text description need machine to understand the statements and work according. So for this approach a new framework of Generative Adversarial Network(GAN) model that learn to discover the essence of data and find a best distribution to represent it. GAN is able to generate better synthetic images than previous generative models, and since then it has become one of the most popular research areas.

1.2 Problem Definition

In this project we are going to deal with the problem of generating image from text description using GAN(generative adversarial network).It consists of two stages the first stage deals with extracting features from text description and in the second stage the image is generated from the features extracted. For feature description two method are provided ,the first is through speech and second one is directly through text. Image generation using text can divided in many modules. The process of image generation is carried out by the generator network and the process of evaluation whether the image is good enough according to the features described in text ,is carried out by the discriminator.

Chapter 2

LITERATURE SURVEY

2.1 Generative Adversarial Nets:

In the paper" Generative Adversarial Nets"[6],a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G,has been proposed.The training procedure for G is to maximize the probability of D making a mistake.

Advantages:

a) The model uses generotor as well as the discriminator which has low benefits of piecewise linear units and high benefits of piecewise linear units respectively.
b) Model can represent very sharp, even degenerate distributions.

Disadvantages:

a) Proper synchronization should be there between the generator and discriminator to avoid Helvetica scenario.
b) Its hard to learn to generate discrete data, like text.
c) Compared to Boltzmann machines, its hard to do things like guess the value of

one pixel given another pixel. GANs are really trained to do just one thing, which is generate all the pixels in one shot.

2.2 StackGAN: Text to Photorealistic Image Synthesis with Stacked Generative Adversarial networks:

In the paper"StackGAN: Text to Photorealistic Image Synthesis with Stacked Generative Adversarial networks"[2] ,a model with two Step StackGAN to generate photorealistic images conditioned on text descriptions has been developed. The Stage-I GAN sketches the primitive shape and colors of the object based on the given text description, yielding Stage-I low-resolution images. The Stage-II GAN takes Stage-I results and text descriptions as inputs, and generates high-resolution images with photo-realistic details. The highlights of the paper are the Stacked GAN with conditional Augmentation and the sketch refinement process.

Advantages:

a) Compared to existing text-to-image generative models, method generates higher resolution images.

b)The result can be improved by using more Stages in the model.

Disadvantages:

a) The results are restricted to the flowers and birds dataset.

b) The model used is the more complex as compared to simple GAN model as 2 Stage model is used.

2.3 Generative Adversarial Networks: An Overview

In the paper"Generative Adversarial Networks: An Overview"[8],an overview of GANs for the signal processing community, drawing on familiar analogies and concepts is provided. In addition to identifying different methods for training and constructing GANs, a point to remaining challenges in their theory and application is also mentioned.

Advantages:

a)More understanding way for training the different GAN networks has been explained.

b)Different types of applications of GAN has also been explained in this work.

Disadvantages:

a) Many new applications can be derived from the GAN.

b) latest GANs like WGAN,etc training is not explained in the work.

2.4 UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS:

In the paper"UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS"[5],bridging the gap between the success of CNNs for supervised learning and unsupervised learning has been done.A class of CNNs called deep convolutional generative adversarial networks (DCGANs),that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning has been introduced.

Advantages:

a) A more stable set of architectures for training generative adversarial network has been proposed i.e DCGAN(DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS).

b)Adversarial networks learn good representations of images for supervised learning and generative modeling using DCGAN.

Disadvantages:

a) Still some forms of model instability remaining - we noticed as models are trained longer they sometimes collapse a subset of filters to a single oscillating mode.

b) Further work is needed to tackle this form of instability.

2.5 Comparative study on Generative Adversarial Networks:

In the paper"Comparative study on Generative Adversarial Networks"[4] ,a comparative study of different generative adversarial networks like VANILLA GAN,CGAN, GRAN,BIGAN,etc is done. Comparison between the basic GAN model and future advanced GAN reveal that the most recent versions of GAN are more robust and has many applications.An empirical analysis can be performed on all the versions using benchmark datasets such as MNIST, CIFAR10 and ImageNet which provides the scope for further enhancement of models.

Advantages:

a) Comparison is done on the basis methodology, architecture and performance.

b) Difference between latest GAN and basic GAN is more precisely explained.

Disadvantages:

a) Latest GANs like Boundary-Seeking Generative Adversarial Networks ,Wasserstein Generative Adversarial Networks(WGAN),etc are not compared.

b) More accurate comparison of the model can be done using more attributes.

2.6 An Introduction to Image Synthesis with Generative Adversarial Nets:

In the paper"An Introduction to Image Synthesis with Generative Adversarial Nets"[3] ,overview of the methods used in image synthesis with GAN is provided and it point out strengths and weaknesses of current methods.Three main approaches for image synthesis i.e direct methods,hierarchical methods and iterative method has be explained in the paper.Detailed discussion on the two important methods like text to image synthesis and image to image translation is also done.

Advantages:

a)For text-to-image synthesis, current methods work well on datasets where each image contains single object such as CUB and Oxford-102

b)Evaluation matrix for the synthetic image is generated which infers that power of GAN largely lies in its discriminators acting as a learned loss function, which makes the model perform better on tasks

Disadvantages:

a)The performance on complex datasets such as MSCOCO is much worse as compaired to CUB and Oxford-102.

b) More complicated objects like human being in a room is really difficult to synthesis for the the simple GAN model.

2.7 Learning Deep Representations of Fine-Grained Visual Description

In the paper "Learning Deep Representations of Fine-Grained Visual Description"[7],a visually-discriminative vector representation of text descriptions is generated by using deep convolutional and recurrent text en-coders that learn a correspondence function with images.A natural language model from scratch; i.e. without pre-training and only consuming words and characters is being used for the same.

Advantages:

a)Natural language provides a flexible and compact way of encoding only the salient visual aspects for distin- guishing categories.

b)By training on raw text, our model can do inference on raw text as well, providing humans a familiar mode both for annotation and retrieval.

c)Model achieves strong performance on zero-shot text-based image retrieval and significantly outperforms the attribute-based state-of-the-art for zero-shot classification.

Disadvantages:

a)The performance on the model decreases as complex query text are provided to system.

b) Model has high performance result for only CUB dataset.For rest other the the results are moderate.

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

3.1.1 Project Scope

The system under development will provide a way to generate image from it's text description.Generating images from natural language is one of the primary applications of recent conditional generative models.Text to image synthesis has many exciting and practical applications such as computer-aided content creation. Recent progress has been made using Generative Adversarial Networks (GANs).Image synthesis using GAN has many applications such as automatic map creation in games or generating images from its text description etc...

3.1.2 User Class and Characteristics :

In this system there are 4 main entities :

- **User Interface:** User Interface basically provides a way for user to give input and select format of input.input can be a text or audio description of the image.

- **Text encoder:** Input given to the system is going to be in text format.The text needs to be encoded in certain format to get features from the text.
- **Generator Neural Network:** The Generator neural network is going to generate the image based on the text description of the image.Generator Neural Network is the main component of this generative model which works in opposite way than that of general model, other models predict the label of the test case based on the features.But in case of generative models it tries to predict the features based on the label.
- **Discriminator Neural Network:** Discriminator Neural Network will try to recognize whether the image is real or fake.real image is the image already existing in the dataset and fake image is the image that will be generated by the generator.

Assumptions and Dependencies:

- Initially the system will be trained on only one type of dataset probably bird dataset or flower dataset,so it will be able to generate images related to only those domains.
- Types of images that can be generated by the model will depend upon the dataset on which it is trained and the availability of dataset.

3.1.3 Functional Requirements :

- **Input Panel:** The input panel will allow user to choose between audio or text mode.And user will give input through it.
- **Text encoder:** In order to generate the image, features of the image are needed. text encoder will extract features of the image from the text description of the image.

- **Generator:** Generator network will generate the images from the text description of the image,this images will be feed to Discriminator network.
- **Discriminator:** Discriminator will try to detect whether image is from existing dataset or it is generated by the generator.

3.2 NON FUNCTIONAL REQUIREMENTS:

Performance:

Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes system safety when it is overloaded.

Reliability:

Reliability defines how likely it is for the software to work without failure for a given period of time. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components. To measure software reliability, you can count the percentage of operations that are completed correctly or track the average period of time the system runs before failing.

Availability:

Availability is gauged by the period of time that the system functionality and services are available for use with all operations. So, scheduled maintenance periods directly influence this parameter. And it is important to define how the impact of maintenance can be minimized. When writing the availability requirements, the team has to define the most critical components of the system that must be available at all time. You should also prepare user notifications in

case the system or one of its parts becomes unavailable.

Security:

Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information.

3.2.1 Safety Requirements

- **Goal :**

  A goal is a statement of the importance of achieving a desired target regarding some behavior, datum, characteristic, interface, or constraint. It is above the level of a policy and not sufficiently formalized to be verifiable.

- **Policy :**

  A policy is any strategic decision that establishes a desired goal.

- **Requirement :**

  A requirement is any mandatory, externally observable, verifiable (e.g., testable), and validatable behavior, datum, characteristic, or interface.

3.2.2 Software Quality Attributes

- **Reliability:**

Measure if product is reliable enough to sustain in any condition. Should give consistently correct results. Product reliability is measured in terms of working of project under different working environment and different conditions.

- **Maintainability:**

  Different versions of the product should be easy to maintain. For development it should be easy to add code to existing system, should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy. System be easy to maintain and correcting defects or making a change in the software.

- **Usability:**

  Different versions of the product should be easy to maintain. For development it should be easy to add code to existing system, should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy. System be easy to maintain and correcting defects or making a change in the software.

- **Portability:**

  This can be measured in terms of Costing issues related to porting, Technical issues related to porting, Behavioral issues related to porting.

- **Correctness:**

  Application should be correct in terms of its functionality, calculations used internally and the navigation should be correct. This means application should adhere to functional requirements.

- **Efficiency:**

  To Major system quality attribute. Measured in terms of time required to

complete any task given to the system. For example system should utilize processor capacity, disk space and memory efficiently. If system is using all the available resources then user will get degraded performance failing the system for efficiency. If system is not efficient then it can not be used in real time applications.

- **Integrity or Security:**

  Integrity comes with security. System integrity or security should be sufficient to prevent unauthorized access to system functions, preventing information loss, ensure that the software is protected from virus infection, and protecting the privacy of data entered into the system.

- **Testability:**

  System should be easy to test and find defects. If required should be easy to divide in different modules for testing.

- **Flexibility:**

  Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.

- **Reusability:**

  Software reuse is a good cost efficient and time saving development way. Different code libraries classes should be generic enough to use easily in different application modules. Dividing application into different modules so that modules can be reused across the application.

- **Interoperability:**

  Interoperability of one system to another should be easy for product to exchange data or services with other systems. Different system modules

should work on different operating system platforms, different databases and protocols conditions. Applying above quality attributes standards we can determine whether system meets the requirements of quality or not.

3.3 SYSTEM REQUIREMENTS

3.3.1 Dataset Requirement

The publicly available datasets are the Oxford-102 flowers dataset and the Caltech CUB-200 birds dataset. These two datasets are the ones which are usually used for research on text to image synthesis. Oxford-102 contains 8,192 images from 102 categories of flowers. The CUB-200 dataset includes 11,788 pictures of 200 types of birds. These datasets include only photos, but no descriptions.But there are publicly available captions collected by Reed et al for these dataset using Amazon Mechanical Turk. Each of the images has five descriptions. They are at least ten words in length, they do not describe the background, and they do not mention the species of the flower or bird.

3.3.2 Software Requirements

This system will work on any platform namely mac,ubuntu,windows etc. which has python installed on it and required packages like tkinter,tensorflow,keras,numpy,pandas etc..

3.3.3 Hardware Requirement

The program will not require any external hardware resource if the dataset size is small,but in case of large dataset a external GPU can be used to enhance the performance of the system.

3.4 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

Software Development Life Cycle:

A software development life cycle is a essentially a series of steps, or phases, that provide a model for the development and life cycle management of an application or piece of software .it is an well defined, structured sequence of stages in software engineering to develop the intended software product or module.

3.4.1 Iterative model

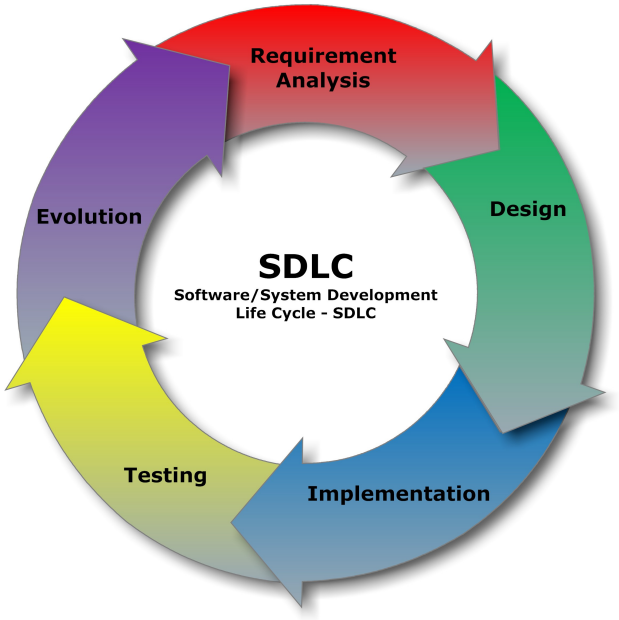

Figure 3.1: SDLC model

- **Communication :**

  It is an first step where the user initiates the request for desired software product. He contacts the service provider and tries to negotiate the terms. He submits his request to the service providing organization in writing.

- **Requirement Gathering :**

  This steps onwards the software development term works to carry on the project . the requirements contemplated and segregated into user requirements, system requirements and functional requirements. The requirements are collected using a number of practioners. After the requirement gathering developer make the documentation . Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

- **Designing The Product Architecture :**

  SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

- **Building or Developing the Product :**

  In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

- **Testing the Product :**

  This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality

standards defined in the SRS.

- **Deployment in the Market and Maintenance :**

  Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization.

3.5 SYSTEM IMPLEMENTATION PLAN

Our aim is to build a working model first which will work for at least one type of images.And if we get proper dataset and for other type of images then we will expand the system to work for those images as well.

Chapter 4

SYSTEM DESIGN

4.1 High level Design:

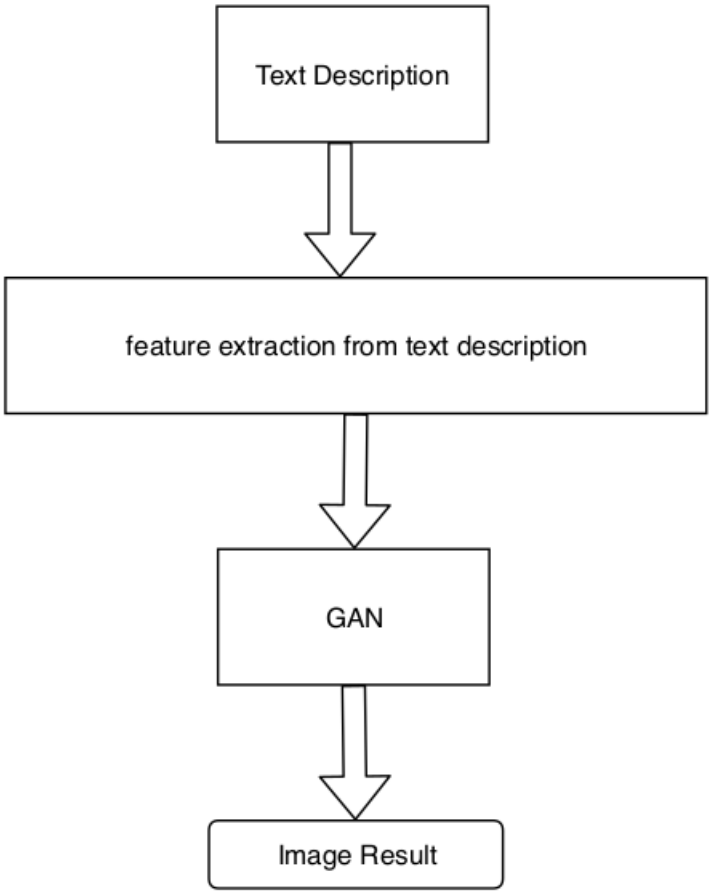

Figure 4.1: High level Diagram

4.2 Low level Design:

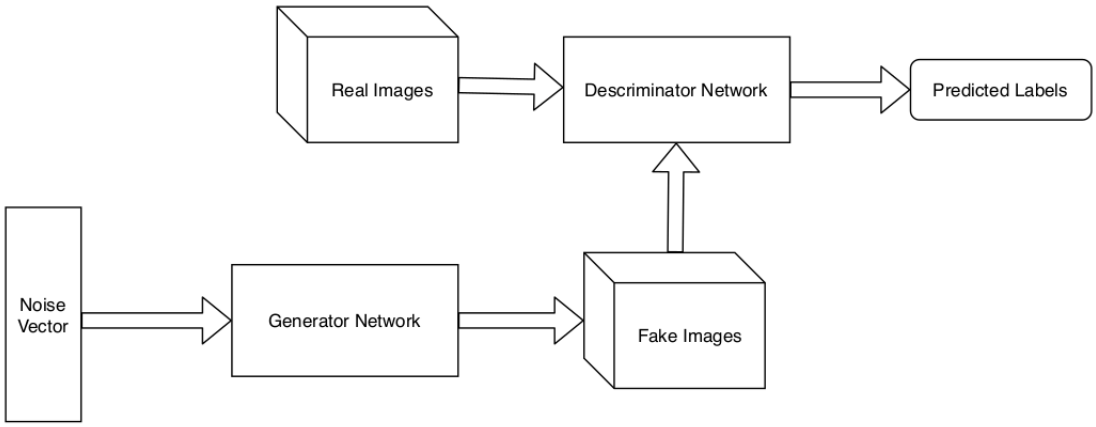

Figure 4.2: Low level Diagram

Use Case:

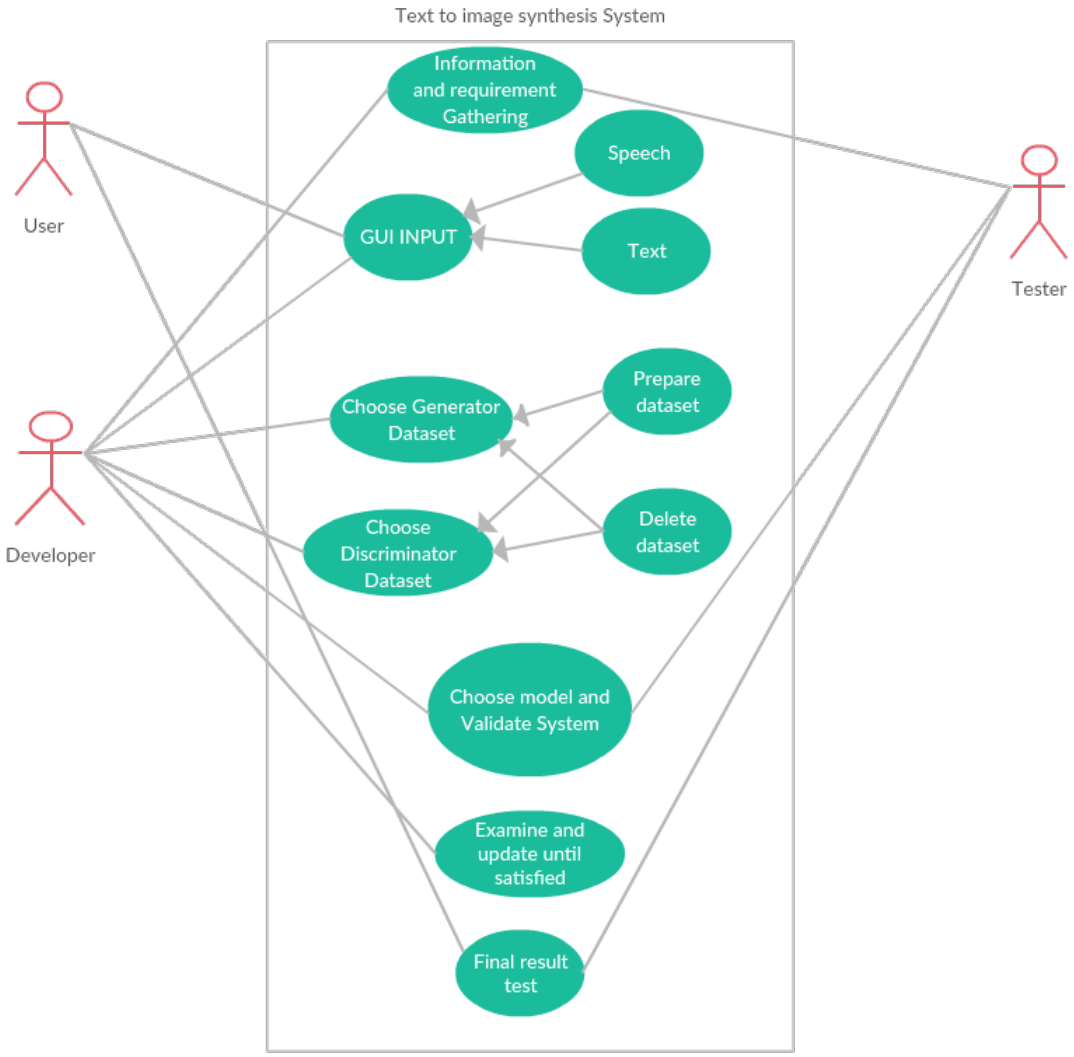

Figure 4.3: Use Case Diagram

4.3 Architecture:

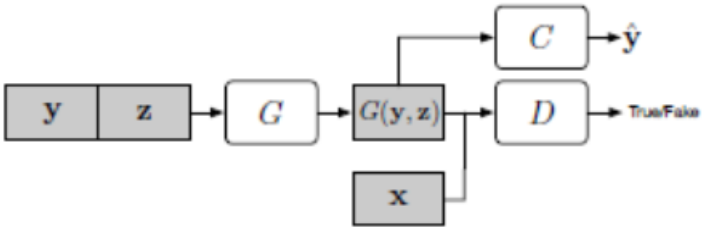

Figure 4.4: GAN architecture

Generative Adversarial Net (GAN) consists of two separate neural networks: a generator G that takes a random noise vector z, and outputs synthetic data G(z); a discriminator D that takes an input x or G(z) and output a probability D(x) or D(G(z)) to indicate whether it is synthetic or from the true data distribution, as shown in Figure 1. Both of the generator and discriminator can be arbitrary neural networks. The first GAN uses fully connected layer as its building block. Later, DCGAN proposes to use fully convolutional neural networks which achieves better performance, and since then convolution and transposed convolution layers have become the core components in many GAN models.

In the original GAN, we have no control of what to be generated, since the output is only dependent on random noise. However, we can add a conditional input c to the random noise z so that the generated image is defined by G(c; z). Typically, the conditional input vector c is concatenated with the noise vector z, and the resulting vector is put into the generator as it is in the original GAN. Besides, we can perform other data augmentation on c and z. The meaning of conditional input c is arbitrary, for example, it can be the class of image, attributes of object or an embedding of text descriptions of the image we want to generate

Generator Network:

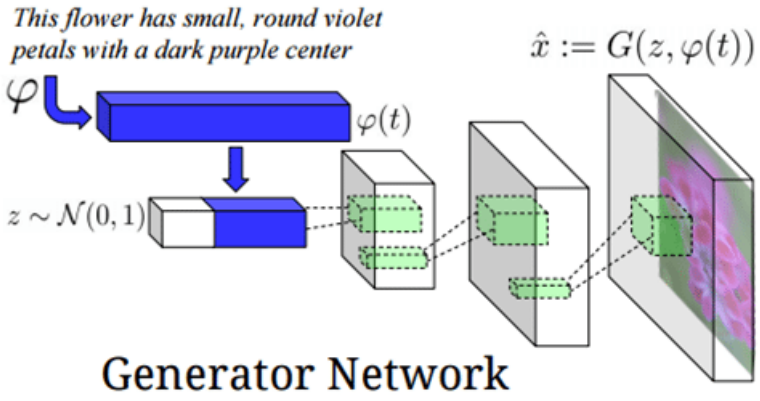

Figure 4.5: Generator network

The generator is an inverse convolutional network, in a sense: While a standard convolutional classifier takes an image and downsamples it to produce a probability, the generator takes a vector of random noise and upsamples it to an image. The first throws away data through downsampling techniques like maxpooling, and the second generates new data.

The generator takes in random numbers and returns an image. However, the generator initially produces garbage images, and the loss value is high. So, the back-propagation updates the generators weights to produce more realistic images as the training continues. This is how the generator get via training the GAN. This generated image is fed into the discriminator alongside a stream of images taken from the actual dataset. The generator is in a feedback loop with the discriminator. By considering above case, one way to think about generative algorithms is that they do the opposite. Instead of predicting a label given certain features, they attempt to predict features given a certain label.

The question a generative algorithm tries to answer is: Assuming this email is spam, how likely are these features? While discriminative models care about the relation between y and x, generative models care about how you get x. They allow you to

capture p(x—y), the probability of x given y, or the probability of features given a class. (That said, generative algorithms can also be used as classifiers. It just so happens that they can do more than categorize input data.)

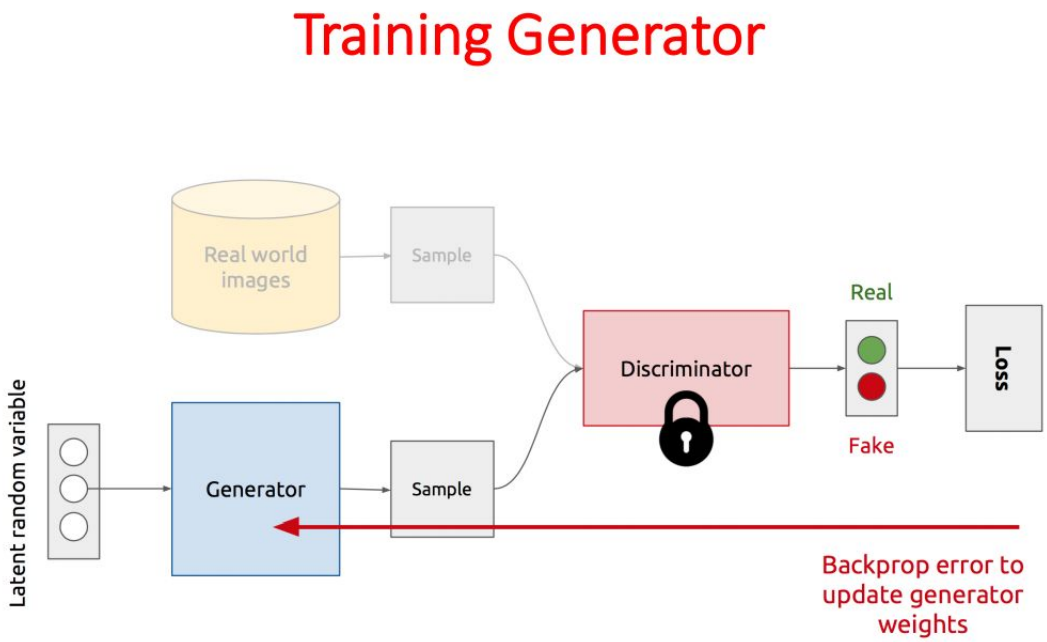

Figure 4.6: Generator training

Discriminator Network:

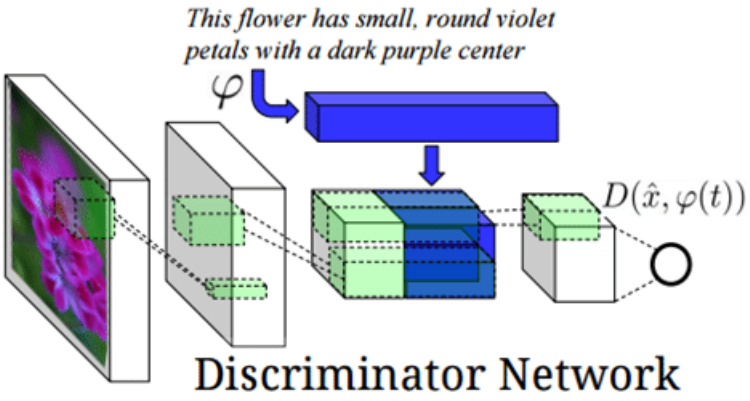

Figure 4.7: Discriminator network

The discriminator network is a standard convolutional network that can categorize the images fed to it, a binomial classifier labeling images as real or fake. The discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a prediction of authenticity and 0 representing fake. Lets not forget that there is need to train the discriminator as well so that it can do a good job as a classifier of real and fake images. To train the discriminator and the generator in turn in a loop as follows:

Step 1) Set the discriminator trainable

Step 2) Train the discriminator with the real dataset images and the images generated by the generator to classify the real and fake images.

The discriminator is in a feedback loop with the ground truth of the images.

Let's take example, given all the words in an email, a discriminative algorithm could predict whether the message is spam or not spam. spam is one of the labels, and the bag of words gathered from the email are the features that constitute the input data. When this problem is expressed mathematically, the label is called yand the features are called x. The formulation p(y—x) is used to mean the probability of y given x, which in this case would translate to the probability that an email is spam given the

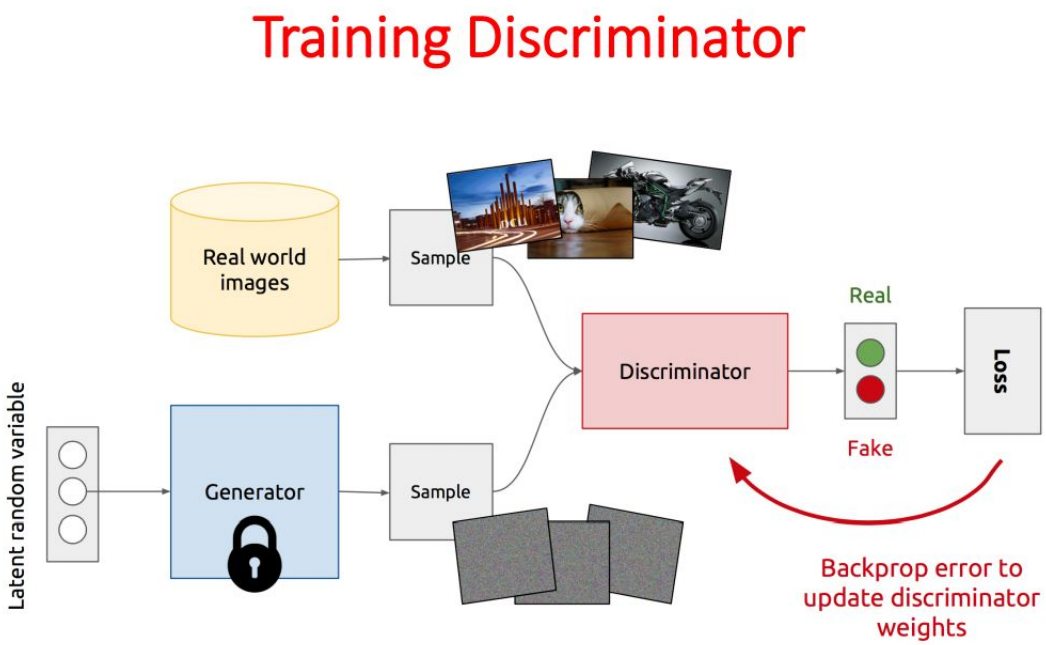

Figure 4.8: Discriminator training

words it contains.

So discriminative algorithms map features to labels. They are concerned solely with that correlation.

Chapter 5

Other Specifications

5.1 Advantages:

a) The model can synthesize many plausible visual interpretations of a given text caption.

b) A generalized model is created which can synthesize image with multiple object and variable backgrounds

c)GANs are a good method for training classifiers in a semi-supervised way.DCGAN model for the implemention has been used which is specialized for the image/video synthesis.

d)GANs generate samples faster than fully visible belief nets (NADE, PixelRNN, WaveNet, etc.) because there is no need to generate the different entries in the sample sequentially.

5.2 Limitations:

a) Images produced are of lower resolution.

b) Input to the system is restricted to the only one language i.e English.

c) Does not incorporate hierarchical structure into the image synthesis model in order to better handle complex multi-object scene.

5.3 Applications:

Create Anime characters:

Game development and animation production are expensive and hire many production artists for relatively routine tasks. GAN can auto-generate and colorize Anime characters.

Pose Guided Person Image Generation:

With an additional input of the pose, we can transform an image into different poses. For example, the top right image is the ground truth while the bottom right is the generated image.The design composes of a 2-stage image generator and a discriminator. The generator reconstruct an image using the meta-data (pose) and the original image. The discriminator uses the original image as part of the label input to a CGAN design.

CycleGAN:

Cross-domain transfer GANs will be likely the first batch of commercial applications. These GANs transform images from one domain (say real scenery) to another domain (Monet paintings or Van Gogh).

PixelDTGAN:

Suggesting merchandise based on celebrity pictures has been popular for fashion blogger and e-commerce. PixelDTGAN creates clothing images and styles from an image.

High-resolution image synthesis:

This is not image segmentation! It is the reverse, generating images from a semantic map. Collecting samples are very expensive. We have trying to supplement training dataset with generated data to lower development cost. It will be handy to generate

videos in training autonomous cars rather than see them cruising in your neighborhood.

Learn Joint Distribution:

It is expensive to create GANs with different combinations of facial characters P(blond, female, smiling, with glasses), P(brown, male, smiling, no glasses) etcThe curse of dimensionality makes the number of GANs to grow exponentially. Instead, we can learn individual data distribution and combine them to form different distributions. i.e. different attribute combinations.

Super resolution

Create super-resolution images from the lower resolution. This is one area where GAN shows very impressive result with immediate commercial possibility.Similar to many GAN designs, it composes of many layers of convolutional layer, batch normalization, advanced ReLU and skip connections.

Chapter 6

CONCLUSION

In this work we developed a simple and effective model for generating images based on detailed visual descriptions. We demonstrated that the model can synthesize many plausible visual interpretations of a given text caption. Our manifold interpolation regularizer substantially improved the text to image synthesis on CUB dataset.

Bibliography

[1] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran "Generative Adversarial Text to Image Synthesis" arXiv preprint arXiv:1605.05396v2 [cs.NE] 5 Jun 2016.

[2] Han Zhang1, Tao Xu2, Hongsheng Li3, Shaoting Zhang4, Xiaogang Wang3, Xiaolei Huang2, Dimitris Metaxas1"StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks" arXiv preprint arXiv:1612.03242v2 [cs.CV] 5 Aug 2017.

[3] He Huang, Phillip S. Yu and Changhu Wang."An Introduction to Image Synthesis with Generative Adversarial Nets" arXiv preprint arXiv:1803.04469v1 [cs.CV] 12 Mar 2018.

[4] David R. Cheriton School of Computer Science, University of Waterloo."Comparative Study on Generative Adversarial Networks" arXiv preprint arXiv:1801.04271v1 [cs.LG] 12 Jan 2018

[5] Alec Radford Luke Metz,Soumith Chintala."UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS" arXiv preprint arXiv:1511.06434v2 [cs.LG] 7 Jan 2016.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley,Sherjil Ozairy, Aaron Courville, Yoshua Bengio."Generative Adversarial Nets" arXiv preprint aarXiv:1406.2661v1 [stat.ML] 10 Jun 2014.

[7] Scott Reed1, Zeynep Akata2, Honglak Lee1 and Bernt Schiele."Learning Deep Representations of Fine-Grained Visual Descriptions"IEEE 2016 Conference DOI-10.1109/CVPR.2016.13 Date-12 December 2016

[8] Antonia Creswellx, Tom White, Vincent Dumoulinz, Kai Arulkumaranx, Biswa Senguptayx and Anil A Bharathx."Generative Adversarial Networks: An Overview" arXiv preprint arXiv:1710.07035v1 [cs.CV] 19 Oct 2017.