Chapter 1

INTRODUCTION

1.1 Overview

One of the most challenging issues in the realm of Computer Vision is incorporating top notch pictures from content portrayals. Almost certainly, this is intriguing and helpful, yet current AI frameworks are a long way from this objective. In recent years, powerful neural network architectures like GANs (Generative Adversarial Networks) have been found to generate good results. [1] Samples generated by existing text-to-image approaches can roughly correspond to the meaning of the given image descriptions, but they fail to contain necessary details and vivid object parts [2]. Through this project, we wished to explore architectures that could help us achieve our task of generating images from given text descriptions. Bad communication frequently prompts error or misconception of the circumstance. It frequently happens that the receiver confuses the message and expect a totally unique view about the context than the sender. This sort of miscommunication frequently prompts enormous issues. It has been scientifically discovered that the human's memory is generally excellent w.r.t graphical representations than to textual descriptions. It is simpler for us to recall pictures or recordings or any graphical introductions as opposed to recollecting textual descriptions.

Bad communication frequently prompts error or misconception of the circumstance. It frequently happens that the receiver confuses the message and expect a totally unique view about the context than the sender. This sort of miscommunication frequently prompts enormous issues. It has been scientifically discovered that the human's memory is generally excellent w.r.t graphical representations than to textual descriptions. It is simpler for us to recall pictures or recordings or any graphical introductions as opposed to recollecting textual descriptions.

Generating pictures from subtitles has been an intriguing field of research with regards to ongoing years. It has different applications in numerous fields. As of now, there are various models that convert any literary depiction into pictures. This project targets consolidating Voice-To-Text along with Text-To-Image transformation and give a Voice based interface to the clients with the goal that they can simply peruse out the inscriptions or any portrayal about a picture and the model produces the comparing pictures for them. The use of Speech Recognition model gives an easy to understand UI.

Visual graphical contents for the most part draw attention of viewers more than simple depictions. Rather than reading (or listening) and understanding an enormous section of depictions about any matter, seeing and getting data from a picture is considerably more quicker and less difficult. It makes the way toward transferring data substantially more productive and recoveries an immense measure of time. Generating photo-realistic images from text has tremendous applications, including photo-editing, computer-aided design, etc.

1.2 Motivation

Effective communication is one of the key parts of progress. This project discovers application in different fields, for example, training, interior designing, medicinal field and so forth. It's constantly more clear any substance by envisioning a graphical portrayal of it as opposed to only perusing or tuning in to its depiction. In interior designing applications, clients can simply depict how they anticipate that the inside plans should be in their own words and the model creates the relating pictures. Clients will actually have the option to picture the rooms before executing it and make changes as needs be. In instructive establishments, speakers can utilize the model to create pictures of the points that they are clarifying and make it simpler for the understudies to comprehend it better.

1.3 Example

Text description: This white and yellow flower has thin white petals and a round yellow stamen.

Generated Images:

Fig. 1.1 Example of image generated from the text description

Converting natural language text descriptions into images is an exceptional demonstration of Deep Learning. Text classification tasks such as sentiment analysis have been successful with Deep Recurrent Neural Networks that are able to learn discriminative vector representations from text. In another domain, Deep Convolutional GANs are able to synthesize images such as interiors of bedrooms from a random noise vector sampled from a normal distribution. The focus of Reed et al. [1,2] is to connect advances in Deep RNN text embeddings and image synthesis with DCGANs, inspired by the idea of Conditional-GANs.

Conditional-GANs work by inputting a one-hot class label vector as input to the generator and discriminator along with the randomly created noise vector. This results in higher training stability, more visually appealing results, as well as controllable generator outputs. The difference between traditional Conditional-GANs and the Text-toImage model lies in the way the input is conditioned. Instead of trying to create a sparse visual characteristic descriptor to condition GANs, the GANs are conditioned on a text embedding trained with a Deep Neural Network. A sparse visual attribute descriptor may represent "a red bird with a white beak" as:
[0 0 0 1 . . . 0 0 . . . 1 . . . 0 0 0 . . . 0 0 1 . . .0 0 0]

The ones in the vector would represent attribute questions such as, red (1/0)? white (1/0)? bird (1/0)? This description is difficult to collect and doesn't work well in practice.

Word embeddings have been the buzzword of natural language processing through the use of concepts such as Word2Vec. Word2Vec forms embeddings by learning to predict the context of a given word. Unfortunately, Word2Vec doesn't quite translate to text-to-image since the context of the word doesn't capture the visual properties as well as an embedding explicitly trained to do so does. Reed et al. [1, 2] present a novel symmetric structured joint embedding of pictures and captions to overcome this challenge.

In addition to constructing good text embeddings, translation of image from text is a highly multi-modal task. The term 'multi-modal' is an important one to become familiar with in Deep Learning research. This refers to the fact that there are multiple

different images of birds that correspond to the textual description “bird”. Another example in speech is that there exist multiple different accents, that would result in different sounds corresponding to the text “bird”. Multi-modal learning is also present in image captioning (image-to-text). However, this is greatly facilitated due to the sequential structure of text such that the model can predict the next word conditioned on the image as well as the previously predicted words. Multi-modal learning is traditionally very difficult, but is made much easier with the advancement of GANs (Generative Adversarial Networks), this framework creates an adaptive loss function which is well-suited for multi-modal tasks such as text-to-image.

1.4 Technology Stack

GANs

GANs, Generative Adversarial Networkss, are generative models that employ deep learning methods mainly convolutional neural networks.

Generative models are unsupervised learning tasks in Machine Learning that involve automatically learning patterns in input such that the model can be used to generate new instances convincingly, that could not have possibly been drawn from the original data.

GANs is a supervised learning problem that includes sub-models: the **generator model** and the **discriminator model.** The **generator model** is trained to generate new instances, while the **discriminator model** is trained to classify instances as either real or fake. The two models are trained in paralled in a zero-sum adversarial game, until the discriminator model is fooled more than half of the times, indicating that the generator is creating credible instances.

GANs have the ability to generate realistic instances across multiple domains, especially in image-to-image translation tasks like translation of photos and in generation of photorealistic images of objects, people and scenes that humans cannot themselves distinguish as fake or real.

GAN and Convolutional Neural Networks

GANs typically employ image data and use Convolutional Neural Networks, or CNNs, as the corresponding generator and discriminator models.

The reason behind this may be because the first description of the technique was in the field of computer vision and used CNNs and image data, and because of the remarkable progress that has been seen in recent years using CNNs more generally to achieve state-of-the-art results on a suite of computer vision tasks such as face recognition and object detection.

Modeling image data means that the latent space, the input to the generator, provides a compressed representation of the set of images or photographs used to train the model. It also means that the generator generates new photographs or images, providing an output that can be viewed easily and assessed by developers of the model.

It is this fact, the ability to visually assess the quality of the generated output, that has both led to the focus of computer vision applications with CNNs and on the massive leaps in the capability of GANs as compared to other generative models, deep learning based or otherwise.

Conditional GAN

A critical application of GAN is in their use for conditionally generating an output.

The generative model is trained to generate new examples from the input domain, where the input which is the random vector from the latent space, is provided with and conditioned by some additional input.

The additional input could be a class value, such as male or female in the generation of photographs of people, or a digit, in the case of generating images of handwritten digits.

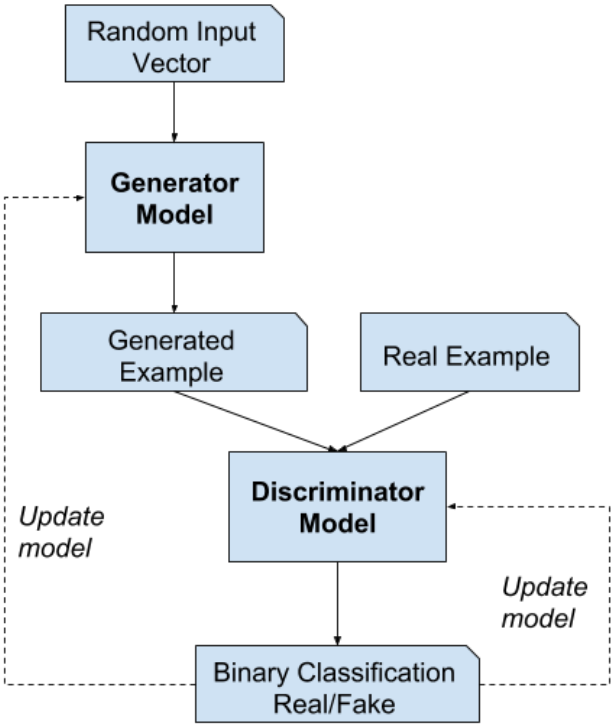

Fig 1.2: Conditional GANs

The discriminator is also conditioned, implication that it is provided both with an input image i.e. either real or fake and the additional input. In the case of a classification label type conditional input, the discriminator would then expect that the input would be of that class, in turn teaching the generator to generate examples of that class in order to fool the discriminator.

In the scenario of conditional GANs used for image-to-image translation, such as transforming day to night, the discriminator is provided examples of real and generated night time photos as well as conditioned real day-time photos as input. The generator is provided with a random noise vector from the latent space as well as conditioned real daytime photos as input.

Speech Recognition

Speech is a natural method for correspondence for individuals, however some of the time it doesn't work for example impaired individual. Over the most recent multi decade, here is a need to empower humans to speak with machines without playing out any content input. Speech recognition is a framework that utilized by the human to tune in, distinguish and get what does the client need by talking.

The Speech recognition innovation is the cutting edge that enables the machine to transform the voice signal into the proper content or order through the way toward distinguishing and understanding. Speech recognition is a cross-disciplinary and includes a wide range. It has an extremely close association with acoustics, phonetics, etymology, data hypothesis, design acknowledgment hypothesis and neurobiology disciplines. With the fast improvement of computer equipment and programming and data innovation, speech recognition innovation is step by step turning into a key innovation in the data handling technology.

Why use a voice driven interface??

Speed: Dictating text messages is much more quicker than typing, even for expert typers.

Hands-free: In some cases, such as when you are doing some other work or even when you're across the room from your device, make speaking rather than typing or tapping much more practical.

Intuitiveness: Everybody knows to speak. Hand a new interface to someone and make it ask that person some questions, and even the users who are less familiar with technology will be able to reply naturally.

Empathy**:** Humans find it difficult to understand tone via the written word alone. Texts mean different when it is said in different tones. Voice conveys a great deal of information.

Types of Speech:

- Isolated word:
    - Isolated word perceives accomplish generally require every utterance to have silence on the two sides of test windows. It takes in single words or single utterances one after another .This is having "Listening and Non Listening state"
- Connected Word:
    - The Connected word framework is like disengaged words,but enable the different utterances to be run together with least delay between them.

- Continuous speech:
    - Continuous speech recognizers enables client to talk nearly normally, while the PC decide the contents. Recognizer with Continuous speech capacities are some of the most hard to make since they use unique strategy to decide articulation limits. This innovation led to first vocabulary recognition systems which were utilized to get to databases
- Spontaneous speech:
    - Spontaneous speech at an essential level, it very well may be thought of as speech that is characteristic Sounding and not practiced. An ASR System with spontaneous speech capacity ought to have the option to deal with an assortment of common speech highlight, for example, words being run together. Spontaneous speech recognition frameworks are able to both perceive the verbally expressed material precisely and comprehend the significance of the verbally expressed material.

CHAPTER 2

LITERATURE SURVEY

The CHiME challenge[8] arrangement plans to progress far field speech recognition innovation by advancing exploration at the interface of sign preparing and programmed speech recognition. It presents the structure and results of the third CHiME Challenge, which targets the presentation of programmed speech recognition in a genuine world, monetarily inspired situation: an individual verbal blistering a tablet gadget that has been fitted with a six-channel amplifier cluster. The paper portrays the information assortment, the errand definition and the gauge frameworks for information recreation, improvement and acknowledgment. The paper at that point displays a review of the 26 frameworks that were submitted to the test concentrating on the techniques that proved to be most fruitful comparative with the MVDR cluster handling and DNN acoustic demonstrating reference framework. Challenge discoveries identified with the job of recreated information in framework preparing and assessment are talked about.

This Speech-to-Text transformation framework in [9] is executed by utilizing the MFCC for feature extraction and HMM as the recognizers. In discourse database, fifty sound documents are recorded also, these are broken down to get include vectors. These highlights are at first displaying in the HMM. From that point forward, the test verbally expressed word is tended to by forward calculation of HMM. From the re-enactment results, it very well may be plainly observed that the normal acknowledgment pace of 87.6% accomplished by the quantity of states (N=5) is preferable precision over some other states. In any case, if the number of states is excessively enormous, there are no enough perceptions per state to prepare the model.

Moulines, E., in his paper "Text-to-speech algorithms based on FFT synthesis" [10] present French speech-to-text framework dependent on diaphone link. FFT union procedures are prepared to do delivering high calibre prosodic alterations of natural speech. A few distinct methodologies are detailed to decrease the twists because of diaphone concatenation.

Sultana, S.; Akhand, M. A H; Das, P.K.; Hafizur Rahman, M.M. investigate Speech-to-Text (STT) change utilizing SAPI for Bangla language. In spite of the fact that accomplished execution is promising for STT related examinations, they recognized a few components to recoup the presentation and might give better precision and guarantee that the topic of this study will likewise be useful for different dialects for Speech- to-Text change and comparative assignments.

Pre-processing: The analog speech signal is changed into digital signals for later handling. This digital signal is moved to the main request channels to spectrally straighten the signals. This aides in expanding the signal's energy at a higher frequency.

The speaker recognition system may be viewed as working in a four stages- Analysis, Feature Extraction, Modelling, Testing. The framework gains discourse at run time through a receiver and procedures the examined discourse to recognize the expressed content. The perceived content can be put away in a document.

An Offline Voice to Text translation framework for Healthcare Organization which can be utilized by counsellors and NGO's to record the discussion during reviews and convert it into content and save it in files is developed using the CMUSphinx toolkit [11] for speech recognition.

According to [12], in the event that the importance of the speaker with speech recognition framework is thought of it as, can be isolated into three classifications: (1) Specific speech recognition framework, which considers just the specific individual voice; (2) Speaker-free speech recognition framework. The recognition of discourse is irrelevant to man. It for the most part needs countless diverse speech databases for the identification framework to learn; (3) Recognition framework for many people. It for the most part can distinguish a gathering of human talks, or be a speech recognition framework for a specific gathering of discourses, the framework just requires to prepare the gathering of voices which should be recognized. ASR frameworks work in two stages. Initial, a training stage, during which the framework learns the reference designs speaking to the distinctive discourse sounds (for example phrases, words, phones) that establish the jargon of the application. Each reference is

found out from spoken models and put away either as formats acquired by some averaging strategy or models that portray the measurable properties of example. Second, a recognising stage, during which an obscure info design, is recognized by considering the arrangement of references.

Chapter 3

AIM

3.1 Aim

A few specialists propose that pictures are bound to be recollected than words, in light of the fact that our minds dually encode pictures, yet encode words just once. It implies that when people see a picture it is put away in their memory "as an image", yet in addition "as a word". Be that as it may, when people hear (or see) a word, it is put away in their memory just "as a word"**.** The brain can process pictures more rapidly than verbal or composed data. Researchers accept that the brain can process pictures approximately 60,000 times more quickly than it forms a comparable measure of composed data. Pictures have an altogether constructive outcome on different levels: it catches eye, upgrades appreciation and improves review. Along these lines, use visuals – it is wonderful to the eyes, brisk for our minds to scan or decode, and simple to recollect.

This project aims to build a voice-to-image system that takes in voice (description of image) as input and generates corresponding images. This system could be used in medical imaging. This model finds its application in education field too. For instance, teachers could use this system to teach the students, since students are likely to learn better with images than words. The classes become more interactive with teachers clearing out doubts there and then, dynamically. This model can also be used in the field of interior designing where clients get an imaginary picture of the rooms before constructing it.

3.2 Problem Statement

Vision is one of the most essential ways in which humans communicate, interact, experience, and learn about the world around them. AI systems that can generate images and video for human users have applications ranging from education and entertainment to that of the creative arts. Such systems also have the potential to serve as accessibility tools for the physically impaired. A system that can follow speech- or text-based instructions and then perform a corresponding image generation task could improve this accessibility substantially. These benefits can easily extend to other domains of image generation such as gaming, animation, creating visual teaching material.

3.3 Objectives

- Build a Speech recognition system that can generate textual data
- GAN model that can translate textual captions into the corresponding image to a high degree of accuracy
- Integrate the two systems to form a single seamless Voice-to-Image synthesis model

Chapter 4

METHODOLOGY

4.1 Image Synthesis

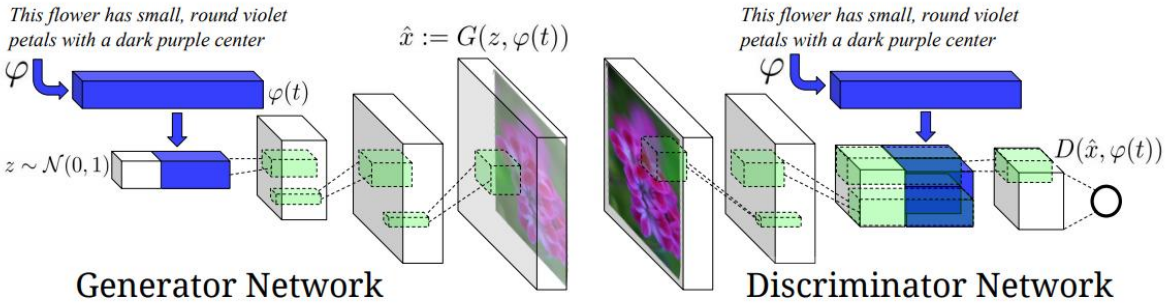

Fig. 4.1 Text conditional convolutional GAN Architecture

The above image depicts the architecture Reed et al. used to train their text-to-image GAN model. The most noteworthy takeaway from this diagram is the visualization of how the text embedding fits into the sequential processing of the model. In the Generator network, the text embedding is filtered trough a fully connected layer and concatenated with the random noise vector z. In the following scenario, the text embedding is converted from a 1024x1 vector to 128x1 and concatenated with the 100x1 random noise vector z. On the side of the discriminator network, the text-embedding is also compressed through a fully connected layer into a 128x1 vector and then reshaped into a 4x4 matrix and depth-wise concatenated with the image representation. This image representation is derived after the input image has been convolved over multiple times, reduce the spatial resolution and extracting information. The following strategy used for the embeddings for the discriminator is different from the conditional-GAN model in which the embedding is concatenated into the original image matrix and then convolved over.

In the architecture diagram we visualize how the DCGAN up samples vectors or low-resolution images to produce high-resolution images. You can see each de-convolutional layer increases the spatial resolution of the image. Additionally, the depth

of the feature maps decreases per layer. Lastly, you can see how the convolutional layers in the discriminator network decreases the spatial resolution and increase the depth of the feature maps as it processes the image.

In the following training process, it is difficult to separate loss based on the generated image not looking realistic or loss based on the generated image not matching the text description. The authors of the paper describe the training dynamics being that initially the discriminator does not pay any attention to the text embedding, since the images created by the generator do not look real at all. Once the Generator generate images that pass the real vs. fake criteria, then we decide factoring in the text embedding.

The authors smooth out the training dynamics of this by adding pairs of real images with incorrect text descriptions which are labelled as 'fake'. The discriminator's sole motive is the binary task of real vs. fake and is not separately considering the image apart from the text. This is in contrast to an approach such as AC-GAN with one-hot encoded class labels. The AC-GAN discriminator outputs real vs. fake and uses an auxiliary classifier sharing the intermediate features to classify the class label of the image.

Constructing a Text Embedding for Visual Attributes

The most interesting component of this paper is how a unique text embedding is constructed that contains visual attributes of the image to be represented as vectors. This vector is constructed through the following process:

$$\frac{1}{N}\sum_{n=1}^{N}\Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n)) \qquad (2)$$

where $\{(v_n, t_n, y_n) : n = 1, ..., N\}$ is the training data set, Δ is the 0-1 loss, v_n are the images, t_n are the corresponding text descriptions, and y_n are the class labels. Classifiers f_v and f_t are parametrized as follows:

$$f_v(v) = \arg\max_{y \in \mathcal{Y}} \mathbb{E}_{t \sim \mathcal{T}(y)}[\phi(v)^T \varphi(t))] \qquad (3)$$

$$f_t(t) = \arg\max_{y \in \mathcal{Y}} \mathbb{E}_{v \sim \mathcal{V}(y)}[\phi(v)^T \varphi(t))] \qquad (4)$$

The loss function noted as equation (2) represents the overall objective of a text classifier that is optimizing the gated loss between two loss functions. These loss functions are shown in equations 3 and 4. The paper describes the intuition for this process as "A text encoding should have a higher compatibility score with images of the corresponding class compared to any other class and vice-versa". The two terms represent an image encoder and a text encoder respectively. The image encoder is derived from the GoogLeNet image classification model. This classifier reduces the dimensionality of images upto the point where it is compressed to a 1024x1 vector. The objective function thus aims to minimize the distance between the text representation from a character-level CNN or LSTM and the image representation from GoogLeNet. Essentially, the vector encoding for the image classification is used to guide the text encodings based on similarity to similar images.

The details of this are expanded on in the following paper, "Learning Deep Representations of Fine-Grained Visual Descriptions" also from Reed et al.

Manifold interpolation

One of the interesting characteristics of Generative Adversarial Networks is that the latent vector z can be used to interpolate new instances, commonly referred to as "latent space addition". An example would be to do "man with glasses" — "man without glasses" + "woman without glasses" and achieve a woman with glasses. In this paper, the authors aims to interpolate between the text embeddings. This is done with the following equation:

$$\mathbb{E}_{t_1,t_2\sim p_{data}}[\log(1 - D(G(z, \beta t_1 + (1-\beta)t_2)))] \quad (5)$$

where z is drawn from the noise distribution and β interpolates between text embeddings t_1 and t_2. In practice we found that fixing $\beta = 0.5$ works well.

The discriminator has been trained to predict if image and text pairs differ or not. Therefore, images from interpolated text embeddings can fill in the gaps in the data manifold that were present during training. Using this as a regularization method for the

training data space is paramount for the successful result of the model presented in this paper. This is a form of data amplification since the interpolated text embeddings can enlarge the dataset used for training the text-to-image GAN.

4.1.1 StackGan[3]

The objective was to generate high resolution images that contain photo-realistic details. The authors proposed an architecture where the process of generating images from text is decomposed into two stages as shown in Fig. 4.2. The two stages are as follows:

Stage-I GAN: The primitive shape and basic colors of the object conditioned on the given text description and the background layout from a random noise vector are drawn, yielding a low-resolution image.

Stage-II GAN: The defects in the low-resolution image from Stage-I are corrected and fine details of the object are given a finishing touch by reading the text description again, producing a high-resolution photo-realistic image.

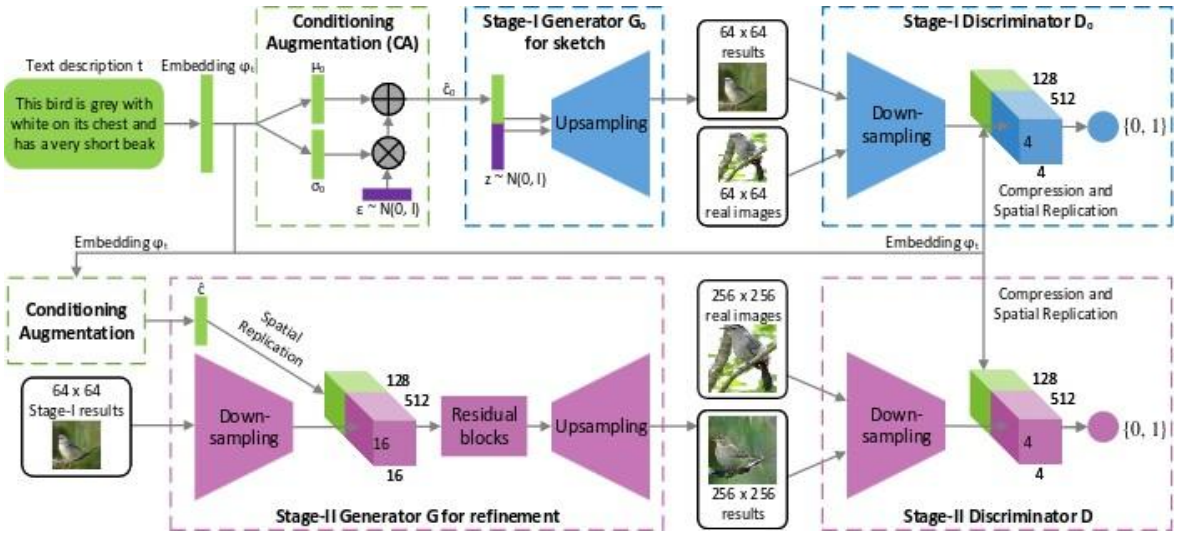

Fig. 4.2 Network Architecture of StackGAN

4.1.2 StackGAN++ [5]

StackGAN++ is an extended version of StackGAN discussed prior. It is an advanced multi-stage generative adversarial network architecture consisting of multiple generators and multiple discriminators arranged in a tree-like structure. The architecture generates images at multiple scales for the same text input. Experiment performed have demonstrated that this new proposed architecture significantly outperforms the other state-of-the-art methods in generating photo-realistic images. Fig. 4.2 shows the architecture.

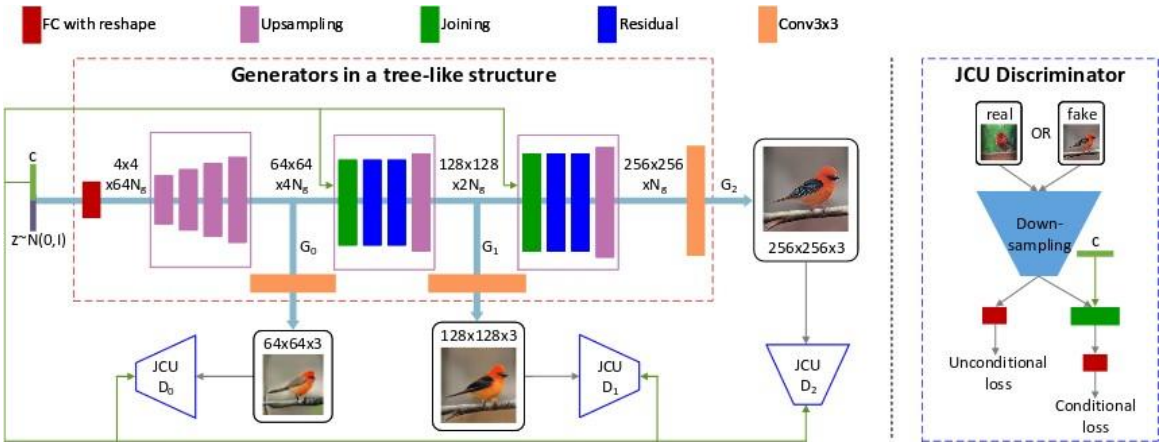

Fig. 4.3 Network Architecture of StackGAN++

4.1.3 AttnGAN[7]

Part 1: Multi-stage Image Refinement (AttnGAN)

The **Attentional Generative Adversarial Network** (called AttnGAN) starts with a low-resolution image, and then further improves it over compounding steps to generate the final image.

The first stage begins as shown below:

Initial step

Like most other Text-to-Image convertors, AttnGAN begins by generating an image from random noise and a summation of the caption's word-embeddings:

$h(0) = F(0)(z, E)$

Here, *z* represents the noise-vector, and *E* represents the sum of individual word-vectors. The 'hidden context' is denoted as h(0) — essentially, AttnGAN representation of what the image should look like. Based on the *h(0)*, we generate *x(0)* — the first image — using a GAN:

$x(0) = G(0)(h(0))$

We also have the Discriminator *D(0)*, that is corresponding to the Generator *G(0)*.

An example of *x(0)* from the paper[7]:

Caption: "*This is a bird that has a green crown black primaries and a white belly*"

Further epochs

One of the issues with generating an image from a combined 'sentence' vector *E*, is that we lose a lot of the fine-grained details hidden in the individual tokens.

For example, consider the example as shown above: When you combine (*green+crown+white+belly*) into a 'bag-of-words', you are much less likely to understand the actual colors of the crown & belly — hence resulting in the hazy coloring in the generated image.

To remedy this, AttnGAN uses a combination of Attention and GAN at every sequential stage, to *iteratively add details* to the image:

$h(i) = F(i)(h(i-1), Attn([e], h(i-1)))$

$x(i) = G(i)(h(i))$

7[h(1), h(2), ... follow the template above.

Compare these to the initial equations:

- *z* is replaced by the previous context *h(i-1)*.
- [*e*] denotes the set of all word-embeddings in the sentence. Using Attention based on *h(i-1)*, we compute a weighted average of [*e*] (*Attn([e], h(i-1))*) **to highlight words that require more detail**.
- Based on the weighted vector, *F(i)* alters *h(i-1)* to yield *h(i)*.
- Last step, a GAN is then used to produce *x(i)* from *h(i)*.

Continuing with the previous example:

The top attended tokens for h(1) stage*:* bird, this, has, belly, white

The top attended tokens for h(2) stage*:* black, green, white, this, bird

Fig 4.4: Generated image of a bird

Corresponding images (x(1) & x(2))

Consider the words for *h(2)*. You can view *x(2) as* being a more colorful generated version of *x(1)*.

The results are not always very accurate, but its a step in the right direction for optimizing the correct objectives. This brings us to the next part of the sequence.

Part 2: Multi-modal loss

The high-level diagram of the system, as given in the paper[7] is given as shown below:

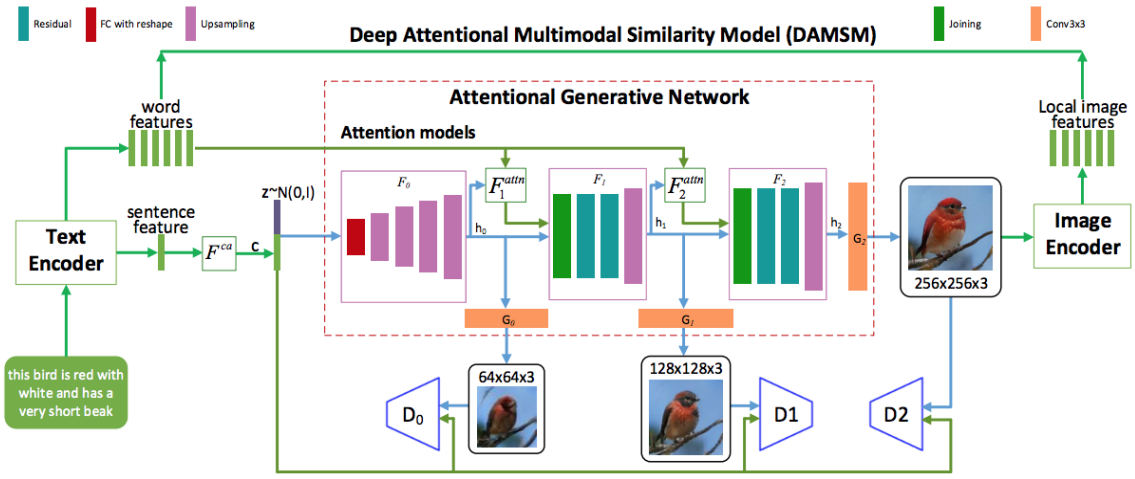

Fig 4.5: High level diagram of DAMSM

Essential parts that have to be covered, are mentioned below:

The Discriminators

Looking at the equations for *h* & *x*, it is natural to wonder why we need the *x*'s a t all, except at the last step. For instance, *x(0)* does not appear in the equations for *h(1)* and *x(1)*

The reason being training. In the learning phase, the *D*'s are trained with scaled-down versions of real image-caption examples (from a dataset as COCO). This makes the *G*'s better at generating *x*'s from the*h*'s. By back-propagation, this makes the *F* functions better at generating the hidden contexts and thereby ensuring that each stage sequentially adds something of meaning to the image.

The Deep Attentional Multimodal Similarity Model (DAMSM)

Post the concept of multi-stage image refinement, the other key feature of the framework is the **Deep Attentional Multimodal Similarity Model.**

While the individual discriminators do enhance the system, we do not yet have an objective that checks if every single word in the caption is appropriately represented in the given image, as the discriminators are trained on the overall caption *E* & the scaled-down image pairs.

To encode this task effectively, we first train — **the DAMSM**. DAMSM takes an input image and the set [*e*], and provides feedback on how well the two coincide with each other. It does this as follows:

- Using a Convolutional Neural Network, the image is converted into a set of feature maps. Each feature map signifies particular sub-regions within the image.
- The dimensionality of the feature maps is made equal to that of the wor d embeddings, so that they can be treated as comparable entities.
- Based on each token in the caption, Attention is applied over the feature maps, to compute a weighted average of them. This attention-vector essentially represents the image representation of the token.
- DAMSM is trained to minimize the difference between the above attention-vector i.e. visual portrayal of the word & the word embedding i.e. the textual meaning of the word. Essentialy we are trying to make the ‘red’ part of the image as ‘red’ as feasible.

The reason DAMSM is called ‘**multimodal**’, is because it defines an objective that combines two different modes of understanding — visual & textual.

Once DAMSM has been extensively trained on a dataset, it can then be used in union with the step-wise discriminators, to provide a rich target for AttnGAN to optimize.

4.2 Voice Recognition

4.2.1 Pytorch-kaldi

The PyTorch-Kaldi aims to make the improvement of automatic speech recognition framework easier, increasingly adaptable and enabling clients to effortlessly module their tweaked acoustic models. The objective is to combine and bridge the gap between the existing Pytorch and Kaldi toolkits. It bolsters numerous component and mark streams just as mixes of neural systems, empowering the utilization of complex neural structures. It contains various powerful features for development of modern speech recognition systems. User defined acoustic models can be easily plugged in and also they can make use of number of pre-implemented neural networks that can be configured to the required specifications using configuration files. Multiple feature and label streams and combination of neural networks is possible.

Labels: Training of acoustic models make use of labels. Context independent targets can be used for monophone regularization.

Feature extraction is done using the Kaldi libraries. Kaldi libraries are basically implemented in C++.Speech Recognition Features are extracted using these libraries and stored in binary archives. These binary archives are then imported in python environment using the kaldi-io libraries.

The dataset is divided into chunks that contains random labels and features. These chunks are bought into CPU or GPU, then processed using neural training algorithm. The features are fed in as input to the acoustic models which outputs the posterior probabilities. The obtained posterior probabilities are then normalized and fed to the kaldi's HMM-based decoder. The decoder makes use of acoustic scores with the language probabilities to retrieve the corresponding transcripts. TIMIT dataset used in training the pytorch-kaldi voice to text model.

The primary content to run tests is run_exp.sh. The main parameter that it takes in input is the setup record. Each training epoch is isolated into numerous chunks. The pytorch code run_nn_single_ep.py performs training over single chunk and gives in yield a model document in .pkl design and a .data record.

After each epoch, the exhibition on the dev-set is observed. On the off chance that the relative execution improvement is underneath a given edge, the learning rate is diminished by a splitting element. The training cycle is iterated for the predetermined number of epochs. When training is done, a forward advance is carried on for producing the arrangement of back probabilities that will be handled by the kaldi decoder.

Subsequent to interpreting, the last translations and scores are accessible in the yield organizer. On the off chance that, for reasons unknown, the training technique is interfered with the procedure can be continued beginning from the last handled piece.

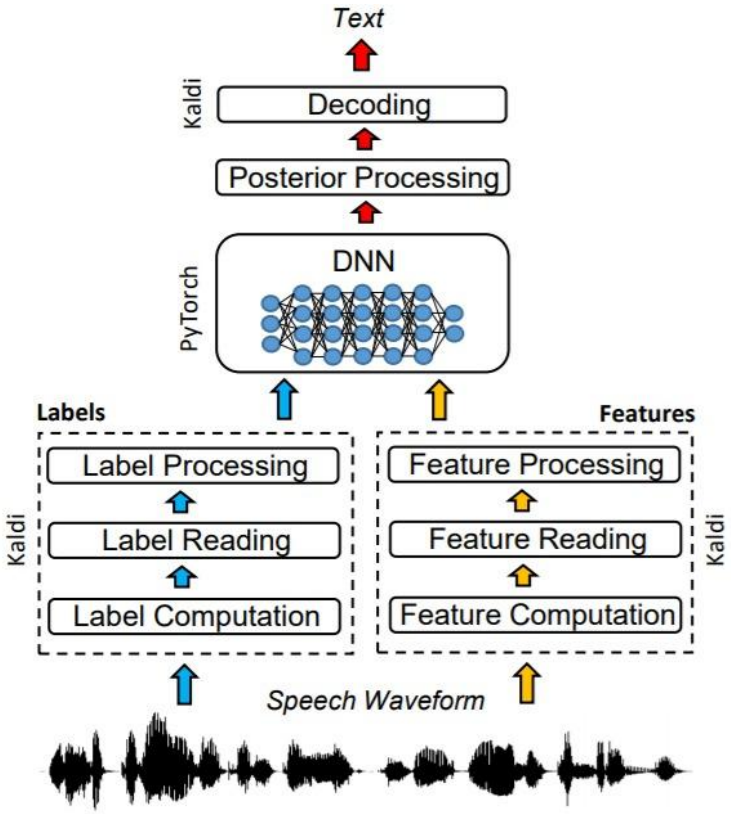

Fig 4.6: Pytorch-kaldi system architecture

Pytorch

Initially, PyTorch was created by Hugh Perkins as a Python wrapper for the LusJIT dependent on Torch structure. There are two PyTorch variations. PyTorch updates and actualizes Torch in Python while having a similar center C libraries for the backend code. PyTorch engineers tuned this back-end code to run Python proficiently. They likewise kept the GPU based equipment quickening just as the extensibility includes that made Lua-based Torch.

The significant highlights of PyTorch are referenced beneath :

- Simple Interface − PyTorch offers simple to utilize API; subsequently it is viewed as extremely easy to work and runs on Python. The code execution in this structure is very simple.
- Python utilization − This library is viewed as Pythonic which easily coordinates with the Python data science stack. In this manner, it can use every one of the administrations and functionalities offered by the Python environment.
- Computational diagrams − PyTorch gives a phenomenal stage which offers dynamic computational charts. Consequently a client can transform them during runtime. This is exceptionally helpful when a designer has no clue about how a lot of memory is required for making a neural system model.
- PyTorch is firmly identified with the lua-based Torch system which is effectively utilized in Facebook.
- PyTorch is generally new contrasted with other focused advances.
- PyTorch incorporates everything in objective and dynamic way.
- Calculation chart in PyTorch is characterized during runtime.
- PyTorch incorporates arrangement highlighted for portable and installed structures.

Advantages of using pytorch :

- It is easier to troubleshoot and comprehend the code.
- It incorporates numerous layers as Torch.
- It incorporates a great deal of loss functions.
- It very well may be considered as NumPy extension to GPUs.
- It permits building networks whose structure is subject to computation itself.

Kaldi

Kaldi intends to give programming that is adaptable and extensible. It bolsters straight changes, MMI, supported MMI and MCE discriminative preparing, include space discriminative preparing, and profound neural systems. It is likewise connected with enormous jargon decoders, for example, HDecode and Jullius. The objective of Kaldi is to have present day and adaptable code that is anything but difficult to comprehend, change and expand.

Kaldi has been consolidated as a component of the CHiME Speech Separation and Recognition Challenge more than a few progressive occasions. The product was at first created as a feature of a 2009 workshop at Johns Hopkins University.

Generic qualities of the Kaldi toolkit :

- It underlines algorithms that are generic and all recipes that are universal.
    - "Generic algorithms" refers to direct changes, as opposed to those that are explicit to discourse here and there. Be that as it may, it doesn't mean to be too closed minded about this, if increasingly explicit algorithms are valuable.
    - It uses recipes that can be run on any dataset, instead of those that must be tweaked.
- It inclines towards provably right algorithms
    - The recipes have been structured so that on a fundamental level they ought to never flop in a cataclysmic manner. There has been a push to stay away from plans and calculations that might come up to failure, regardless of whether they don't flop in the "ordinary case" .
- The code is straightforward.
    - Despite the fact that the Kaldi toolkit in general may get extremely enormous, it goes to every individual piece of it to be justifiable without a lot of exertion.

4.2.2 Deep Speech Model

This is a state-of-the-art speech recognition system. It is developed using end-to-end deep learning. The design is essentially less complex than conventional speech systems, which depend on relentlessly engineered processing pipelines. These conventional frameworks additionally will in general perform inadequately when utilized in noisy environments. In contrast, this system needn't bother with hand-planned parts to display background noise, speaker variety or resonation, yet rather legitimately learns a capacity that is robust to such impacts. It doesn't bother with a phoneme word reference, nor even the idea of a "phoneme." Key to this methodology is a well-upgraded RNN training system that uses different GPUs, just as a lot of novel information union strategies that enable us to productively acquire a lot of fluctuated information for training. This system, called Deep Speech, outflanks recently distributed outcomes on the generally considered Switchboard Hub5'00, accomplishing 16.0% blunder on the full test set. Deep Speech additionally handles noisy environments better than generally utilized, state-of-the-art commercial speech systems.

Conventional speech systems utilize numerous vigorously built processing stages, including particular input features, acoustic models, and Hidden Markov Models (HMMs). To improve these pipelines, area specialists must contribute a lot of exertion tuning their features and models. The presentation of deep learning algorithms has improved speech system execution, as a rule by improving acoustic models. While this improvement has been huge, deep learning still assumes just a constrained role in traditional speech pipelines. Subsequently, to improve execution on an undertaking, for example, recognizing speech in a noisy environment, one should difficultly design the remainder of the system for power. Conversely, this system applies deep learning end-to-end utilizing recurrent neural networks. This exploits the limit given by deep learning systems to gain from enormous datasets to improve the overall execution. This model is trained end to end to deliver interpretations and along these lines, with adequate information and processing power, can learn robustness to commotion or speaker minor departure from its own.

Tapping the advantages of end to end deep learning, be that as it may, represents a few difficulties: (i) we should discover inventive approaches to fabricate huge, labelled training sets and (ii) we should have the option to train networks that are sufficiently

huge to successfully use the entirety of this information. One test for taking care of labelled data in speech systems is finding the arrangement of content transcripts with input speech. This issue has been tended to by Graves, Ferna'ndez, Gomez and Schmidhuber, in this way empowering neural networks to effectively expend unaligned, interpreted audio during training. In the interim, fast training of enormous neural systems has been handled by Coates et al., showing the speed focal points of multi-GPU calculation. We plan to use these bits of knowledge to satisfy the vision of a generic learning system, in light of huge speech datasets and versatile RNN preparing, that can outperform more complicated conventional techniques.

This RNN model explicitly delineates to GPUs and utilizes a novel model standard partition plan to improve parallelization. Also, we propose a procedure for collecting enormous amounts of labelled speech data displaying the twists that the system ought to figure out how to deal with. Utilizing a blend of gathered and orchestrated information, this system learns strength to practical noise and speaker variation. Taken together, these thoughts get the job done to assemble an end to end speech system that is less complex than customary pipelines yet additionally performs better on troublesome speech tasks. Deep Speech accomplishes an error rate of 16.0% on the full Switchboard Hub5'00 test set—the best distributed outcome. Further, on another speech recognition dataset, this system accomplishes a word error rate of 19.1% where the best business systems accomplish 30.5% error.

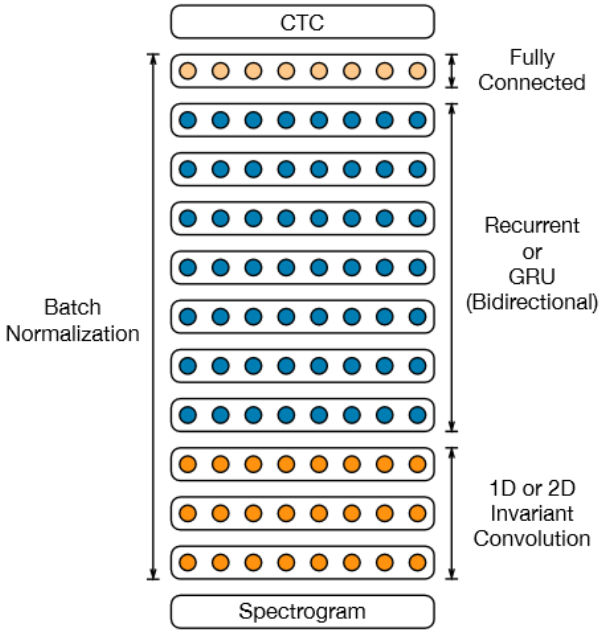

Fig 4.7: Architecture of Deep Speech2

A straightforward multi-layer model with a single recurrent layer can't exploit large number of long hours of labeled speech. So as to gain from datasets this huge, we increment the model limit by means of depth. We investigate structures with up to 11 layers including numerous bidirectional recurrent layers and convolution layers. These models have about 8 times the measure of computation per information model as the models in DeepSpeech1 making quick streamlining and computation basic. So as to enhance these models effectively, we use Batch Normalization for RNNs and a novel improvement educational plan we call SortaGrad. We additionally misuse long walks between RNN inputs to diminish computation per model by a factor of 3. This is useful for both training and evaluation, however requires some modifications so as to function admirably with CTC. At last, however a considerable lot of our exploration results utilize bidirectional recurrent layers, we find that excellent models exist utilizing just unidirectional recurrent layers an element that makes such models a lot simpler to deploy. Taken together these highlights enable us to tractably streamline deep RNNs and improve execution by over 40% in both English and Mandarin error rates over the smaller standard models.

The complete RNN model is delineated in Figure 4.7. It consists of a solitary recurrent layer (which is the hardest to parallelize) and it doesn’t utilize Long-Short-Term-Memory (LSTM) circuits. One inconvenience of LSTM cells is that they require registering and putting away numerous gating neuron reactions at each progression. Since the forward and in reverse repeats are successive, this little extra cost can turn into a computational bottleneck. By utilizing a homogeneous model we have made the calculation of the recurrent activation as effective as could be allowed: figuring the ReLu outputs includes just a couple profoundly upgraded BLAS tasks on the GPU and a solitary point-wise nonlinearity.

4.2.3 Google Speech API

Consideration based encoder-decoder designs, for example, Listen, Attend, and Spell (LAS), subsume the acoustic, elocution and language model segments of a customary automatic speech recognition (ASR) framework into a solitary neural system. In the past work, they have indicated that such designs are equivalent to state-of-the-art ASR frameworks on transcription assignments, yet it was not clear if such structures would be practical for all the more testing undertakings, for example, voice search. In this work, they investigate an assortment of auxiliary and advancement enhancements to our LAS model which fundamentally improve execution. On the basic side, this shows that word piece models can be utilized rather than graphemes. They present a multi-head consideration engineering, which offers enhancements over the generally utilized single-head consideration. On the streamlining side, we investigate strategies, for example, synchronous training, booked sampling, label smoothing, and least word error rate enhancement, which are altogether appeared to improve precision. The outcomes come with a unidirectional LSTM encoder for gushing acknowledgment. On a 12,500 hour voice search task, it was found that the proposed changes improve the WER of the LAS framework from 9.2% to 5.6%, while the best ordinary framework accomplish 6.7% WER. They tested the two models on a transcription dataset, and this model gave 4.1% WER while the traditional framework gave 5% WER.

System	Clean (94)	Noisy (82)	Combined (176)
Apple Dictation	14.24	43.76	26.73
Bing Speech	11.73	36.12	22.05
Google API	6.64	30.47	16.72
wit.ai	7.94	35.06	19.41
Deep Speech	**6.56**	**19.06**	**11.85**

Table 4.1: Results (%WER) for 5 systems evaluated on the original audio.

4.3 Design

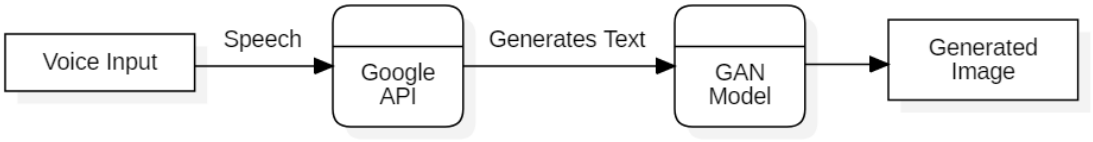

Fig 4.8: High level diagram of Voice-to-Image model

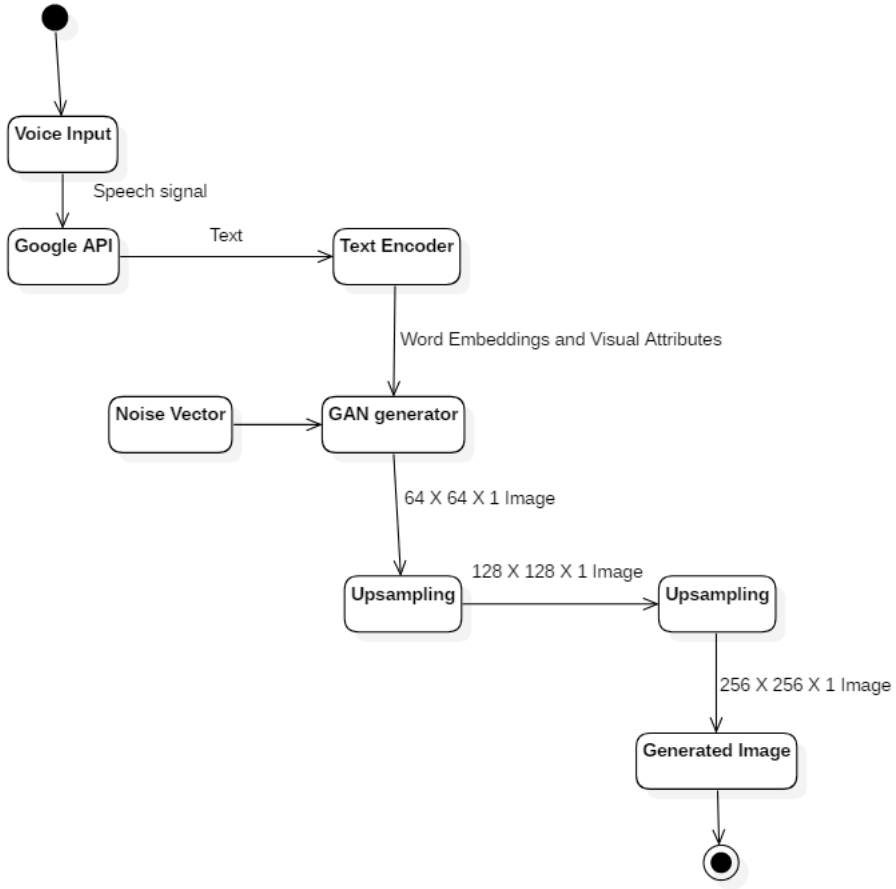

Fig 4.9: Activity diagram of Voice-to-Image model

- Speech waveforms are given as input to the Voice-to-Text model, that in turn generates the corresponding text.
- The text is then passed to the GAN model, specifically the Generator model, that generates multiple images from the text.

4.4 Pseudo Code

4.4.1 Speech to text

```
import speech_recognition as sr
import os
import sys
sys.path.append('/home/anita_damodaran_306/ffmpeg-4.2.1')

from pydub import AudioSegment
from pydub.silence import split_on_silence

def recognize():
    song = AudioSegment.from_wav(path)
    fh = open('recognized.txt', 'w')
    silence = AudioSegment.silent(duration = 10)
    audio = silence + song + silence

    r = sr.Recognizer()

    with sr.Microphone(device_index=0) as source:
        print('Listening...')
        r.adjust_for_ambient_noise(source)
        audio_listened = r.record(source, duration=5)

        try:
            progress(0.3)
            text = r.recognize_google(audio_listened)
            fh.write(text + '.')

        except:
            print('Sorry could not recognize your voice')

    fh.close()
    return text
```

4.4.2 Text to Image

```
def generate_bird(caption, copies=3):
 # load word vector
 captions, cap_lens = vectorize_caption(wordtoix, caption, copies)
 n_words = len(wordtoix)

 # only one to generate
 batch_size = captions.shape[0]

 hidden = text_encoder.init_hidden(batch_size)
 mask = (captions == 0)
 word_emb, sentence_emb = text_encoder(captions, cap_lens, hidden)
```

```
progress(0.6)
cap_lens = cap_lens.cpu().data.numpy()


fake_imgs, attention_maps, _, _ = netG(noise, sent_emb, words_embs, mask)

# storing to blob storage
container_name = "images"
full_path = "https://attgan.blob.core.windows.net/images/%s"
prefix = datetime.now().strftime('%Y/%B/%d/%H_%M_%S_%f')
imgs = []

progress(0.8)

for _ in range(batch_size):
    for k in range(len(fake_imgs)):
        im = fake_imgs[k][j].data.cpu().numpy()
        im = (im + 1.0) * 127.5
        im = im.astype(np.uint8)
        im = np.transpose(im, (1, 2, 0))
        image = Img.fromarray(im)
        image.save('{}_{}.jpg'.format(str(j), str(k)))
        if k==2:
            imgs.append(image)

   for i in range(len(imgs)):
       plt.figure()
       plt.imshow(imgs[i])
```

Chapter 5

RESULTS AND DISCUSSION

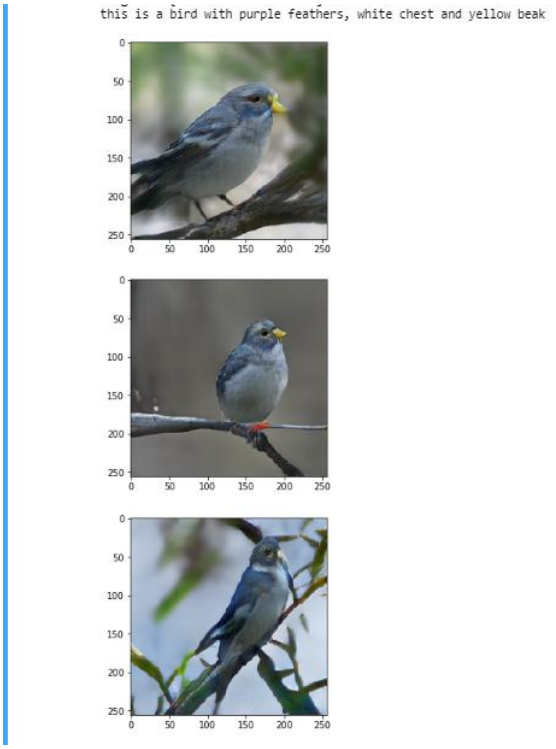

Fig 5.1 Bird with purple feathers, white chest and yellow beak

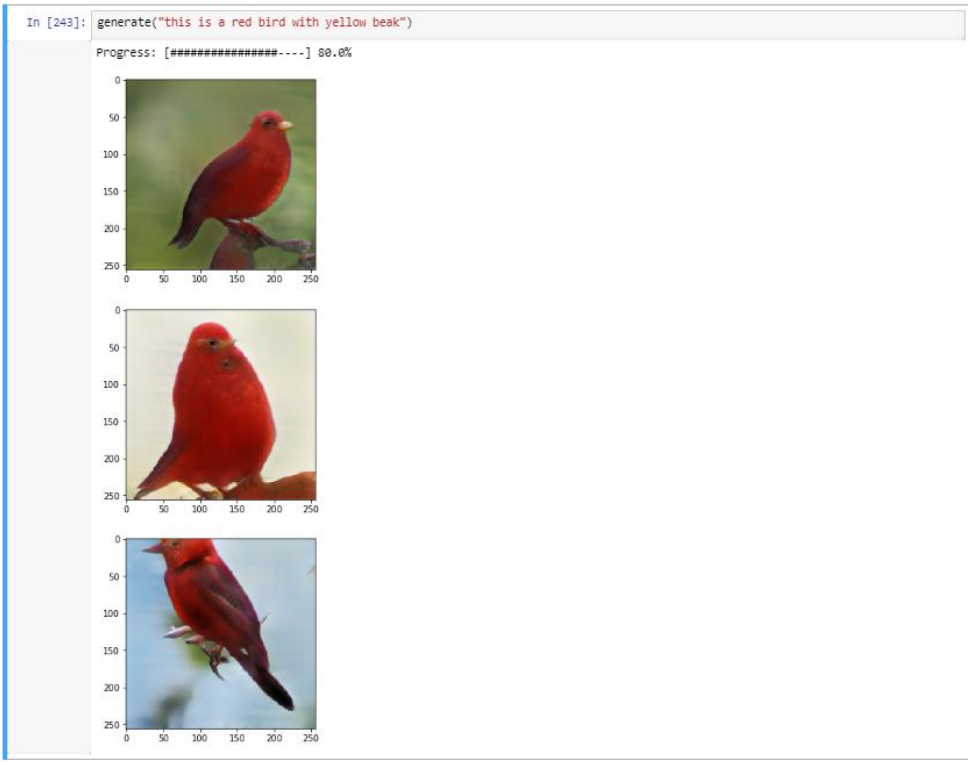

Fig 5.2: Red bird with yellow beak

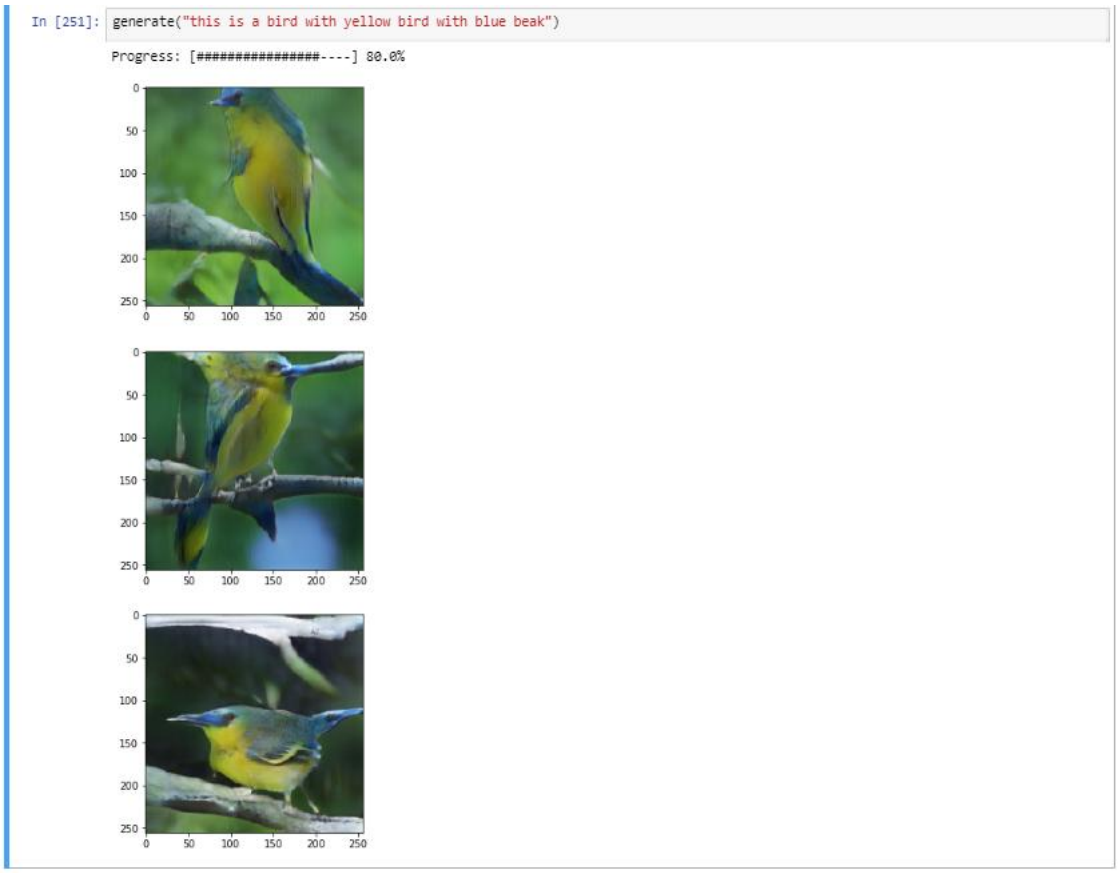

Fig 5.3 Yellow bird with blue beak

Fig 5.4: A clock that is on the side of a tower

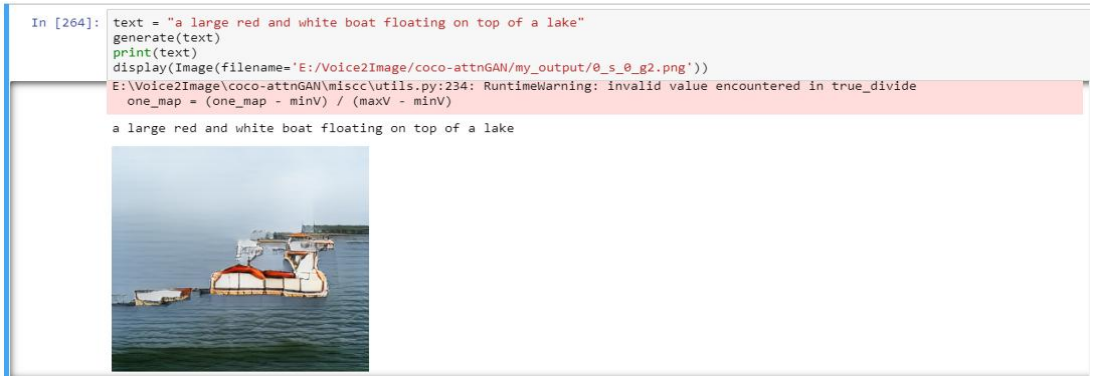

Fig 5.5: A large red and white boat floating on top of a lake

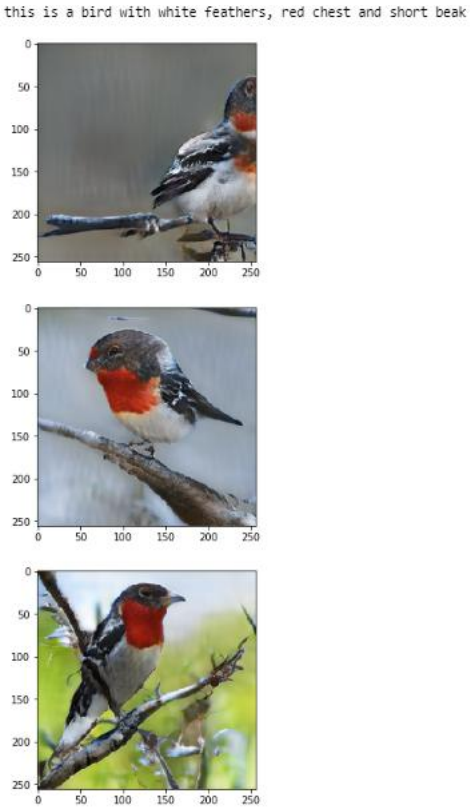

Fig 5.6: This is a bird with white feathers, red chest and short beak

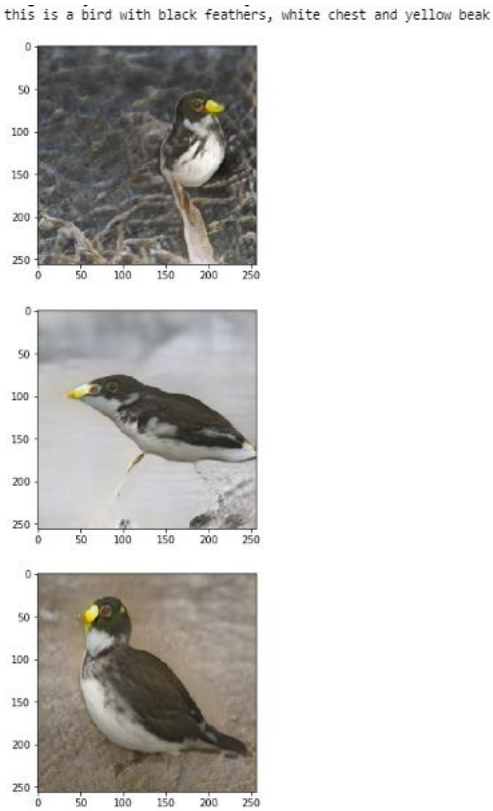

Fig 5.7: This ia a bird with black feathers, white chest and yellow beak

Fig 5.8: Flat screen television on top of an old tv console

Fig 5.9: A photo of a homemade swirly pasta with broccoli carrots and onions

Fig 5.1: A fruit stand display with bananas and kiwi

- A key point to notice among the above images generated is that the Generator model is able to generate significantly stronger vector representations for text, when the text belong to a specific domain.

- In this scenario, the images generated by the GAN model trained on Bird images, was able to generate much more accurate and varying images of bird, when compared to the COCO model.
- The visual attributes pertaining to birds follow a specific pattern and always employs the same set of descriptive words. Thus, providing for better understanding of the text relating to the image.
- Therefore, when the model is trained on images pertaining to a fixed domain of knowledge, it will learn textual features much more significantly than compared to a model on generalized text.

Chapter 6

CONCLUSION

The “Voice to Image” model proposed in this report was successful in generating images based on the voice input (Description of image). We were able to achieve the 3 fold objectives used for the development of this model. One of the models has been trained specifically on the “birds” dataset and other model on the “coco” dataset which consists of all possible objects observed in daily context. The model is successful in generating images of birds (or any other thing in case of coco) that doesn’t exist in real life. From the results, it has been observed that the system yields the most accurate results when trained using dataset corresponding to a specific domain than the ones trained on general datasets (i.e., The model trained specifically on birds dataset fetched much more efficient results than the one trained with coco dataset). Hence it can be said that the system can be used in the required fields such as interior designing, medical etc by training it with corresponding domain specific datasets.