Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766300

TEXT TO IMAGE SYNTHESIZER USING GENERATIVE ADVERSARIAL NETWORKS

J.Jayapradha*, B.Malathee Kumari, Divyadarshini.R, A.V.Haritha Sai Sri
[a]Assistant Professor,[b] Department of Computer Science Engineering, [c]Manakula Vinayagar Institute Of Technology,[d]puducherry.
[a]Manakula Vinayagar Institute of Technology, [b]Puducherry; [c]Manakula Vinayagar Institute of Technology, [d]Puducherry;
[a] Manakula Vinayagar Institute of Technology, [b]Puducherry; [c]Manakula Vinayagar Institute of Technology, [d]Puducherry.

Abstract— Computer vision is the field of artificial intelligence (AI) used in industrial applications. Text-to-image creation with Generative Adversarial Networks (GAN) is a deep learning model which can provide images from text descriptions. This project proposes the development of a text-to image synthesis system using Generative Adversarial Networks (GANs). The GAN architecture will be utilized to generate high-quality images from text descriptions. The system will be trained on a dataset of text descriptions and paired images. It will be evaluated on the quality of the generated images and the accuracy in capturing the semantic information from the text descriptions. The results will be compared against existing text-to-image synthesis systems. This project will provide a useful tool for researchers in the field of computer vision and natural language processing. Keywords— Artificial Intelligence, GAN(Generative Adversarial Networks) .

I. INTRODUCTION

In recent years, artificial intelligence (AI) has accelerated its integration into traditional industries and developed into various applications. The application used in our model is Computer Vision. Computer vision is artificial intelligence that enables computers and systems to give meaningful data from digital images, videos, and other visual inputs and take actions or make changes based on that data. By using computer vision systems at the edge, you can set up automated production inspection lines that detect product defects. Generative modeling is an unmanaged gaining knowledge of project in device gaining knowledge of that includes routinely coming across and gaining knowledge of the regularities or styles in enter records in this kind of manner that the version may be used to generate or output new examples that plausibly might have been drawn from the unique dataset.

GANs are a smart manner of education a generative version via way of means of framing the hassle as a supervised gaining knowledge of hassle with sub-fashions: GANs are an thrilling and swiftly converting field, turning in at the promise of generative fashions of their capacity to generate practical examples throughout quite a number hassle domains, maximum significantly in photograph-to- photograph translation duties including translating images of summer time season to iciness or day to night, and in producing photorealistic images of objects, scenes, and those that even people can't inform are fake. In this work, we are interested in converting text in the form of human- written single-sentence descriptions directly to image pixels. The problem of generating images from visual descriptions is of growing interest in the

research community, but is far from being solved. Eventually, this type of detailed visual data about an image has been provided by attribute representations, which are vector-encoded features of the object category, in particular to enable zero-shot visual detection, and more recently for conditional imaging. Properties of attribute representations are more brief attributes are also attractive to obtain as they may require for the text description related to domain-specific knowledge. In comparison, traditional language offers a general and flexible interface for describing objects in any kind of visual categories. Generative Adversarial Networks (GANs) are a strong class of neural networks used for unmaintained learning. GANs are learning, generating and improving, so they can create anything they give them. To understand GANs, you first need to know very little about convolutional neural networks. A CNN is trained to classify images in terms of their labels as they are fed into his CNN. A CNN analyzes an image pixel-by-pixel and passes it to the nodes in the hidden layer of the CNN, telling them what the image is or looks like as an output. In the photo. For example, if a CNN has been trained to classify dogs and cats, and the CNN is fed an image, it can determine whether the image is a dog or a cat. Therefore, it can also be called a classification algorithm. A Stack GAN is so called because it consists of several GANs that are stacked together to create a network that can produce high quality images. Artificial Intelligence (AI) refers to the elaboration of computer systems which can perform jobs that typically require human intelligence, such as understanding natural language, recognizing objects, making decisions, and solving problems. AI involves the development of algorithms and models that can learn from data and experience, and then use that knowledge to make predictions, automate tasks, or provide insights. There are various sub-domains of AI, including machine learning, deep learning, natural language processing, computer vision, and robotics. AI technologies are increasingly used in a wide range of suitability, embrace self-driving cars, virtual assistants, fraud detection, medical diagnosis, and many others. While AI has the potential to revolutionize many industries and improve people's lives in numerous ways, there are also anxiety about its effect on employment, privacy, and security, as well as the need for ethical considerations in the development and use of these technologies.

II. LITERATURE REVIEW

Generative image modeling is a common problem in computer vision. The arrival of deep learning techniques which made significant progress in this direction. Variational Auto encoders (VAE) formulated the problem using a probabilistic graphical model whose goal is to maximize the lower bounds on the data probabilities. Autoregressive models that use neural networks to model conditional distributions in pixel space (such as Pixel(RNN) have also produced satisfactory synthetic images. Recently, Generative Adversarial Networks (GANs) have shown the potential to generate sharper images. However, due to training instability, it is difficult for GAN models to generate high-resolution (e.g., 256x256) images. Several techniques have been proposed to stabilize the training process and achieve compelling results. Energy-based his GAN has also been proposed for more stable training behaviour. Based on these generative models, conditional image generation was also investigated. Most methods used simple conditioning variables such as attributes and class names. There is also image adjustment work on image generation, including photo manipulation, domain transfer, and super-resolution. Super-resolution techniques, however, are unable to correct significant flaws

like the proposed StackGAN and can only add a small amount of detail to low-resolution images. Recently, various methods have been prosperity to conceive images from unstructured text. Mansimonv et al. created the Align DRAW model by erudition to estimate the adequately between the text and the generated canvas. Reed et al. conditional He uses Pixel CNN to generate images using textual descriptions and object position constraints. Nguyen et al. previously owned by an approximate Langevin sampling way to generate text- related images. Their sampling strategy, meanwhile, necessitates a laborious iterative optimization procedure. Based on Reed et al textual .'s description, we were able to produce credible 64x64 images of birds and flowers using conditional GANs. In their follow-up work, they were able to generate 128 x 128 images with additional annotations on the positions of object parts. In addition to using a single GAN for image generation, there are also studies using many GANs for image generation. We divide the process of creating indoor scenes into structure generation and style generation using the S2-GAN developed by Wang et al. Denton et al constructed a series of GANs within the Laplacian pyramid framework. A residual image was generated at each level of the pyramid, adjusted based on the previous level's image, and added to the input image to generate the next level's input.

Huang et al. recently introduced a deep-learning model and it also found that stacking multiple GANs to reconstruct a multi-level representation of a pre-trained discriminative model can produce better images. showed. However, they were only able to generate 32x32 images of his, whereas our method uses a simpler architecture to generate 256x256 images with photorealistic detail and 64 times more pixels generate.

1. Ubiquitous defects are detected at the first stage by reconstructing the input tissue image using an enhanced GAN. Defective regions are located based on the difference amongst the input image and the rebuilt version.
2. The second stage is targeted error detection. Defect regions are used to guide the extraction of defect-related features from the generator. A central loss constraint is introduced to improve the detection performance.

Stacked Generative Adversarial Networks: To generate high-decision snap shots with photo-realistic details, we advise a easy but powerful Stacked Generative Adversarial Networks. It decomposes the text-to-image generative method into stages

Level I GAN: It determines the initial shape and basic colours of the object based on the given text description and the background structure draws a random noise vector to produce a low-resolution image.

Level II GAN: It corrects errors in the low-resolution image of the Level I and details the object of the by reading the text description, creating a high-resolution photorealistic image.

III. EXISTING TECHNOLOGY

StackGAN++: This system was developed by researchers at Carnegie Mellon University and Microsoft Research. It uses a two-stage GAN architecture to generate high-quality images from text descriptions. The first stage generates low- resolution images based on the text description, and the second stage refines these images to produce high-resolution images.

AttnGAN: This system was developed by researchers at the University of Washington and Microsoft Research. It uses an attention mechanism to focus on different parts of the text

description when generating the image. This allows the system to generate more diverse and detailed images.
MirrorGAN: This system was developed by researchers at Peking University and Microsoft Research Asia. It uses a mirror generator that takes both the text description and a partially generated image as input. The generator then adjusts the image to better match the text description, resulting in more accurate and realistic images. The current system for generating images from text using Generative Adversarial Networks (GANs) involves training a deep neural network on a large dataset of image-caption pairs. The network is typically composed of two main components: a generator network and a discriminator network.

The generator network takes in a textual description as input and generates an image that matches the description. The discriminator network takes in both the generated image and the corresponding textual description, and tries to determine whether the image matches the description.

During training, the generator network tries to generate images that fool the discriminator network, while the discriminator network tries to correctly distinguish between real and generated images. This back-and-forth process continues until both networks reach a point of equilibrium, where the generated images are indistinguishable from real images.

Once the GAN model is trained, it can be used to generate images from text descriptions. This involves feeding a textual description to the generator network, which then produces an image that matches the description. The resulting image can then be refined or adjusted as necessary to improve its quality or match the user's preferences.

Overall, the current system for text-to-image using GANs represents a promising approach to generating high-quality images from natural language descriptions, with many potential applications in fields such as computer vision, art, and design.
Preliminarie : Generative Adversarial Networks (GAN) consists of two models that are alternately trained to compete with each other. Generator G is optimized to reproduce real data distribution information by generating images that are difficult for discriminator D to distinguish from real images. In the meantime, D is tuned to discriminate between actual photos and fake images produced by G. Where x is the actual p data representation of the data distribution, and z is a noise vector sampled from a distribution (e.g uniform or Gaussian distribution).A variation of GAN called conditional GAN uses additional conditioning variables as both the generator and the discriminator, resulting in G (z, c) and D. (x, c). This formula allows G to generate variable images.

IV. MOTIVATION

The aim of our project work is to generate photorealistic images from given natural language descriptions. Previous work on Generative Adversarial Networks (GAN`s) has come a long way. All the same, it is still tough to bring forth intact objects and well-defined textures. Moreover, in text- to-image generation tasks, the set of text conditioning manifolds is often sparse due to the limited number of text- image pairs for training, and such a sparse set leads

to GAN becomes difficult to train. Therefore, we propose a new conditioning extension technique to promote smoothness of the underlying conditioning manifold. The diversity of the synthesised images is increased due to the minor random disturbances that are permitted in the conditioning diversity.

The Generative Module : Create a generator network that takes the text feature vector and a random noise vector as input and produces a high- resolution image as output. One common approach is to use a conditional GAN, where the text feature vector is concatenated with the noise vector and fed into the generator network. The generator network typically consists of multiple layers of up sampling and convolutional layers.

The Discriminative Module : The discriminational model operates like a normal double classifier that's suitable to classify images into different orders. It determines whether an image is real and from a given dataset or its instinctively generated.

The discriminational model falls under the supervised literacy branch. In a bracket task, given that the data is labelled, it tries to distinguish among classes, for illustration, a auto, business light and a truck. Also known as classifiers, these models correspond image samples X to class markers Y, and discover the probability of image sample. Create a discriminator network that takes an image as input and outputs a probability that the image is real or fake. The discriminator network can be a standard convolutional neural network that is trained to classify images as a real or fake.

Data Preparation Module :

Collect a dataset of textual descriptions and corresponding images. The dataset should be large enough to train a deep neural network. Preprocess the images by resizing them to a consistent size and normalizing the pixel values. Preprocess the textual descriptions by tokenizing them, converting them to lowercase, and removing punctuation and stop words.

Textual Attention Module:

Textual Attention Module is presented to modeling the mapping relationship between the word features e and the visual features h from former retired subcaste, which can induce fine-granulated visual details that are semantic connected to the textbook. First, we align the word and image features into the common semantic confines through a transfigure network. also, the attention weights are attained by calculating its applicability of word and vision features within fleck product and softmax normalization. Next, to acquired the word- environment vector ci for each sub- region of image, ci is reckoned by weighted averaging on the word features ai with the attention weights. Train a text encoder model to convert textual descriptions into a fixed- length feature vector. One common approach is to use a pre-trained language model, such as BERT or GPT-2, to extract features from the textual descriptions. Alternatively, you can train a text encoder from scratch using a recurrent neural network or a transformer network. Training Module: Train the generator and discriminator networks in an adversarial manner, where the generator tries to produce images that fool the discriminator, and the discriminator tries to distinguish between real and fake images. The training process involves alternating between updating the generator and discriminator networks. One common approach is to use the Adam optimizer and the binary cross- entropy loss function.

V. PROPOSED WORK

Text-to-image synthesis refers to the task of generating realistic images from textual descriptions. StackGAN is a deep neural network architecture designed for this task, which generates images in multiple stages, improving the visual quality of the output image. StackGAN consists of two stages of generative networks: the Stage-I generator and the Stage-II generator. The Stage-I generator takes the textual description as input and produces a low-resolution image as output. This low-resolution image is then fed to the Stage-II generator along with the textual description to produce a high-resolution image. The Stage-I generator uses a conditional GAN to generate a 64x64 image from the textual description. This generator learns to capture the overall structure and the rough colors of the object described in the text. The Stage-II generator takes the Stage-I output image and the textual description as input and generates a high resolution (256x256) image. It uses a stack of GANs to refine the output image in multiple stages, progressively adding details and increasing the resolution of the output image. StackGAN achieves state-of-the-art performance in text-to-image synthesis and produces realistic and diverse images from textual descriptions. The model has been trained on large-scale datasets such as the COCO dataset and has been used in a variety of operations, including computer vision, plates, and robotics

VI. IMPLEMENTATION

The text description t is first encoded with an encoder, resulting in the embedded text `t. In former work textbook embeddings were nonlinearly converted to induce latent tentative variables as input to the creator. However, the cache space for Text- entries is usually large (> 100 dimensions). With a limited amount of data, this usually results in a deadlock in latent data changes, which is undesirable. For learning the generator. To alleviate this problem, we introduce a exertion addition fashion to produce fresh exertion variables ĉ. In contrast to the fixed conditioning text variable c in, we randomly sample the latent variables ĉ from an independent Gaussian distribution N (μ (t), Σ (t)), where the mean μ(t) and slant covariance matrix Σ(t) are functions of the textbook bedding t.The proposed Conditioning addition yields more training dyads given a small number of image textbook dyads, and therefore encourages robustness to small disquiet along the exertion manifold. To further apply the smoothness over the exertion manifold and avoid overfitting, we add the following regularization term to the ideal of the creator during training, which is the Kullback-Leibler divergence (KL divergence) between the standard Gaussian distribution and the conditional Gaussian distribution. The randomness introduced in Conditioning Augmentation is useful for modeling text-to-image translations, since the same set usually corresponds to objects with different poses and appearances.

Data Preprocessing: To train the StackGAN model, we need to have a dataset consisting of pairs of images and their corresponding text descriptions. For example, we can use the CUB-200-2011 dataset, which consists of images of birds and their corresponding text descriptions. We can preprocess the dataset by resizing the images, normalizing the pixel values, and creating a vocabulary from the text descriptions.

Text Embedding: In the first stage of StackGAN, we convert the text descriptions into a fixed-length vector representation. To achieve this, we use a pre-trained word embedding model such as Word2Vector Glove. We feed the text descriptions through the embedding model to obtain

a vector representation for each word in the description. We then average these word vectors to obtain a single vector representation for the entire text description.

Conditioning Augmentation: To improve the diversity of the generated images, we use a technique called conditioning augmentation. In this technique, we generate multiple random noise vectors and use them to condition the image synthesis stage. By using different noise vectors, we can generate multiple variations of the same image.

Image Synthesis: In the second stage of StackGAN, we use the text embeddings and the conditioning noise vectors to generate high-quality images. We use a generative adversarial network (GAN) architecture to generate the images. The GAN consists of two neural networks: a generator network and a discriminator network. The generator network takes the text embeddings and the conditioning noise vectors as input and generates an image. The discriminator network takes the generated image and the corresponding text embedding as input and tries to distinguish between real and fake images. The two networks are trained together in an adversarial process to generate high-quality images that match the input text description.

Evaluation: To evaluate the quality of the generated images, we can use metrics such as Inception Score or Fréchet Inception Distance (FID). These metrics measure the quality and diversity of the generated images.

Stage-I GAN: Instead of directly generating a high-resolution image from the text description, we simplify the task by first generating a low-resolution image with a Stage-I GAN, which only focuses on capturing the rough shapes and correct colors of the object. Let t be the text embedding of the given description, generated by a pre-trained coder in this article. The Gaussian conditioning variables cˆ0 for the text input are to obtain the values of `t with variations. Minimize LG0 where the actual image is I0 and the text description t p data is the data distribution. z is a noise vector randomly selected from the given distribution pz (Gaussian distribution in this document). λ is a normalization parameter that combines the two terms in Eq. (4). We set λ = 1 for all our experiments. Using the reparameterization trick introduced in, both μ0(t) and Σ0(t) are studied along with the rest of the network.

Model Architecture: To obtain the text state variable ĉ0 for the G0 generator, the input text t is first are given to the fully connected layer to generate μ0 and σ0 (σ0 are values on the diagonal of Σ0) for the Gaussian distribution N (μ0(t), Σ0(t)). ĉ0 is chosen after the Gaussian distribution. Our Ng-dimensional conditioning vector ĉ0 is computed by ĉ0 = μ0 + σ0 ⊙∈ (where # is the element-wise multiplication, ∈ ←·· N(0, I)). Then ĉ0 is convolved with the Nz-dimensional noise vector to generate the image W0 xH0 through a series of upscaling blocks. For a D0 discriminator, the textbook input t is first compressed to Nd using a completely connected subcaste of size and also spatially replicated to form a tensor Md ⇥ Md⇥ Nd. Meanwhile, the image is subsampled through a series of blocks until it has a spatial dimension of Md ⇥ Md . The image filter map is then concatenated to texture tensors along the channel dimension. The resulting tensor is further fed into a 1⇥1 convolutional layer to learn features about the image and text. Finally, a fully connected layer with one node is used to generate the decision value.

B. Stage-II GAN : Low-quality images generated by Phase I GANs usually lack clear parts of the object and may have shape distortions. Some texture detail can also be dropped in the first

stage of the, which is important for producing photorealistic images. Our Tier II GAN builds on the results of Tier I GAN to produce high quality images. Due to low quality images as well as text input, correct errors are returned in Phase I results. Phase II GAN complements previously ignored texture data to generate more photorealistic details. Due to the low resolution result S0 = G0(z, ĉ0) and Gaussian latent variables ĉ0, the discriminator D and the generator G are trained in phase II GAN by an alternative using LD and minimize LG.

Unlike the original GAN formula, the random noise z is not used in this step, considering that the random noises are already stored by s0. The Gaussian exertion variables used in this stage and ĉ0 used in stage I partake the GAN the same pre-trained textbook encoder, generating the same text book input t. Still, Stage- I and Stage- II Augmentation Conditioning have completely connected layers to induce different means and standard diversions. In this way, the Phase II GAN learns to extract useful information in the text that the Phase I GAN misses.

We design Stage-II generator as an encoder-decoder network with residual blocks . Similar to the previous stage, the text embedding t is used to generate the Ng dimensional text conditioning vector ĉ, which is spatially replicated to form an Mgx, Mgx, Ng tensor.The image features and the text features are concatenated along the channel dimension. The decoded image features coupled with textbook features are fed into several residual blocks which are designed to learn multi-modal representations across image and textbook features.

Finally, a series of up-sampling layers 5910 (i.e., decoder) are used to generate a W x H high-resolution image. Such a generator is capable of correcting errors in the input image by adding additional details to produce a realistic high- resolution image. Use the StackGAN architecture to generate images at multiple resolutions. This involves training multiple generator and discriminator networks, where each network is responsible for generating images at a specific resolution. The outputs of the lower-resolution generator networks are fed into the higher-resolution generator networks to produce higher-quality images.

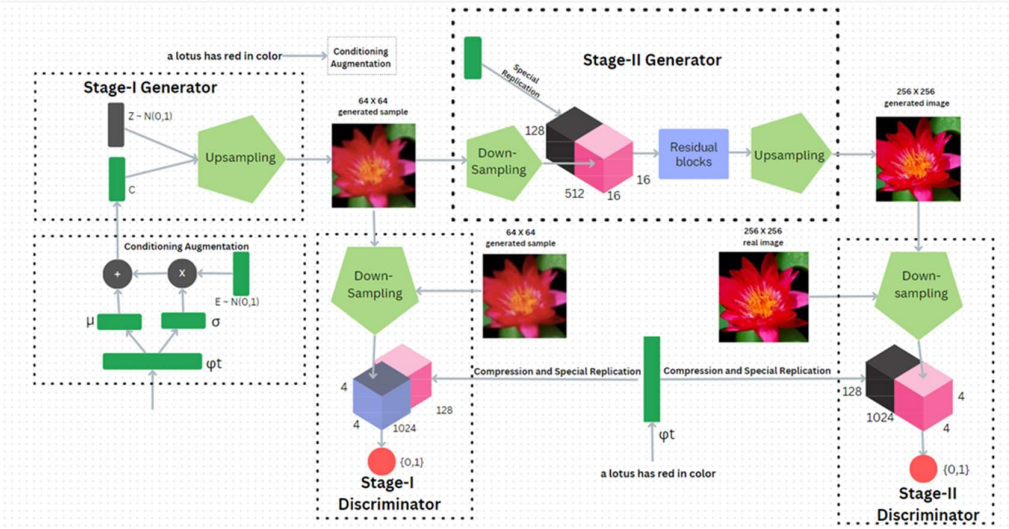


Fig 1.1 Architecture diagram of Text To Image Synthesizer Using StackGAN

The structure of the discriminator is analogous to that of the Stage I discriminator, with only the lower sample blocks of the because the image size is larger in this stage. To explicitly force the GAN to learn a better match between the conditioned image and text, instead of using vanilla discrimination, we adopt the Reed et al. for both levels. During training, the discriminator takes real images and corresponding textual descriptions as positive pattern dyads, while negative pattern dyads are divided into two groups. The former are real images with inappropriate text overlays, while the latter are synthetic images with appropriate text overlays.

VI. EXPERIMENT & DISCUSSION

To validate our method, we conduct complicated quantitative and qualitative evaluations. Two state-of-the-art text-to-image synthesis methods, GAN-INT-CLS and GAWWN , are compared. The results of the two compared methods are generated by the code published by their authors. In addition, we design several basic models to verify the general design and main components of the proposed StackGAN. For the first step, we directly train the Phase I GAN to test 64x64 and 256x256 images to find out whether the proposed stacked structure and the conditioning extension are useful. Next, we modify our StackGAN to generate 128x128 and 256x256 images to check whether larger images yield higher image quality using our method. We also check whether text input is useful in both stages of StackGAN.

During training, the generator networks are trained to produce high-quality images that match the input text descriptions, while the discriminator network is trained to distinguish between real images and generated images.

The StackGAN model has been evaluated on several datasets, such as the Oxford-102 Flowers dataset and the CUB-200-2011 dataset, and has shown impressive results in generating high-quality images that match the input text descriptions.

The discussion of StackGAN and its performance in text-to- image synthesis involves several factors, such as the quality of the generated images, the diversity of the generated images, and the efficiency of the model in terms of training time and computational resources. While the StackGAN model has shown promising results in generating high-quality images that match the input text descriptions, there are still limitations and challenges that need to be addressed, such as the difficulty in generating images with complex backgrounds and the lack of diversity in the generated images. In conclusion, the StackGAN model is a successful approach for text-to-image synthesis, which has shown impressive results in generating high-quality images that match the input text descriptions. However, there is still room for improvement, and future research can focus on addressing the limitations and challenges of the model to enhance its performance further.

Datasets and evaluation metrics: CUB carries 2 hundred chook species with 11,788 photos. Since 80% of birds on this dataset have object-picture length Ratios of much less than 0.five, as a pre-processing step, we Crop all photos to make certain that bounding packing containers of birds have greater-than-0.seventy five object-picture length ratios. Oxford-102 carries 8,189 photos of plant life from 102 distinctive categories. To display the generalization functionality of our approach, an extra hard dataset, MS COCO is also applied for evaluation. Different from

CUB and Oxford102, the MS COCO dataset carries photos with multiple items and diverse backgrounds. It has a schooling set with 80k photos and a validation set with 40k photos. Each picture in COCO has five descriptions, at the same time as 10 descriptions are supplied via way of means of for each picture in CUB and Oxford102 datasets. Following the experimental setup in, we at once use the schooling and validation units supplied via way of means of COCO; in the meantime we cut up CUB and Oxford-102 into class-disjoint schooling and take a look at units.

Evaluation metrics: One commonly used metric is Inception Score (IS), which measures the quality and diversity of generated images. It is calculated by feeding generated images to a pre-trained Inception-v3 model and computing the KL divergence between the predicted class distribution and the marginal distribution of the generated images. Higher IS values indicate better quality and diversity of generated images.Another metric is Frechet Inception Distance (FID), which also measures the quality and diversity of generated images. It calculates the distance between the feature representations of real and generated images extracted from a pre-trained Inception-v3model.
Perceptual Evaluation of Image and Video Quality (PEIV) is another metric used to evaluate the visual quality of generated images. It measures the similarity between generated and real images using several perceptual factors, such as luminance, contrast, and structural similarity. Finally, Precision and Recall scores can also be used to evaluate the performance of StackGAN. Precision measures the percentage of generated images that are correctly classified as matching the input text, while recall measures the percentage of input texts that are correctly matched to generated images. Higher precision and recall values indicate better performance of the StackGAN model. It is tough to assess the overall performance of generative models (e.g., GAN).

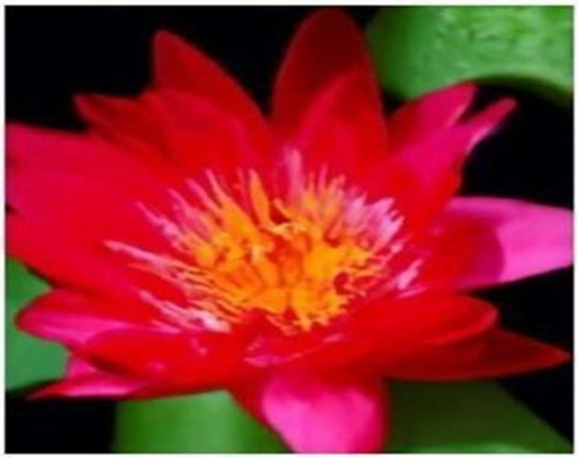
We pick these days proposed numerical evaluation approach “inception score” for quantitative evaluation, Wherein x denotes one generated sample and y is the label Expected with the aid of using the Inception version. The instinct behind this metric is that right fashions must generate numerous but significant images. Thus, the KL divergence between the borderline distribution p(y) and the tentative distribution p(y| x) must be huge. In our experiments, we directly use the pre-educated Inception version for COCO dataset. For fine-grained datasets, CUB and Oxford-102, we fine-tune an Inception version for every of them. As advised in , we compare this metric on a huge wide variety of samples (i.e.,30k randomly decided on samples) for every version. Although the inception rating has proven to properly correlate with human belief on visible excellent of samples, it cannot mirror whether or not the generated pics are properly conditioned at the given textual content descriptions. Therefore, we also behaviour human evaluation. We randomly pick 50 textual content descriptions for every magnificence of CUB and Oxford-102 check sets. For COCO dataset, 4k textual content descriptions are randomly decided on from its validation set. For every sentence, five pics are generated through every model. Given the identical textual content descriptions, 10 customers (now no longer which includes any of the authors) are requested to rank the consequences through exceptional methods. The average Ranks through human customers are calculated to assess all as compared methods.

```
#generative stage 1
prompt = "a lotus has red in colour"
batch_size = 1
guidance_scale = 3.0

# Tune this parameter to control the sharpness of 256x256 images.
# A value of 1.0 is sharper, but sometimes results in grainy artifacts.
upsample_temp = 0.997
```

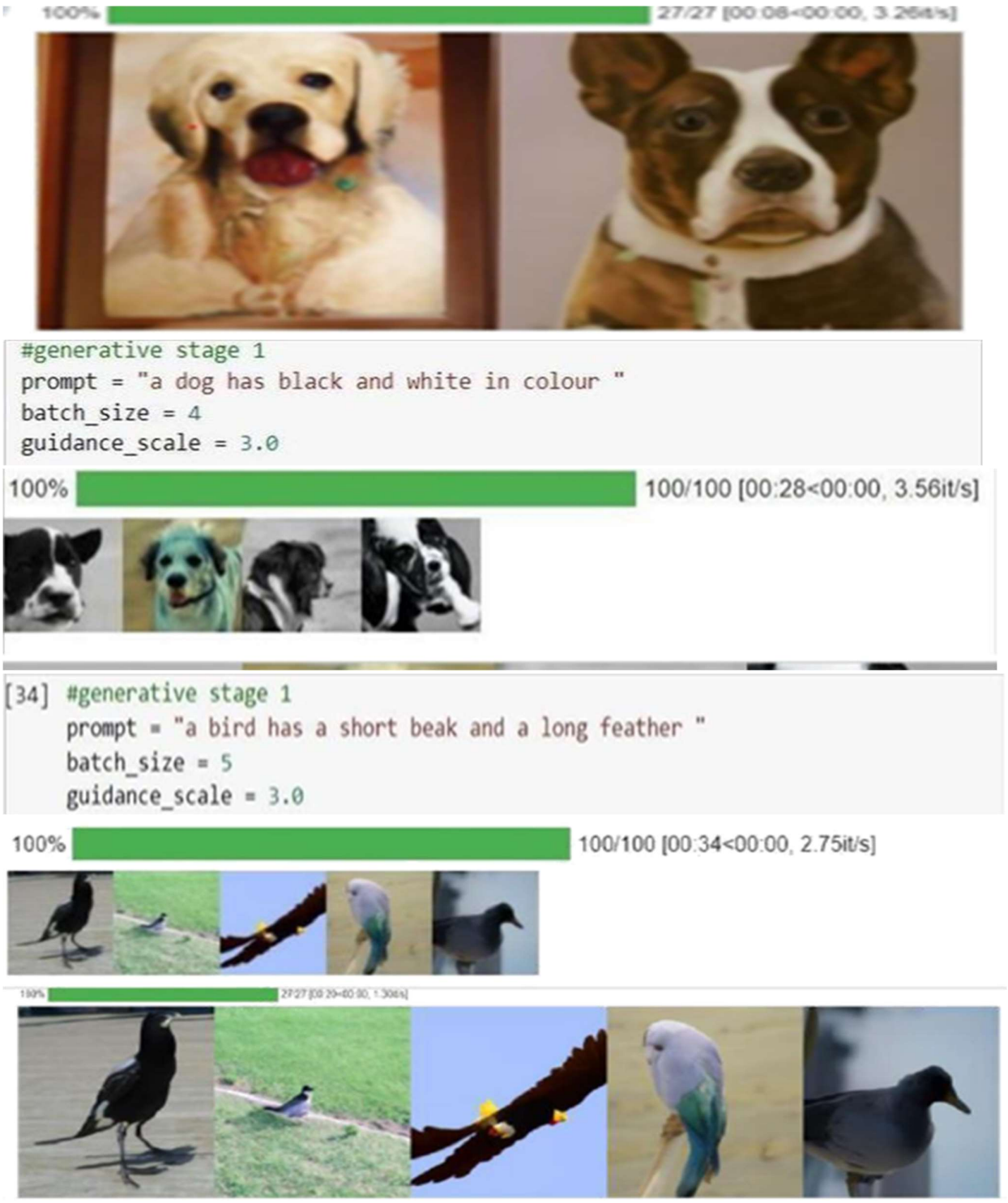
```
100% 100/100 [00:12<00:00, 8.92it/s]
```

```
100% 27/27 [00:05<00:00, 4.39it/s]
```



```
#generative stage 1
prompt = "an oil painting of a dog "
batch_size = 2
guidance_scale = 3.0
```

```
100% 100/100 [00:16<00:00, 6.24it/s]
```



VIII CONCLUSIONS & FUTUREWORK

In this paper, we endorse Stacked Generative Adversarial Networks (StackGAN) with Conditioning Augmentation for synthesizing photo-sensible photos. The proposed approach decomposes the textual content-to-photo synthesis to a novel sketch-refinement process. Stage-I GAN sketches the item following primary shade action and form constraints from given

textual content descriptions. Stage-II GAN corrects the defects in Stage-I outcomes and provides extra details, yielding better decision photos with higher photo quality Expansive quantitative and qualitative issues show the effectiveness of our proposed approach. Compared to current textual content- to- print generative models, our approach generates better decision prints(e.g., 256x256) with redundant print-sensible details and diversity.

StackGAN is a deep learning architecture that can generate high-resolution images from text descriptions. Here are some potential areas for future work in text-to-image generation using StackGAN:

Bettered textbook embedding:The quality of generated images is largely dependent on the quality of the textbook embeddings used as input. Experimenters can explore better ways of garbling textual information similar as using pre-trained language models like BERT or GPT-3.

Multi-modal input: The current StackGAN architecture only takes text descriptions as input. However, incorporating other modalities like audio or video could lead to more diverse and creative image generation.

Domain-specific image generation: The current StackGAN model is trained on a general dataset, which may not be suitable for specific domains like medical imaging or fashion. Researchers can explore domain- specific training datasets and fine-tune the StackGAN architecture accordingly.

Improved training techniques: Adversarial training, which is used in StackGAN, can be challenging to train and susceptible to mode collapse. Researchers can explore alternative training techniques like energy-based models or contrastive learning to overcome these challenges.

Controllable image generation: The current StackGAN model generates images from a single text description, with no control over the visual attributes of the generated image. Future work could focus on developing controllable text-to-image models that allow users to specify different visual attributes like color, shape, and texture.

REFERENCES

1. T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In ICLR, 2017.
2. C. Doersch. Tutorial on variational autoencoders. ArXiv: 1606.05908, 2016.
3. A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML, 2017.
4. S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In CVPR, 2016.
5. S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. de Freitas. Generating interpretable images with controllable structure. Technical report, 2016.
6. Md. Zakir Hossain 1 , (Student Member, Ieee), Ferdous Sohel 1 , (Senior Member, Ieee), Mohd Fairuz Shiratuddin1 , Hamid Laga 1 , And Mohammed Bennamoun 2 , (Senior Member, Ieee), ‘‘Text to Image ImageCaptioning,’’2021,r10.110.[Online].Available:ht tps://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumb er=9416431

7. Fangxiang Feng, TianruiNiu, Ruifan Li, Member, IEEE, and Xiaojie Wang, ‘‘Modality Disentangled Discriminator for Text-to-Image Synthesis,’’ 09:18:10UTCfromIEEEXplore. Available:https://sci-hub.se/10.1109/tmm.2021.3075997 .
8. S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, ‘‘Generative adversarial text to image synthesis,’’ in Proc. ICML, New York, NY, USA, 2016.
9. Mesay Belete Bejiga Student Member, IEEE, Genc Hoxha, Student Member, IEEE, and Farid Melgani, Fellow, IEEE.”Improving Text Encoding For Retro RemoteSensing”June14,2020at07:40:27 Available: https://sci-hub.se/10.1109/lgrs.2020.2983851.
10. H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, ‘‘Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,’’ in Proc. ICCV, Venice, Italy, Oct. 2017, pp. A5907–5915.
11. A. Krizhevsky, I. Sutskever, and G. E. Hinton,``ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Pro-cess. Syst., 2012, pp. 1097_1105.
12. K.Simonyan and A. Zisserman, ``Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015, pp. 1_14.
13. K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,Jun. 2016, pp. 770_778.
Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ``Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis.Pattern Recognit. (CVPR),Jul. 2017, pp. 2261_2269.
14. M. Z. Hossain, F. Sohel, M. F. Shiratuddin, H. Laga, and M. Bennamoun,`Bi-SAN-CAP: Bi-directional self-attention for image captioning,"in Proc. Digit. Image Comput., Techn. Appl. (DICTA), Dec. 2019, pp. 1_7.
15. H.Wei, Z. Li, C. Zhang, T. Zhou, andY. Quan, ``Image captioning based on sentence-level and word-level attention," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2019, pp. 1_8.
16. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona,
D. Ramanan, P. Dollár, and C. L. Zitnick, ``Microsoft COCO: Common objects in context," in Proc. Eur. Conf. Comput. Vis. Zürich, Switzerland: Springer, 2014, pp. 740_755.
17. B. A. Plummer, L.Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, ``Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 2641_2649.
18. M. Hodosh, P. Young, and J. Hockenmaier, ``Framing image description as a ranking task: Data, models and evaluation metrics," J. Artif. Intell. Res., vol. 47, pp. 853_899, Aug. 2013.
19. Dr.N. Palanivel, “Identification of Leaf disease using Fuzzy c-mean and kernal fuzzy c-mean and suggesting the pesticides”, International Journal of Advanced Research in Science, Engineering and Technology (IJARSET), Volume 2 Issue 2, May 2017.

20. S. Hinterstoisser,V. Lepetit, P.Wohlhart, and K.Konolige, ``On pre-trained image features and synthetic images for deep learning," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 682_697.
21. K. Seetharaman, N. Palanivel and R. Indumathi "Post Invariant Face Recognition Using HMM and SVM with PCA for Dimensionality Reduction", International Journal of Advanced and Innovative Research ISSN : 2278-7844, Volume 3, Issue 1, January 2014.
22. K. Seetharaman and N. Palanivel "Texture characterization, representation, description, and classification based on full range Gaussian Markov random field model with Bayesian approach", International Journal of Image and Data Fusion, ISSN: 1947-9832 Volume 4, Issue 4, 2013, pages 342-362.