



An Improved Image Generation Conditioned on Text Using Stable Diffusion Model

Sara Faez Abdulghani^a, Ashwan Anwer Abdulmunem^b

^aDepartment of Radiological Techniques, College of Health and Medical Techniques, Al-Zahraa (a.s) University for Women, Karbala, Iraq.
sara.faez@alzahraa.edu.iq

^bDepartment of Computer Science, college of Computer Science and Information Technology, Karbala University, Karbala, Iraq.
ashwan.a@uokerbala.edu.iq

ARTICLE INFO

Article history:

Received: 05 /09/2024

Rrevised form: 19 /09/2024

Accepted : 1 /10/2024

Available online: 30 /12/2024

Keywords:

Artificial Intelligence,

Text-to-image generation,

Stable Diffusion model,

Realistic Images,

and Generative model

ABSTRACT

One technique for creating visuals that correspond to textual descriptions is called "text-to-image generation." It affects a wide range of applications and research fields (e.g., photo-editing, photo-searching, art-making, computer-aided design, image reconstruction, captioning, and portrait drawing). With the development of text-to-image generation models, artificial intelligence (AI) has reached a turning point where robots are now able to convert human language into aesthetically beautiful and coherent images, creating new opportunities for creativity and innovation. The creation of stable diffusion models is one of this field's most noteworthy developments. These models provide a strong framework for producing realistic images that are semantically linked with the given textual descriptions. But even with their remarkable abilities, conventional text-to-image models frequently have serious shortcomings, especially when it comes to training timeframes and computing costs. These models can be costly and time-consuming to train because they usually need large amounts of processing power and long training times. The main goal of this work is to develop a better Stable Diffusion model to overcome these shortcomings and produce high-quality images from text. The suggested model will drastically cut down on training durations and processing needs without sacrificing the quality of the output photos. The proposed method shows that the fine-tuning of the Stable Diffusion model results in a considerable improvement in producing images that are more akin to the original. The results of the improved model denoted a lower FID score (212.52) when contrasted with the base model (251.22).

MSC..

<https://doi.org/10.29304/jqcm.2024.16.41772>

1. Introduction

When humans hear or read a story, they immediately visualize images in their minds envisioning the content in their heads. The ability to visualize and understand the intricate relationship between the visual world and language is so natural that we rarely think about it. Visual

*Sara Faez Abdulghani

Email addresses: sara.faez@alzahraa.edu.iq

Communicated by 'sub etitor'

mental imagery or “seeing with the mind’s eye” also plays an important role in many cognitive processes such as memory, spatial navigation, and reasoning [1], [2], [3]. Text-to-image generation concentrates on creating images from provided written prompts. For instance, given the input “A head of broccoli made out of modeling clay, smiling in the sun,” we would like the model to be able to generate an image that accurately matches the text prompt [4]. Inspired by how humans visualize scenes, building a system that understands the relationship between vision and language, and that can create images reflecting the meaning of text descriptions, is a major milestone toward human-like intelligence [5], [6]. In the last few years, computer vision applications and image processing techniques have greatly benefited from advancements enabled by the breakthrough of deep learning. One of these is the field of image synthesis which is the process of generating new images and manipulating existing ones [7]. Image synthesis is an interesting and important task because of many practical applications such as art generation, image editing, virtual reality, video games, and computer-aided design [8], [9]. In the digital era, the process of image production from the main textual input was described. The methods were specifically dependent on the stable diffusion models [10], [11]. As for the generative models, they have contributed much to the development of artificial intelligence letting generate rather realistic images from the text. They have been used in various forms and areas in scope including, but not limited to, artworks and designs, data sampling, and entertainment [12]. The transformation among these models signified significant progress in AI as these machines are capable of interpreting the human language and transforming it into visual forms inclusive of aesthetics in a picture form that is deemed appealing to the eye hence paving the way to greater opportunities as regards to creativity and resourcefulness. Unquestionably, one of the defining milestones in this sphere is known as the Stable Diffusion model. These models provide a solid foundation for generating different variations of the images that are semantically close to the textual descriptions. Though, such models are terrific in terms of performance, even they have issues with certain restrictions such as high computational complexity and training time [13]. In other words, most of these models are computationally intensive and take a lot of time to train which poses a disadvantage because it is costly. To meet these limitations, the main aim of this paper is to provide a new generation model that will enhance the creation of images from the text. The objective is to develop a model that ideally, retains or improves the quality of the generated images while drastically cutting down on the processing power and training time. Consequently, in the context of this work, emphasis is laid on producing floral pictures from textual data. This research uses flowers as objects of consideration because they represent a rather intricate object, given the ability of generative models to analyze them. The concept of generating images of flowers from text entails the exposure of minute features of flowers such as the formation, color, texture, and arrangement of petals, therefore making it useful in the assessment of the effectiveness of the proposed method. Generative models for the text-to-image synthesis task have advanced quite a lot, as models output incredibly realistic images based on text input. However, some critical issues are yet to be solved, especially with regard to the time consumption and the training time [13], [14]. These issues are addressed by this research through the formulation of a problem that deals with generating images with high quality and efficiency from text descriptions using a Stable Diffusion model. The core problem is related to the computational complexity and heavy training times that are inevitable in conventional text-to-image models. These models, even though can produce high-quality images, require a high-end machine and a large amount of time for training. This requirement hinders ease of use for researchers and practitioners with restricted computational capacity. Therefore, there is a critical necessity to determine an enhanced method that provides the same or even better quality of the image but with less computational complexity. Traditional text-to-image models typically involve multiple stages: Three types of operations include text encoding, the manipulation of the latent space, and image decoding, where all three are very delicate processes that need a lot of time and computational power [12]. The earlier suggested models like Generative Adversarial Networks (GANs) [5] and Variational Autoencoders (VAEs) [15], [16], have demonstrated certain levels of success, but are based on iterative optimization procedures which are computationally expensive and less efficient. However, these models often imply large datasets for improving performance; therefore, a high demand for computational resources can be expected. Where T is the set comprised of textual descriptions and I is the set of images. The objective is to develop a function $f: T \rightarrow I$ that translates a text description $t \in T$ into a photographic image that is $i \in I$. Conventional approaches demand a lot of computing power C and time τ to solve the problem of generating an image i with high-quality Q . The latter can be measured by specific performance indicators, namely IS and FID. The rest of this paper is organized as follows: Section II presents the literature review of previous work, the third section presents the proposed methodology that solves the research problem. While the fourth section presents the results achieved using the method used. The conclusion of this research was presented in Section Five.

NOMENCLATURE

Aradius of

Bposition of

Cfurther nomenclature continues down the page inside the text box

2. Literature Review

This section provides an analysis of several studies related to the topic of text-to-image generation. The goal of analyzing these studies is to identify research gaps and select a suitable model that can address them. Table 1 summarizes some of the studies that were analyzed in relation to text image generation.

Table 1: summary of previous work about image generation from text

Reference	Dataset	Model	Limitation
[17]	ImageNet 128×128	ddpm	Access to labeled datasets is limited.
[21]	MS COCO	ChatPainter's architecture	In many circumstances, the results are unrecognizable. Training the model using conversation data is also quite unstable.
[18]	oxford flowers-102	CanvasGAN	discontinuity in higher-dimensional latent mapping because of insufficient data
[19]	-OXFORD-102 cub	Conditional GAN	No assessment metric exists.
[20]	CUB <i>COCO</i>	AttnGAN	Not quite accurate in representing the world's cohesive structures.
[21]	CUB <i>COCO</i>	CAGAN	fails to create realistic-looking photos, although scoring better ISs than the AttnGAN.
[22]	ImageNet	Diffusion-CLIP	Perhaps used to deceive individuals with modified realistic consequences.
[23]	CUB-200 OXFORD-102 MSCOCO CONCEPTUAL CAPTIONS LAION-400M FFHQ256	VQ-Diffusion	Token substitution can have a major impact on the semantics of the port representation. The model does not know which tokens have been substituted, which increases its robustness throughout the denoising process.
[20]	OXFORD CUB	LEICAGAN	In the current implementation, the TVE models received independent training from the MPA and CAG models.

3. Proposed Methodology

This study highlights the process of problem formulation, data pre-processing, model setup, training strategies, validation, and assessment tools used to bring advanced generative models to more people. Through the reduction of computational costs and training time, these models can be made accessible to fewer resources, advancing creativity and invention in diverse industries and applying artificial intelligence in areas not possible before.

This study direct attention to creating floral images from textual data, using flowers as a complex object due to the ability of generative models to analyze them. The process involves shedding light on precise features of flowers, such as formation, color, texture, and arrangement, to evaluate the effectiveness of the proposed method.

This section demonstrates a proposed approach to generate images from text, concentrating on problem formulation, methodological descriptions, and utilization of pre-trained models in the Stable Diffusion pipeline. The difficulties and goals associated with image generation were described, arranging the groundwork for the study's specific concerns.

In this study, pre-trained models for feature engineering, text embedding, and image generation were engaged. Each model experienced evaluation and modification to achieve optimal performance. The training hyperparameters clarify the training process and influence model accuracy and performance. The best hyper-parameter values are determined for each task, by systematically employing domain values. Pseudocode for training and evaluation is provided, to verify repeatability and follow the same process.

The guide presents an inclusive guide on the Stable Diffusion model, explaining the steps from data initialization and preprocessing to training and validation, and acts as a practical reference for flower generation. Last, we explain the steps or the evaluation metrics that are used in the assessment of the model. Hence, the Inception Score (IS) [1] is an evaluation method used to calculate the quality of the images generated and the Fréchet Inception Distance (FID) [2] is another metric that determines image diversity.

3.1 . Evaluation Method

The evaluation methodology involves assessing the performance of the Stable Diffusion model using two key metrics: IS and FID are two of them which are respectively known as Inception Score and Frechet Inception Distance. These metrics give a measure of how good and how varied the generated images are.

3.1.1.Inception Score (IS)

The Inception Score is used to assess the quality of generated images through the use of the Inception v3 model. It measures two properties:

- Image Quality: How different the generated image is.
- Diversity: To what extent they are different from each other the images in the generated set; it is computed thus:

$$\mathbf{IS} = \exp(\mathbb{E}_x \mathbf{D}_{\text{KL}}(\mathbf{p}(y | x) \parallel \mathbf{p}(y))) \quad (6)$$

where \mathbf{D}_{KL} is the Kullback-Leibler divergence, $\mathbf{p}(y | x)$ is the conditional label distribution and $\mathbf{p}(y)$ is the marginal label distribution.

3.1.2. Frechet Inception Distance (Fid)

The Frechet Inception Distance measures the similarity between the distributions of generated images and real images. It uses a pre-trained Inception v3 model to extract features and compares their mean and covariance:

$$\mathbf{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}) \quad (13)$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the mean and covariance of the features from the real and generated images, respectively.

3.2. Research Objectives:

In this section, Five research objectives have listed to illustrates well the contributions of this papers, see Equations 1,2,3,4 and 5 [24], [25].

1. Computational Costs Reduction:

$$\min \mathbf{C} = \min \sum_{k=1}^N c_k \quad (1)$$

where c_k = computational cost of the k-th component or process in the model;

N = total number of components or processes.

2. Training Time Minimization:

$$\min \mathbf{T} = \min \sum_{j=1}^M \tau_j \quad (2)$$

where τ_j = training time for the j-th iteration or epoch;

M = total number of iterations or epochs.

3. Image Quality Maintenance/Improvement:

$$\max \mathbf{Q} = \max(\mathbf{IS}, \mathbf{FID}) \quad (3)$$

where IS = Inception Score;

FID = Frechet Inception Distance.

These are metrics used to evaluate the generated images' quality.

IS can be calculated as:

$$\mathbf{IS} = \exp(\mathbb{E}_{x \sim p_g}[\mathbf{D}_{\text{KL}}(\mathbf{p}(y | x) \parallel \mathbf{p}(y))]) \quad (4)$$

where p_g = distribution of generated images;

$p(y | x)$ = conditional label distribution given image x ;

$p(y)$ = marginal distribution over all labels.

FID can be computed as:

$$FID = \|\mu_r - \mu_g\|^2 + \text{Tr} (\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \dots\dots\dots(5)$$

where μ_r and Σ_r = mean and covariance of real images' features.

μ_g and Σ_g = mean and covariance of generated images' features.

4. Ensuring Accessibility and Scalability:

Make the model easily extendable to other computational settings or platforms as one way of enhancing scalability and accessibility. This may involve a process of retraining and restructuring the model so that it can provide optimal performance on the different machines.

5. Create a Strong Evaluation Framework:

For the evaluation of the models, one needs to employ a number of general performance indicators. This involves computing IS and FID to give the degree of insanity of the generated images as well as the extent of diversity of the images.

3.3. Design Methodology

The general workflow of the Stable Diffusion model aimed at flower generation consists of several vital steps, which are described below to match the proposed Stable Diffusion model effectively and generate high-quality images from textual descriptions. This section describes the steps: data preparation for model construction, model testing, and selection, and focuses on the part concerning model improvement based on performance assessment. It is described in detail in the following flowchart:

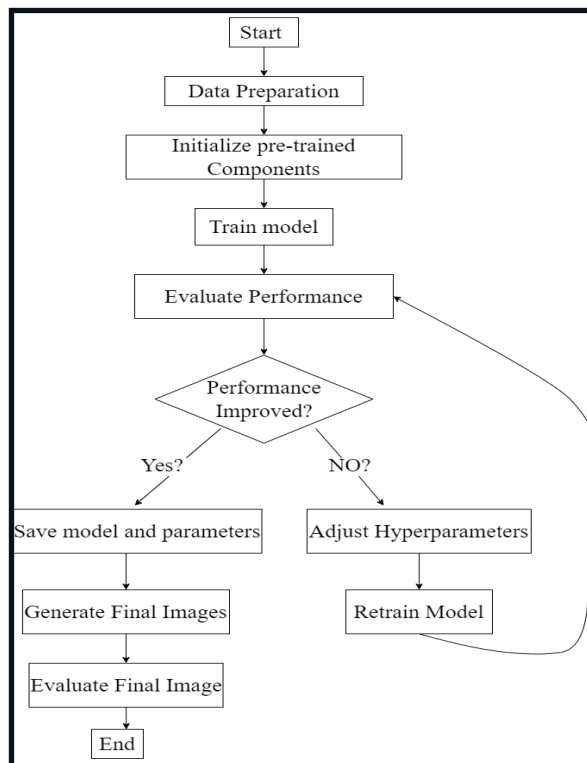


Figure 1: Methodology Flowchart for Stable Diffusion Model with Iterative Refinement

One of the essential features of the implementation of the outlined methodology is the iterative feedback procedure. If the metrics of the evaluation namely IS and FID do not improve then the hyper-parameters are tuned and the model is trained again. This iterative loop does guarantee steady improvement in any of the models involved in the analytic process. Especially, static parameters like the learning rate, the number of samples to be trained at once, and the number of training rounds are adjusted. The steps involved in this iterative refinement are as follows: Optimize Performance, compute IS and FID of the generated images, compare the results to the previous iteration, and decide whether the hyper-parameters were set correctly, if not adjust and retrain the model. This evolutionary process is performed until the degree of improvement of the system performance is attained.

They involve the generation of the final images after the model has been optimized to produce good images. Input texts are given to the model along with the Image style transfer model generating images from such described texts. The last images acquired are also assessed to check their quality in compliance with the set criteria in the activity.

The generated images are then checked whether they are of good quality and such evaluation is done by using IS and FID on the final generated images. This step allows us to obtain images that not only are aesthetically pleasing but also contain the proper semantic meaning given by the optimized model. The individual steps of the approach as well as the conditional iterative refinement process are shown in figure 1. This flowchart gives a concise description of the entire approach, data preparation for evaluation, and how there is a cycle of improvement for performance enhancement.

3.4. Implementation of The Proposed Method

The given pseudo-code shows the steps in the training and testing of the chosen model known as Stable Diffusion which focuses on the generation of flower images based on textual descriptions. The process is divided into several key phases: This step involves separate processes including the initialization of models, training, testing, as well as validating all in a bid to have a desirable model.

Starting with the first phase – the initialization phase. In this phase, such components as tokenizer, text encoder, VAE (Variational Autoencoder), UNet, and the rest are being loaded. These pre-trained models are part of the Stable Diffusion pipeline. The latter belongs to the tokenizer which is designed to prepare the text descriptions to be read by the model. It can be then followed by the text encoder that translates such tokens into more meaningful embedding. The VAE encodes the images while removing all the high-dimensional elements and retaining image-relevant information. The UNet is a neural network architecture used when predicting noise and during the process of denoising. When these components are loaded, a Stable Diffusion pipeline is erected combining all these components.

The training phase includes repeated passes over the training data that contain text inputs and related images. When it comes to the training data records, the text prompts are first divided into tokens forming a sequence of tokens. These tokens transform to get text embedding with the help of the pre-training text encoder. At the same time, the input images are passed into a VAE to get the corresponding latent representations. To these latent vectors, Gaussian noise is added and during training, the model has to learn how to remove this noise, thus helping guide the model towards the right direction. The loop with the noise scheduler time steps consists of predicting the noise using the UNet and calculating the mean square error between the predicted noise and the actual noise added in the loop. This loss function quantifies the discrepancy of the noise prediction results. To update the parameters of the given model, backpropagation is carried out where the loss function used is MSE. This cycle of training improves the model's ability to generate images of high quality as it learns how to denoise the latent vectors.

The testing stage of the model includes the generation of images via the model from new text samples that it has not seen during the training. Therefore, for each record in the testing data, a text prompt is created and then processed to generate its tokenized and encoded form from the text encoder. Both the encoder and the decoder start with the latent vectors which are comprised of random noise, and the model gradually removes the noise by passing the vectors through the UNet. The final latent vectors are then generated and the VAE is used to decode these latent vectors to get the images. These generated images are stored or can be viewed for assessment purposes.

The validation phase involves assessing the quality of the generated images using two key metrics: Two metrics are widely used in the form of Inception Score (IS) and Frechet Inception Distance (FID). To do this, the IS assesses the quality of the images to the intended categories within the theme and the variance of the images produced. The FID quantifies how similar the distribution of the created images and the genuine images is in terms of the mean and covariance. These are outcomes that can be quantified and are necessary for the assessment of the model's performance.

If the performance metrics show that the model has not evolved better with the help of new, added features or if any parameter related to the performance of the model fails to improve, then the hyper-parameters are modified and the model is retrained. Such an approach of refining the content in cycles guarantees constant enhancement of the results. Parameters are adjusted to improve the training process including; learning rate, size in each batch, and number of iterations. Thus, the new hyper-parameters are used to retrain the model, and it is tested on its performance. The performance results are collected and compared to the baseline and if the desired improvement is not reached, then the loop is run again.

Algorithm 1 Stable Diffusion Model Training and Evaluation for Flower Generation

Require: Pre-trained models (Tokenizer, Text Encoder, VAE, U-Net), Noise

Scheduler, Training Data (text prompts, images)

Ensure: Trained Stable Diffusion Model, Generated Images

1: Initialization:

2: Load pre-trained Tokenizer, Text Encoder, VAE, U-Net

3: Construct Stable Diffusion Pipeline

4: Training:

5: for each record in Training Data (text prompts, images) do

6: Tokenize input text prompts

7: Encode input images into latent space via VAE

8: Add Gaussian noise to latents

9: for each iteration in noise scheduler timesteps do

10: Predict noise using U-Net

11: Compute mean squared error (MSE) loss

12: Perform backpropagation and update model parameters

13: end for

14: end for

15: Testing:

16: for each record in Testing Data (text prompts) do

17: Define text prompt for image synthesis

18: Tokenize the prompt

19: Initialize latent vectors with random noise

20: for each iteration in noise scheduler timesteps do

21: Predict noise using U-Net

22: Denoise latent vectors progressively

23: end for

24: Decode the final latents to generate the image using VAE

25: Save or display the generated image

26: end for

27: Validation:

28: Compute Inception Score (IS) for the generated images

29: Compute Frechet Inception Distance (FID) between generated images and real images

30: Output:

31: Input: Text prompt (e.g., "A field of sunflowers.")

32: Output: Generated image based on the prompt

Thus, the pseudocode encapsulates a detailed and cyclical workflow for training and testing the Stable Diffusion model concerning flower generation. It underlines the necessity of initialization with the pre-trained models, the strict division into the training and test phases, and the obligatory validation of the results with the help of professional criteria. The aspect of iteration guarantees that the model keeps on acquiring and updating the relevant information hence providing quality images generated from textual descriptions. It plays a key role in constructing the systematic and efficient model for the transformation of textual descriptions to visually good and semantically correct images.





4. Results And Discussion

This section presents the experimental design and results of our study on fine-tuning the Stable Diffusion model for improved image generation from textual descriptions. Building upon the methodology outlined in section2, we conducted a series of experiments to evaluate the performance of our fine-tuned model compared to the base model.

4.1. Dataset

Every machine learning model is trained and evaluated using data, frequently in the form of static datasets. The features of these datasets have a fundamental impact on the behavior of the model; for example, a model is unlikely to perform well in the real world if its deployment context does not match its training or evaluation datasets, or if these datasets reflect unwelcome societal biases. A watershed moment in the Deep Learning revolution that reshaped Computer Vision (CV) and AI in general occurred with the introduction of the ImageNet dataset [26]. Researchers in the fields of computer vision and image processing used small datasets like CalTech101 (9k photos) dataset [18], PASCAL-VOC dataset (30k images) dataset [27], LabelMe (37k images) dataset, and the SUN (131k images) dataset [28] to build image classification models before ImageNet. For a text-to-image generation, it is needed to have a dataset containing an image and its description as a text. However, we need to convert the text into images, the database must contain images along with their captions. Therefore, this section will look at the considered database, Oxford flowers-102, comprises 103 various classes of flowers. Each flower sample is described in detail, many characteristics are described; whether the flower is native, and its texture, the outline of the border formation of petals, the overall formation of the space, and the color pattern [18]. The Oxford flowers-102 dataset [29] has 103 different flower classifications. A thorough description highlighting many aspects of each flower specimen is included, including its natural form and texture, the outline of its edges, the intricate arrangement of its petals, and its color. The database can be found at the following reference [30]. The table 2 below contains examples of the Oxford 102 flower dataset images and their corresponding text (only the first sentence):

Table 2: Samples of images and the first description associated with the image from the dataset [30].

Data	Description
	Outer petals are green in color and larger, inner petals are needle-shaped.
	There are several shapes, sizes, and colors of petals on this complex flower.
	The stamens are towering over the stigma which cannot be seen.
	This flower is white and purple in color, with petals that are oval shaped.

4.2. Experimental Steup

The training of the model utilizes a high-performance computational environment, specifically a Google Colab instance with an NVIDIA T4 Tensor Core GPU. The T4 GPU, based on the Turing architecture, includes 2560 CUDA cores, 320 Tensor cores, and 16 GB of GDDR6 memory, offering peak performance of 8.1 TFLOPS (FP32) and 130 TOPS (INT8). This GPU is optimized for machine learning and AI workloads, providing high throughput for training and inference tasks. The instance also features a high RAM configuration with a capacity exceeding 25 GB, which is crucial for loading large datasets and models, as well as performing intermediate computations during training. The CPU in this setup is a high-performance virtual CPU with multiple vCPUs (typically 2-4 cores), efficiently handling data preprocessing, augmentation, and other CPU-bound tasks. The fast SSD storage ensures quick access to large datasets and model checkpoints, reducing I/O bottlenecks during training. The software stack includes a Linux-based operating system provided by Google Colab, with PyTorch as the deep learning framework, supported by CUDA for GPU acceleration. Additionally, libraries for data handling (pandas, numpy), image processing (PIL), and model-specific modules (transformers, diffusers) are utilized. The experiments were initialized using a pre-trained Stable Diffusion model from the CompVis/stable-diffusion-v1-4 repository. This served as our base model and starting point for fine-tuning. Table 3 lists the Specifications of the Computational Resources Used for Training the Stable Diffusion Model.

Table 3: Specifications of the Computational Resources Used for Training the Stable Diffusion Model

Component	Details
GPU	NVIDIA T4 Tensor Core GPU
Architecture	Turing
CUDA Cores	2560
Tensor Cores	320
Memory	16 GB GDDR6
Peak Performance	8.1 TFLOPS (FP32), 130 TOPS (INT8)
RAM	High RAM configuration (25 GB+)
CPU	High-performance virtual CPU (2-4 vCPUs)
Storage	Fast SSD storage
Operating System	The Linux-based environment provided by Google Colab
Deep Learning Framework	PyTorch with CUDA support for GPU acceleration
Additional Libraries	pandas, numpy, PIL, transformers, diffusers

The training phase involves utilizing the Stable Diffusion model to generate high-quality images based on textual descriptions. This phase employs specific settings and parameters to ensure the accuracy and efficiency of the results. In this experiment, the number of epochs is set to 12, meaning the model will iterate through the entire training dataset 12 times, allowing the model to progressively refine its weights and learn important patterns in the data.

The image resolution is determined as 512×512 pixels, which means that the generated images will have sufficient quality for such purposes as graphic design or fine arts. This decision allows catering to the need for quality imagery while at the same time not requiring significant computational power and memory, the latter of which increases with the resolution. The number of warmup steps is 25 steps. The learning rate schedule is defined in that, in deep learning, there are warmup steps that are aimed at increasing the learning rate from zero to some assigned

value during the initial training. This aids in making training more stable such that large weight updates would cause unstable training. Next, the learning rate is set to $1e-6$ since a small learning rate is preferred to minimize large jumps of values that affect loss in precision with relation to the data. Selection of the learning rate is very important in the training process; a high learning rate could make the training oscillate and be unstable while a small learning rate makes the training slow to converge to the best solution. The batch size is chosen to be 2, which seems to be quite small. The specifications about the batch size have to do with the available calculation potential and the training speed against update accuracy. In this case, it is advisable to set a small batch size as it will ease model stability during training even though the training will be slow. The procedure to train the program involves the use of different libraries such as PyTorch and CUDA which help in the computation and the use of GPU respectively [31], the scheduling methods include DDPMScheduler and DDIMScheduler which helps in gradually transforming noise into image. The training is regulated by the model supervisor and the latter also conducts a check of the generated images for admissibility of release. The testing phase succeeds the training phase which seeks to determine that the model works well and tests it on data it has not been trained on. Hypotheses are made during the test, and the quality of images which are generated from new textual inputs is assessed according to specified metrics such as accuracy, clarity, and relevance to the texts. The values of the resolution and batch size, which were used during training are also used in testing for comparability of results between two phases. In this phase, model weights are not altered. Instead the weights are frozen to evaluate the model based on the knowledge obtained during the training phase. The testing procedure also involves the usage of training tools and libraries with the aim of testing performance indicators and establishing any vulnerable sections. The outcomes are as follows and it is compared with the planned objectives to check the efficiency of the tasks performed by the model. Finally, the model is assessed by the quantity and quality measures and gives an overall understanding of the extent to which the objectives were achieved and how well the model fits a different task. Additional comments on the results obtained and suggestions for improvement of the model in the future or the creation of more efficient modifications of the proposed model are incorporated. The batch size of 2 was selected for several factors, which are connected with computational performance and numerical resources. Making the batch size small can be useful in a circumstance where GPU memory is scarce since large batch sizes consume more memory which might not be present. Further, since a small batch size feeds the model lesser data, the model is updated more frequently and this can have the effect that the updates to the model parameters increase in their precision and hence, there is a reduced chance that the method falls into void solutions that are local optima of the error function. This balance between precision and available resources is a primary reason for choosing a small batch size, ensuring that the model can be trained efficiently without exceeding the memory limits of the device used. Figure 2. shows the Overview of the Stable Diffusion Model Pipeline for Generating Images from Textual Descriptions.

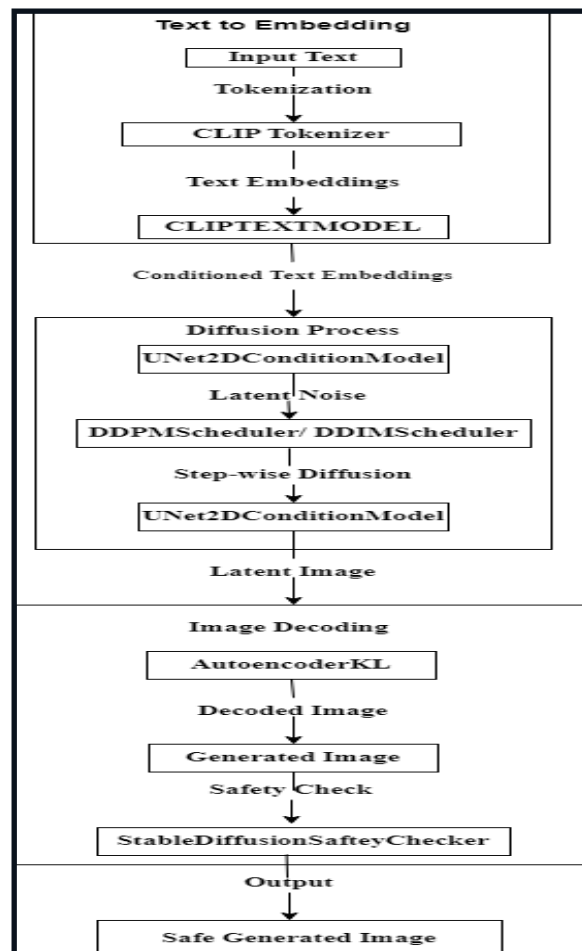








Figure 2: Overview of the Stable Diffusion Model Pipeline for Generating Images from Textual Descriptions

The testing results of the proposed model using the hyper-parameters are explained in table 4.

Table 4. Results of the proposed model

Text	Original image	Generated image
This flower is blue and green, with oval-shaped petals.		
Outer petals are green in color and larger, and inner petals are needle-shaped.		
This flower is blue and green, with oval-shaped petals.		

The above table 4. explains the achievement (generated image) from the given text and the utilized general image dataset using hyper-parameters. the proposed stable diffusion model pipeline for generating images from textual descriptions has generated the flowers based on the training process. In this study, both the encoder and the decoder start with the latent vectors which are comprised of random noise, and the model gradually removes the noise by passing the vectors through the U-net. The final latent vectors are then generated and the VAE is used to decode these latent vectors to get the images. to validate the accuracy of the generated images, two metrics were used in the form of inception score (is) and frechet inception distance (fid). The is assesses the quality of the images to the intended categories within the theme and the variance of the images produced. The fid quantifies how similar the distribution of the created images and the genuine images is in terms of the mean and covariance. These are outcomes that can be quantified and are necessary for the assessment of the proposed model's performance.

4.3. Experiment 1: Fine-Tuning Performance Across Epochs

We evaluated the performance of our fine-tuned model across different epochs. The results are presented in Table 5.

Table 5: Fine-Tuned Stable Diffusion Model Performance across Epochs

Model	IS Mean	IS Std	FID
Base Model	1.5960649	0.016032545	248.748256
Epoch 1	1.6064876	0.021265263	252.899013
Epoch 2	1.6140872	0.019562684	252.355621

Epoch 3	1.6145291	0.018562684	252.133787
Epoch 4	1.6163372	0.018729529	251.981384
Epoch 5	1.6181474	0.018896375	251.828981
Epoch 6	1.6199575	0.01906322	251.676578
Epoch 7	1.6217676	0.019230066	251.524175
Epoch 8	1.6235778	0.019396911	251.371772
Epoch 9	1.6244419	0.019563757	251.219369

A. Key Observations Of Experiment 1:

1. The Inception Score (IS) showed a general trend of improvement across epochs, with the highest mean value of 1.6244419 achieved at epoch 9.
2. The FID scores for the fine-tuned models were consistently higher than the base model, indicating that the fine-tuned models may have diverged slightly from the original distribution of real images.
3. The base model achieved the lowest FID score of 248.748256, suggesting it maintained the closest similarity to real images in terms of overall distribution.

4.4. Experiment 2: Comparison Of Fine-Tuned Model With Base Model

In this section we compared the performance of our fine-tuned model (labeled as "Stable diffusion finetune") with the base model (labeled as "Stable diffusion V4"). The results are presented in Table 6.

Table 6 Comparison of Fine-Tuned and Base Models

Model	IS Mean	IS Std	FID
Stable diffusion V4	1.61	0.02	251.22
Stable diffusion finetune	1.60	0.04	212.52

B. KEY OBSERVATIONS FOR EXPERIMENT 2:

1. The Inception Scores (IS) for both models were comparable, with the base model (Stable diffusion V4) slightly outperforming the fine-tuned model.
2. The FID score of the fine-tuned model got down to 212.52, which was significantly better as compared to the FID score of the base model (251.22), testifying that the images produced by the fine-tuned model were closer to the real images in the training set. Nevertheless, the finetuning of the model resulted in a higher standard deviation connected to the quality and the variety of the generated IS, which means that it was more variable.

4.5. Research Limitation

Despite the improvements observed in the Inception Score (IS) over multiple epochs, there are several limitations to our approach.

1. FID Score Divergence: The FID scores for the models fine-tuned in SigOpt were higher than the base model FID scores meaning that the models have deviated from the distribution of real images. This means that although the model can create sensible images, they might not be very realistic and hence the applicability in realistic image distribution scenarios might be harmed.

2. Small Batch Size: In this case, because of memory issues, the batch size was set to 2, which may have affected the convergence speed and the model's capability of generalizing from the training data. The mtDNA was unable to significantly increase the batch size for training which normally aids in stabilizing training and increasing the generality of the model.

3. Limited Training Epochs: The number of training epochs was limited to 12; though the model did not get enough epochs to converge and achieve the optimum results. Increasing the number of epochs can also improve the performance rates received by the NN.

4. Resource Constraints: The training process was carried out on an NVIDIA T4 GPU with a high RAM configuration; however, such a configuration has restrictions regarding the maximal model size and batch sizes. The availability of a higher level of hardware might allow one to try more extensive tests and, thus, achieve improved outcomes.

5. Evaluation Metrics: The use of IS and FID as the evaluation criteria can be considered conventional, and though effective, it does not address the qualitative nature of the generated images as much as it should. Further studies may include extra measures or human evaluations to conduct a broader outlook toward the model's performance.

5. Conclusion

The study unveils that the stable Diffusion model was improved by fine-tuning it showing that the fine-tuned model proved a considerably lower FID score (212.52) when contrasted with the base model (251.22), pointing out that the generated images from the fine-tuned model were more intimate to the target distribution in the synthetic dataset. However, there are trade-offs between image quality and flexibility in performance metrics. The fine-tuning process gradually improves the model's global ability to produce better and more diverse digital imagery. Proper tuning of hyper-parameters is essential in the fine-tuning stage.

The fine-tuned model has a lower FID score (248.748256), suggesting a higher likelihood of attaining higher image distribution similarity to the targeted dataset. This approach is particularly useful for cases requesting similarity to a specific set of real pictures. Future research should aim to fine-tune the model to achieve higher Inception Scores or equal FID scores while achieving lower FID scores. Refinements of the hyper-parameter optimization algorithm can contribute to further enhancement.

Future fine-tuning research should focus on enhancing or maintaining Inception Scores while achieving low FID scores and testing more complex hyper-parameter optimization methods. This model could be used for video generation from text due to its robustness.

References

-
- [1] M. Żelazczyk and J. Mańdziuk, "Text-to-Image Cross-Modal Generation: A Systematic Review," *arXiv Prepr. arXiv2401.11631*, 2024.
 - [2] S. Frolov, T. Hinz, F. Raue, J. Hees, and A. Dengel, "Adversarial text-to-image synthesis: A review," *Neural Networks*, vol. 144, pp. 187–209, 2021.
 - [3] M. M. Hashim, H. J. Alhamdane, A. H. Herez, and M. S. Taha, "Based on Competitive Marketing: A New Framework mechanism in Social Media," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, p. 12121.
 - [4] T. Sousa, J. Corre-ia, V. Pereira, and M. Rocha, "Generative deep learning for targeted compound design," *J. Chem. Inf. Model.*, vol. 61, no. 11, pp. 5343–5361, 2021.
 - [5] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International conference on machine learning*, PMLR, 2016, pp. 1060–1069.
 - [6] M. Dubova, "Building human-like communicative intelligence: A grounded perspective," *Cogn. Syst. Res.*, vol. 72, pp. 63–79, 2022.
 - [7] N. S. Ali, A. F. Alsafo, H. D. Ali, and M. S. Taha, "An Effective Face Detection and Recognition Model Based on Improved YOLO v3 and VGG 16 Networks," *J. homepage <http://iicta.org/journals/ijcmem>*, vol. 12, no. 2, pp. 107–119, 2024.
 - [8] M. Elasri, O. Elharrouss, S. Al-Maadeed, and H. Tairi, "Image generation: A review," *Neural Process. Lett.*, vol. 54, no. 5, pp. 4609–4646, 2022.
 - [9] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.
 - [10] C. Bodnar, "Text to image synthesis using generative adversarial networks," *arXiv Prepr. arXiv1805.00676*, 2018.
 - [11] M. Otani *et al.*, "Toward verifiable and reproducible human evaluation for text-to-image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14277–14286.
 - [12] R. Gal *et al.*, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv Prepr. arXiv2208.01618*, 2022.
 - [13] Z. Jin, X. Shen, B. Li, and X. Xue, "Training-free diffusion model adaptation for variable-sized text-to-image synthesis," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

-
- [14] S. Ramzan, M. M. Iqbal, and T. Kalsum, "Text-to-Image generation using deep learning," *Eng. Proc.*, vol. 20, no. 1, p. 16, 2022.
- [15] H. Tibebu, A. Malik, and V. De Silva, "Text to image synthesis using stacked conditional variational autoencoders and conditional generative adversarial networks," in *Science and Information Conference*, Springer, 2022, pp. 560–580.
- [16] K. D. Kumar, S. Srang, and D. Vally, "A Review of Generative Adversarial Networks (GANs) for Technology-Assisted Learning: Solving Teaching and Learning Challenges," in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, IEEE, 2022, pp. 820–826.
- [17] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 8780–8794, 2021.
- [18] A. Singh and S. Agrawal, "CanvasGAN: A simple baseline for text to image generation by incrementally patching a canvas," *arXiv Prepr. arXiv1810.02833*, 2018.
- [19] X. Ouyang, X. Zhang, D. Ma, and G. Agam, "Generating image sequence from description with LSTM conditional GAN," in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 2456–2461.
- [20] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Learn, imagine and create: Text-to-image generation from prior knowledge," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [21] A. A.-A. Hadad, H. N. Khalid, Z. S. Naser, and M. S. Taha, "A Robust Color Image Watermarking Scheme Based on Discrete Wavelet Transform Domain and Discrete Slantlet Transform Technique," *Ing. des Syst. d'Information*, vol. 27, no. 2, p. 313, 2022.
- [22] G. Kim, T. Kwon, and J. C. Ye, "Diffusionclip: Text-guided diffusion models for robust image manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2426–2435.
- [23] S. Gu *et al.*, "Vector quantized diffusion model for text-to-image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10696–10706.
- [24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, IEEE, 2009, pp. 248–255.
- [27] S. Vicente, J. Carreira, L. Agapito, and J. Batista, "Reconstructing pascal voc," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 41–48.
- [28] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "Sun database: Exploring a large collection of scene categories," *Int. J. Comput. Vis.*, vol. 119, pp. 3–22, 2016.
- [29] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3558–3568.
- [30] M.-E. Nilsback, "An automatic visual flora-segmentation and classification of flower images." Oxford University, 2009.
- [31] Z. S. Naser, H. N. Khalid, A. S. Ahmed, M. S. Taha, and M. M. Hashim, "Artificial Neural Network-Based Fingerprint Classification and Recognition," *Rev. d'Intelligence Artif.*, vol. 37, no. 1, 2023.