Project Report on

STABLE DIFFUSION OF TEXT-TO-IMAGE

Submitted by

RAJITHA VARDHI-R170946

Under the guidance of

M.HIMABINDU

Assistant Professor

Department of Computer Science and Engineering

RK VALLEY

Rajiv Gandhi University of Knowledge and Technologies (RGUKT),R.K.Valley ,Kadapa ,Andhra Pradesh.

DECLARATION

We, hereby declare that this report entitled “STABLE DIFFUSION OF TEXT-TO-IMAGE” submitted by us under the guidance and supervision of M.Himabindhu is a bonafide work . We also declare that it has not been submitted previously in part or in full to this university or other university or institution for the award of any degree or diploma. All information included from other sources have been duly acknowledged.

We will be solely responsible if any kind of plagiarism is found.

Place : R.K.Valley V.RAJITHA(R170946)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our Director.Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.N.SATYANANDARAM for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Ms.M.HIMABINDU for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

ABSTRACT

The Stable Diffuison of Text-to-image project aims to generate high-quality images based on textual descriptions.The approach relies on a stable diffusion process, which is a model that uses attention mechanisms and discrete diffusion steps to control the diffusion of information.The project uses a pre-trained language model to generate textual descriptions, which are then used as input to a pre-trained image generation model. The image generation model is fine-tuned on a large dataset of images to generate high-quality images that match the input textual description.The project uses several schedulers, such as PNDMScheduler, DDIMScheduler, LMSDiscreteScheduler, EulerDiscreteScheduler, and DPMSolverMultistepScheduler, to control the diffusion process and improve the quality of generated images. The project also uses several packages, such as transformers, accelerate, mediapy, triton, ftfy, spacy and xformers, to enhance the performance and capabilities of image generation process. The generated images can be used for various applicatons, such as content creation, advertising, and art.

INDEX

1.INTRODUCTION

Stable Diffusion is a state-of-the-art image generation technique that uses a generative model to generate high-quality images based on a given text prompt. The technique is based on the idea of iteratively refining a noisy image to gradually improve its quality until a final high-quality image is produced.

Stable Diffusion has been shown to be highly effective for generating photorealistic images from textual descriptions, and has many potential applications in fields such as art, design, and advertising. It is also being used in the development of advanced virtual and augmented reality applications.

The Stable Diffusion technique is implemented the '**diffusers'** Python library, which provides a convenient way to use pre-trained models and generate images from text prompts. The '**StableDiffusionPipeline'** is a key component of this library, providing a high-level interface for generating images uing Stable Diffusion technique. By providing a text prompt and other parameters such as the image size and number of inference steps, the '**StableDiffusionPipeline'** can generate high-quality images that match the given prompt

2.Purpose

The purpose of stable diffusion of text to image is to generate high-quality, photorealistic images based on a given textual description. This technique is useful in a variety of contexts, including art, design, advertising, and entertainment.

Stable diffusion of text to image can be used to create images for marketing campaigns, product advertisements, and

other promotional materials. It can also be used to generate artwork, book covers, and illustrations for novels and other publications. In the entertainment industry, stable diffusion can be used to create visual effects for films, television shows, and video games.

The technique is also being used in the development of virtual and augmented reality applications. By generating realistic images from text prompts, stable diffusion can help to create immersive and interactive virtual environments that are more engaging and lifelike. Overall, stable diffusion of text to image is a powerful and versatile technique that has many potential applications in a wide range of fields.

3.Scope

The scope of stable diffusion of text to image is quite broad and can be applied in many different contexts. Here are a few examples:

- **Art and design:** Stable diffusion can be used to create unique and original artwork, illustrations, and designs based on textual descriptions.

- **Advertising and marketing:** Stable diffusion can be used to create promotional materials, such as advertisements and product images, that are more engaging and visually appealing.

- **Entertainment:** Stable diffusion can be used to create visual effects for films, television shows, and video games, enhancing the overall visual experience for viewers and players.

➔ **Publishing:** Stable diffusion can be used to generate book covers and illustrations for novels and other publications, helping to capture the essence of the story and engage readers.

➔ **Virtual and augmented reality:** Stable diffusion can be used to create realistic and immersive virtual environments, which can be used for training, education, or entertainment purposes.

4.Requirement Specification

Hardware Configuration:

- **GPU:** A powerful GPU is essential for stable diffusion, as it requires a significant amount of computational power. A GPU with at least 8 GB of VRAM is recommended, but a GPU with 16 GB or more is ideal for generating high-quality images.
- **CPU:** A multi-core CPU with a clock speed of at least 3 GHz is recommended.
- **RAM:** A minimum of 16 GB of RAM is recommended, but 32 GB or more is ideal for generating high-quality images.
- **Storage:** Stable diffusion requires a significant amount of storage space, as it generates large image files. A fast SSD with at least 500 GB of storage is recommended.

Software Configuration:

Programming Language	Python 3.0
Packages	Diffusers, transformers, xformers, mediapy, pytorch, accelerate, triton
Operating System	Windows, Linux, macOS
Technology	Machine Learning

Python:

Python is a popular programming language for machine learning due to its simplicity, flexibility, and rich ecosystem of libraries and tools specifically designed for machine learning. Python has a large and active community of developers and users, which has contributed to the development of many powerful and user-friendly machine learning frameworks and libraries.

Packages:

Some of the most popular machine learning libraries in Python include:

1. **Diffusers:** The ‘diffusers’ library is a Python package for generating images from textual prompts using a technique called stable diffusion. It is built on top of PyTorch and provides a simple API for generating images from a given text prompt. The library

includes pre-trained models that have been trained on large datasets of images and textual prompts, and can generate high-quality images that are semantically related to the input prompt.

Scheduler: 'scheduler' is an object that defines the schedule for gradually increasing the diffusion timesteps during the generation of an image.Diffusion timesteps determine the level of noise in the generated image, and gradually increase them can help to produce smootheer, more realistic images.

Here are some schedulers:

- PNDM Scheduler: PNDM Scheduler is a diffusion-based scheduler provided by the diffusers Python library. PNDM stands for Progressive Noisy Diffusion-based Method, which is a diffusion-based method that adds progressively more noise to the image at each step of the diffusion process.

- DDIM Scheduler: DDIM stands for Differentiable Diffusion-based Image Manipulation, which is a diffusion-based method that allows for the manipulation of images during diffusion process. The scheduler determines the amount of nise to be added to the image at each iteration and also takes in a control signal that can be used to manipulate the image during the diffusion process.

- LMSDiscrete Scheduler: The LMS is the name stands for the Leaky Memory Solver, which is an algorithm used to solve the diffusion equation that governs the diffusion process.The LMSDiscreteScheduler works by discretizing the diffusion equation and solving it using a numerical solver.

- EulerDiscrete Scheduler: The Euler in the name refers to the Euler method,which is a numerical method for solving ordinary differential equations.

- DPMSolverMultistepScheduler: DPMSolverMultistepScheduler is a scheduler in the stable diffusion framework that uses a multistep method to solve the diffision process. It is based on the DPM(Discretized Progressive Method) algorithm which computes the diffusion process as a sequence of small discrete steps.

- StableDiffusionPipeline: StableDiffusionPipeline is a class provided by the diffusers library, which is designed for generating high-quality images from textual prompts using diffusion models. The pipeline takes a prompt as input and produces one or more images as output. The pipeline has several configurable parameters that can be adjusted to control the quality and diversity of the generated images.

2.**Transformers:** The 'transformers' library is a python package for natural language processing (NLP) tasks, such as text classification, question answering, and language generation. The library includes pre-trained models that have been trained on large datasets of text, and can be fine-tuned on custom datasets for specific tasks. It also includes tools for tokenization, data preprocessing, and model evaluation, making it a comprehensive toolkit for NLP tasks.

3.**Xformers:** xformers is a package that provides PyTorch implementations of various memory-efficient self attention mechanism.These techniques can significantly reduce the memory usage of self attention, making it possible to apply self -attention to

longer sequences, ehich is particularly useful in natural language processing(NLP) and image generation tasks.

4.**Accelerate:** Accelerate is a python library designed to provide easy-to-use, high-level abstractions for parallel computing on CPU and GPUs. The main advantage of Accelerate are faster training times and improved scalability for deep learning models.

5.**mediapy: ‘**mediapy’ is a python library that provides a simple interface for displaying images, videos, and audio files in Jupyter notebooks or as standalone applications.It is built on top of popular media processing libraries such as OpenCV and Ffmpeg, anf allows for easy manipulation and visualizing of media files.

6.**Pytorch:** PyTorch is an open source machine learning framework that is used for developing and training deep learning models. PyTorch provides easy to use APIs for implementing various machine learning algorithms and also supports dynamic computation graphs which makes it easy to debug and optimize models. It is highly optimized for performance and can run on multiple hardware platforms such as GPUs and TPUs.

7.**Triton:** Triton is an open-source project by NVIDIA for building and deploying machine learning models in production. It provides a flexible and scalable platform to deploy models using different machine learning frameworks, including TensorFlow, PyTorch. With Triton, you can deploy your models on a variety of hardware platforms, including CPU, GPUs, and TPUs.

In summary, Python is an excellent choice for developing machine learning models due to its simplicity, flexibility, and rich ecosystem of libraries and tools. It provides an easy-to-learn syntax and offers powerful and efficient libraries for building and training machine learning models.

Machine Learning:

Machine learning is a subfield of artificial intelligence that involves the development of algorithms and statistical models that enable computer systems to learn from and make predictions or decisions based on data. The goal of machine learning is to build systems that can automatically improve their performance on a given task over time, without being explicitly programmed.

The basic idea behind machine learning is to train a model using a set of input data and corresponding output data, known as a training set. The model then uses this training data to learn patterns or relationships in the data that can be used to make predictions or decisions on new, unseen data.

Machine learning has a wide range of applications, including image recognition, natural language processing, speech recognition, recommendation systems, fraud detection, and many others. Machine learning is becoming increasingly important in various industries, including healthcare, finance, transportation, and e-commerce.

5.Analysis and Design

Stable Diffusion is a powerful techinque that can be used to generate high-quality images from textual prompts. The Stable Diffusion model can be used to generate a wide variety of images, from photorealistic to abstract. It can also generate multiplr images from a single prompt, allowing users to explore a wide range of visual possibilities. The model is capable of generating high-resolution images, up tp 768x768 pixels.

The stable Diffusion pipeline consists of three main components:the scheduler, the diffusion model, and the image decoder. The Sheduler is responsible for controlling the diffusion process, which involves iteratively adding noise to the image and then

removing it to reveal a new image. The diffusion model is responsible for generating the image from the textual prompt, and the image decoder is responsible for converting the generated image into a usable format.

The stable Diffusion model is based on a combination of transformer networks and diffusion processes. The transformer network is used to encode the textual prompt, and the diffusion process is used to generate the image from the encoded prompt. The diffusion process involves iteratively adding noise to the image and then removing it, using a series of diffusion steps. This process helps to ensure that the generated image is of high quality and is consistent with the textual prompt.

In conclusion, the Stable Diffusion model is a powerful tool for generating high-quality images from textual prompts. It combines the power of transformer networks and diffusion processes to generate images that are consistent with the textual prompt and of high quality. Thw model is highly flexible and can be used for a wide range of apllications, making it a valuable tool for artists, researchers, and others who nees to generate high quality images from textual prompts.

Design:

The design of stable diffusion of text to image involves the following steps:

1.Preprocessing the text prompt: The input text prompt is first preprocessed to remove any unwanted characters and normalize the text.

2.Generating multiple images: The stable diffusion model generates multiple images for a given text prompt. The number of images to generate is configurable, and it affects the quality of the generated images.

3. Configuring the model parameters: The model parameters, such as the diffusion steps, guidance scale, and the image size, are configured based on the requirements of the user.

4.Fine-tuning the model: The stable diffusion model is fine-tuned on a large dataset of images to generate high-quality images for a given text prompt.
5.Selecting a scheduler: The stable diffusion model uses a scheduler to control the diffusion process. There are different schedulers available, such as PNDMScheduler, DDIMScheduler,LMSDiscreteScheduler,EulerDiscreteSch-eduler, and DPMSolverMultistepScheduler. The scheduler can be selected based on he requirements of the user.
6.Generating the images: The stable diffusion model generates the images for a given text prompt using the selected scheduler and model parameters.
7.Post-processing the images: The generated images are post-processed to remove any artifacts and improve the visual quality.
8. Displaying the images: The generated images are displayed to the user for evaluation and further processing, such as saving or sharing.

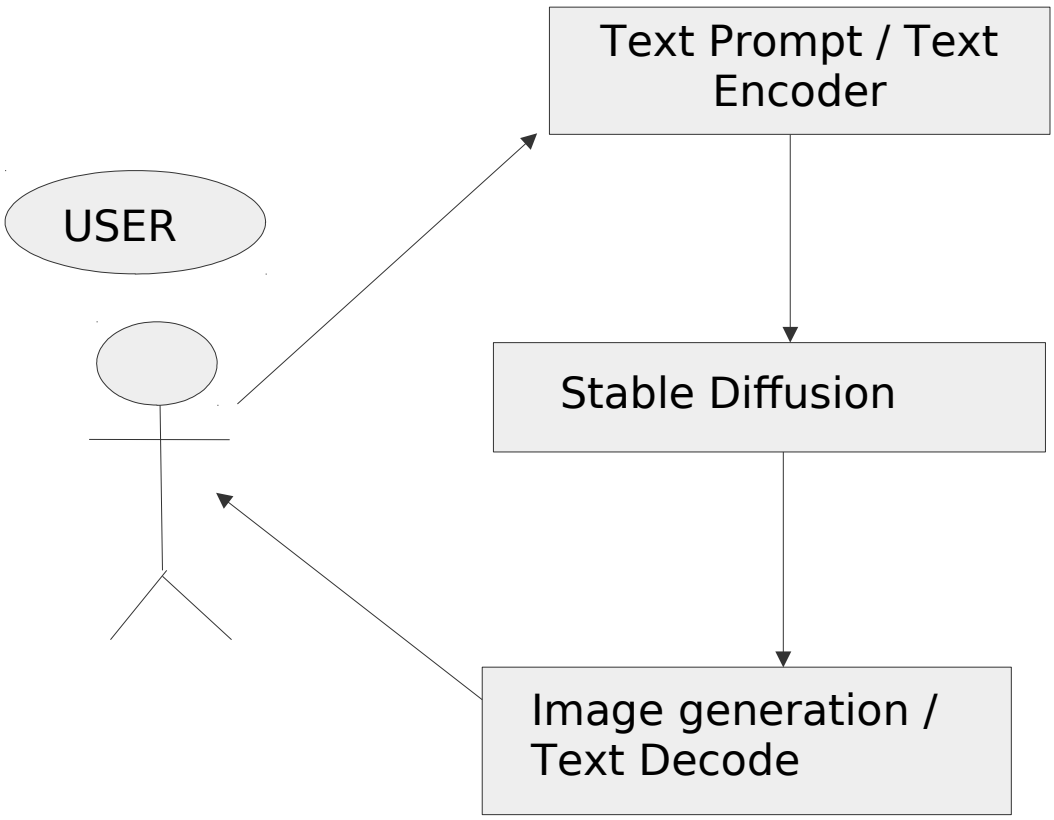
UML Diagrams:

The Unified Modelling Language (UML) is a graphical language for visualizing, specifying, constructing and documenting of a software intensive system. The UML gives a standard way to write a system blueprints, covering conceptual things, such as classes written in a specified programmed language, database schemas and reusable software components.

- Use-Case Diagram
- Activity Diagram

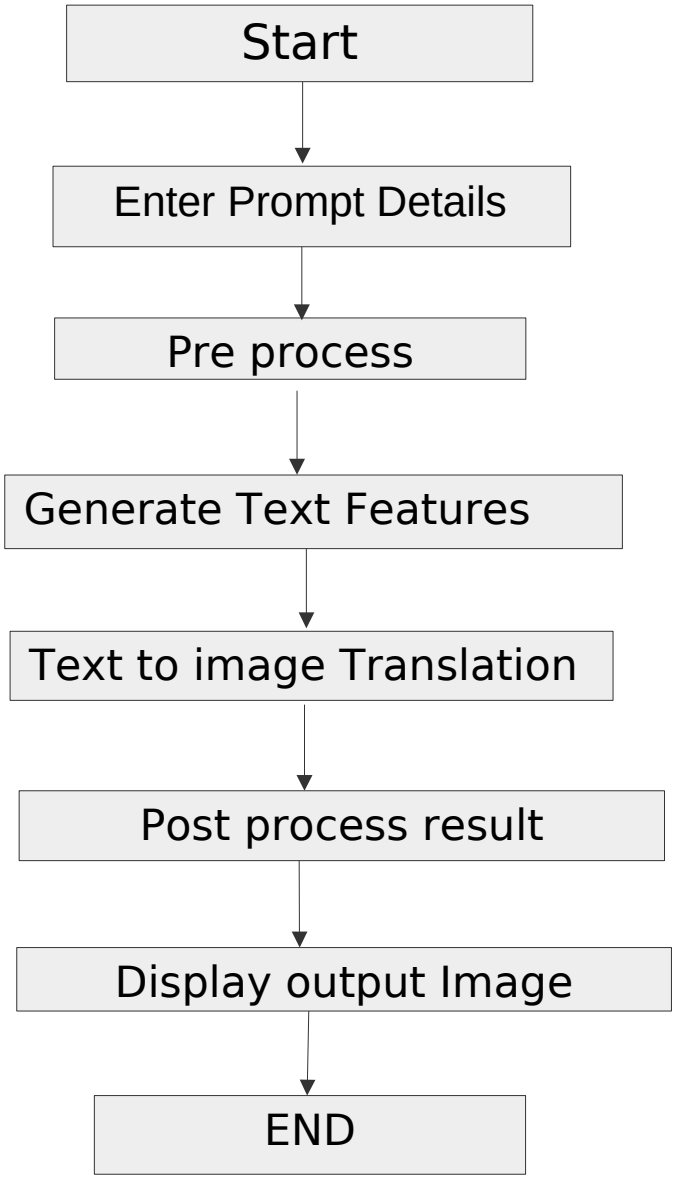
Use case Diagram:

The use case diagram shows the primary actors and their interactions with the Text-to image stable diffusion. The t user gives text prompt to the system. From pre-trained dataset using stable diffusion process, text diffuses to image based on the prompt . Once image is decoded from the dataset it gives output the image to user



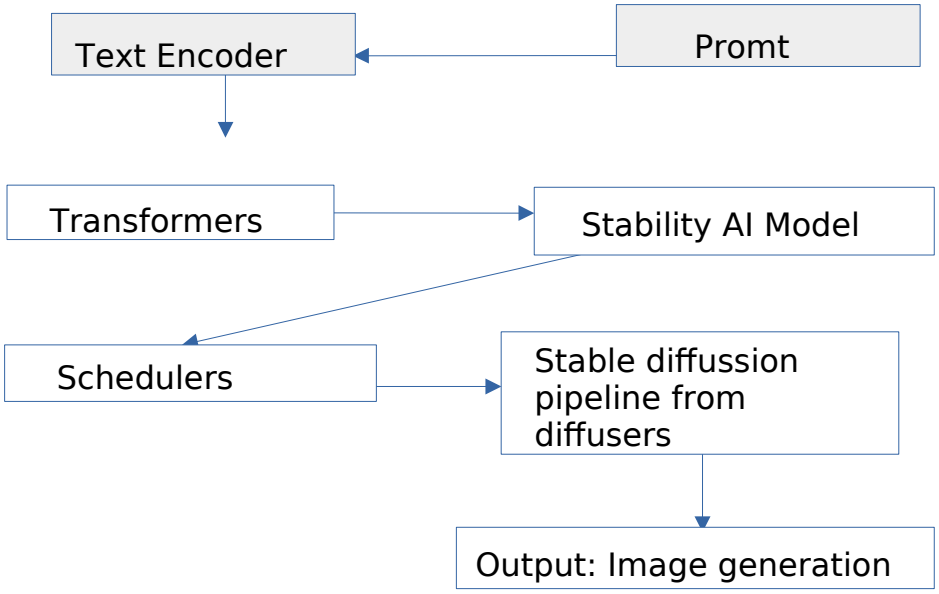
Activity Diagram:

An activity diagram is a type of UML (Unified Modeling Language) diagram that models the o flow of activities or actions within a system or process. Activity diagrams are commonly used in software design and business process modeling to describe the steps and interactions involved in a particular process or use case.



Deployment Diagram:

A deployment diagram in UML is a type of diagram that shows the physical deployment of components in a system.It is used to illustrate how the software components of a system are deployed onto hardware components such as servers, processors, or other devices.



6.Implementation and Testing

After the model have been perfectly created with good accuracy, the system will be implemented and the system can be used.

System Testing

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2 .Integration testing

Unit Testing:

Unit testing involves testing for stable diffusion of text to image involves testing the functionality of the code at the individual function or method level. In this case, we can write unit tests for each function or method used in the code to ensure that it is performing as expected.

1.Testing the functionality of the StableDiffusionPipeline module from the diffusers library by passing in different model Ids and checking the output.

2.Testing the functionality of the transformers library by creating a custom tokenizer and checking if it correctly tokenizes a given prompt.

3.Testing the functionality of the xformers library by creating a custom self-attention layer and checking if it correctly computes the attention scores for a given input.

4. Testing the functionality of the image generation pipeline by passing in different text prompts and checking if the generated images are visually coherent and match the input prompt.

5. Testing the performance of the system by measuring the time it takes to generate images for a given text prompt and comparing it to the expected runtime.

Integration Testing:

Integration testing involves testing the entire system as a whole to ensure that all of the components work together correctly. This could involve testing the integration of different modules, the accuracy of the predictions made by the software, and the overall user experience of the system. Integration testing is done manually and can help us identify any issues that may arise when different components of the software are combined.

Some possible integration test cases for stable diffusion of text to image could include:

1. Testing the integration between the DiffusionPipeline and the Transformers library by verifying that the pipeline is able to correctly load and use a pre-trained language model.

2. Testing the integration between the DiffusionPipeline and the Xformers library by verifying that the pipeline is able to correctly use the memory-efficient attention mechanism provided by Xformers.

3. Testing the end-to-end functionality of the system by providing various prompts and verifying that the pipeline is able to generate coherent and realistic images that correspond to the prompts.

4. Testing the robustness of the system by providing various input data types and sizes, and verifying that the pipeline is able to handle them without crashing or producing incorrect results.

5. Testing the scalability of the system by measuring its ability to generate images in parallel on multiple GPUs or on a distributed computing cluster.

7. CODE:

```
%pip install --quiet --upgrade diffusers transformers accelerate mediapy
triton scapy ftfy spacy==3.4.4

# The xformers package is mandatory to be able to create several 768x768
images.

%pip install -q xformers==0.0.16rc425

# model_id = "stabilityai/stable-diffusion-2-1-base"

# model_id = "stabilityai/stable-diffusion-2-1"
model_id = "dreamlike-art/dreamlike-photoreal-2.0"
from diffusers import PNDMScheduler, DDIMScheduler, LMSDiscreteScheduler,
EulerDiscreteScheduler, DPMSolverMultistepScheduler

scheduler = None
# scheduler=PNDMScheduler.from_pretrained(model_id,subfolder="scheduler")
# scheduler=DDIMScheduler.from_pretrained(model_id,subfolder="scheduler")
import mediapy as media

import torch

from diffusers import StableDiffusionPipeline
device = "cuda"
```

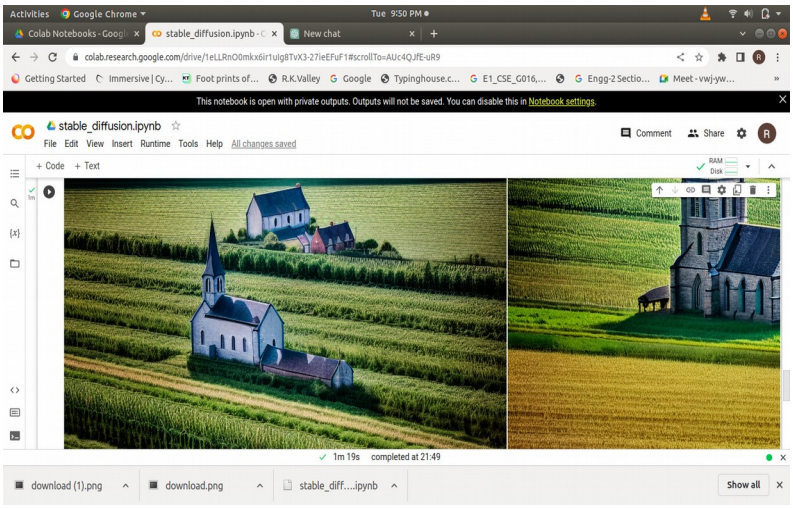
```
if model_id.startswith("stabilityai/"):
      model_revision = "fp16"
else:
      model_revision = None
if scheduler is None:
      pipe = StableDiffusionPipeline.from_pretrained(
      model_id,
      torch_dtype=torch.float16,
      revision=model_revision,
      )
else:
      pipe = StableDiffusionPipeline.from_pretrained(
      model_id,
      scheduler=scheduler,
      torch_dtype=torch.float16,
      revision=model_revision,
      )
pipe = pipe.to(device)
pipe.enable_xformers_memory_efficient_attention()
if model_id.endswith('-base'):
      image_length = 512
else:
      image_length = 768
prompt = "a photo of church in the middle of the crop field "
remove_safety = False
num_images = 4
if remove_safety:
      negative_prompt = None
else:
      negative_prompt = "nude, naked"
images = pipe(
          prompt,
          height = image_length,
          width = image_length,
          num_inference_steps = 25,
          guidance_scale = 9,
          num_images_per_prompt = num_images,
          negative_prompt = negative_prompt,
        ).images
media.show_images(images)
images[0].save("output.jpg")
```

8.Evalution:

Input:

```
prompt = "a photo of church in the middle of the crop field ".
```

Output:



9. Conclusion:

In conclusion,stable diffusion of text to image is a powerful technique that enables the generation of high-quality images from textual descriptions. It relies on advanced deep learning models and complex algorithms to convert the input text into an image that closely matches the description. Thw technique has numerous applications in various fields, including art, design and marketing. However, the process is resource-intensive and requires high-end hardware to achieve optimal performance. Additionally, the technique may face ethical concerns, such as generating inappropriate or baised images based on the input text. Therefore, it is crucial to apply this technique responsibly and ethically. Overall, stable diffusion of text to image has the potential to revolutionize the way we create and consume visual content, and it will be exciting to see how this technique evolves in the future.

10. REFRENCES:

- Pre-trained Dataset and Model from hugging face
  https://huggingface.co/spaces/stabilityai/stable-diffusion
  https://huggingface.co/dreamlike-art/dreamlike-photoreal-2.0

- An arXiv on Multi-Concept Customization of Text-to-Image Diffusion
  https://arxiv.org/abs/2212.04488

**********THANKYOU**********