



Tribhuvan University

Institute of Science and Technology

TEXT-TO-IMAGE GENERATOR

A Final Year Project Submission in

Partial Fulfillment of the Requirement for the Degree of

Bachelor of Science in Computer Science and Information Technology

Under the Supervision of

Mr. Avishek Kuinkel

Submitted by:

Neha Shrestha (24287 / 7th Semester / 2076)

Norden Ghising Tamang (24290 / 7th Semester / 2076)

Submitted to:

TRINITY INTERNATIONAL COLLEGE

Department of Computer Science and Information Technology

Dillibazar Height, Kathmandu, Nepal

March, 2024

4.2 Algorithm Details

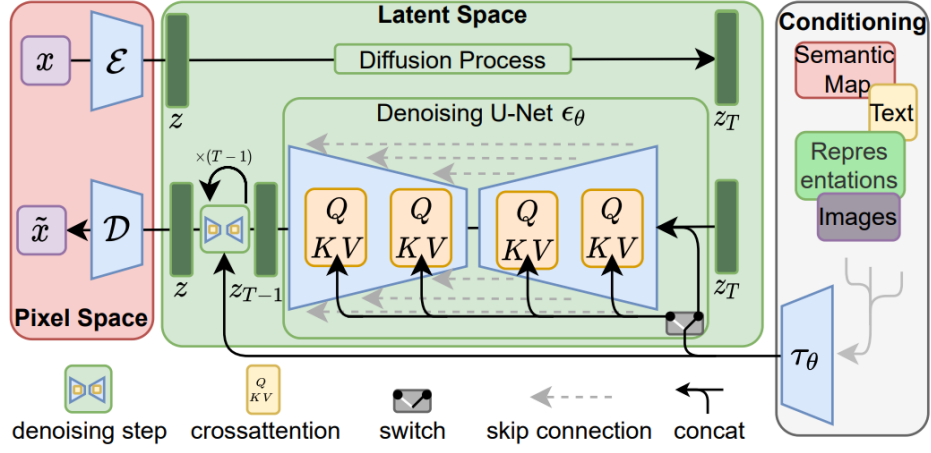


Figure 6: Latent Diffusion Architecture [1]

Figure 6 is the architecture of the Latent Diffusion Model which we are going to implement in our project. The components which make up this architecture are listed below:

i. Auto-encoder:

An auto-encoder is a neural network used for data compression and reconstruction, consisting of an encoder and a decoder. It compresses input into latent space, reconstructs it and aids in training diffusion models on latent space. As proposed in the paper, KL regularized VAE will be used in order to encode the images into latent images during training process and likewise decode the latent images into images for both training and sampling process.

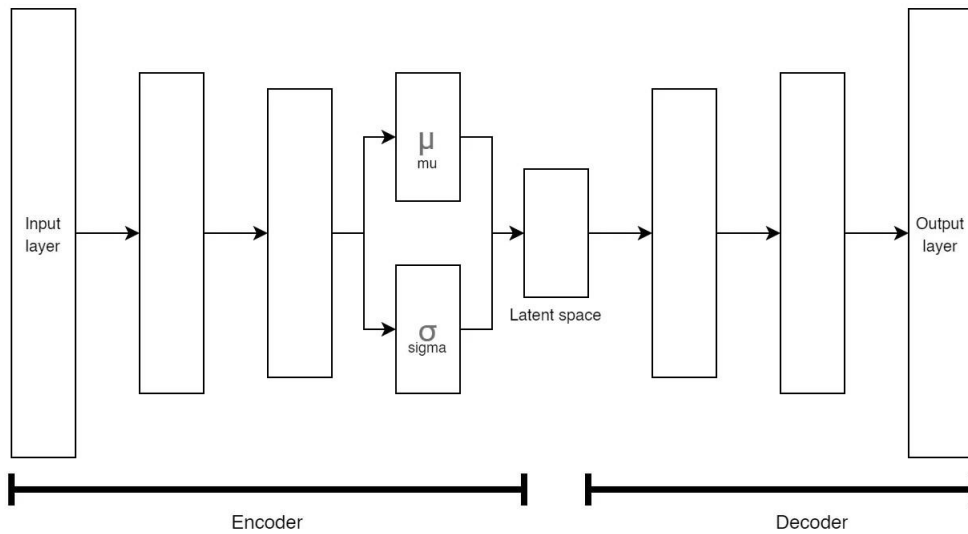


Figure 7: VAE Architecture

x. `ldm.ipynb`

This Jupyter notebook is used for training purpose. Following are the tasks carried out by the notebook.

- Instantiates Unet, VAE and DDPMscheduler.
- Utilizes the functionality provided by “dataset.py” and “data_preprocessing.py” for data preprocessing.
- Executes a standard PyTorch training loop for 1000 epochs, saving the trained model.
- Plots the loss during training.
- Conducts inference using the trained model, demonstrating the generative capabilities of the text-to-image model.

5.2 Testing

5.2.1 Unit Testing:

1. VAE Test:

- Description: Verify that the encoder module correctly converts image into latent and decoder does the vice versa.
- Test Steps:
 - Provide an input image to the encoder or latent to decoder.
 - Compare the output latent generated by the encoder with expected latent or output image generated by the decoder with expected image.
- Expected Outcome: The encoder produces latent that accurately represent the pixel space input image.

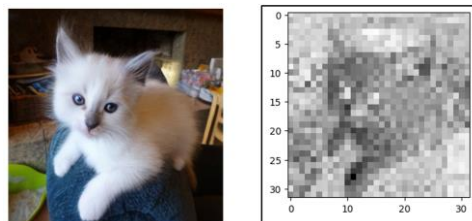


Figure 16: VAE Test Case. Left (Image), Right(Latent).

2. U-Net Test:

- Description: Verify that the UNet correctly predicts the noise mask in an image.
- Test Steps:
 - Provide an input noisy image to UNet.

- Compare the segmented noise from the image with actual noise.
- Expected Outcome: The UNet provide proper noise segmentation and predicts noise.

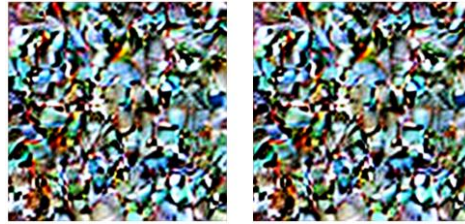


Figure 17: U-Net Noise Prediction Test. Left (Actual), Right (Predicted).

5.2.2 System Testing:

- Description: Verify that the system can generate images from textual prompts seamlessly.
- Test Steps:
 - Provide an input text-prompt and relevant output generation.
 - Compare the output generated with the provided text prompt.
- Expected Outcome: The system consistently produces images that accurately represent the content specified by the textual inputs.



Figure 18: System Test – I

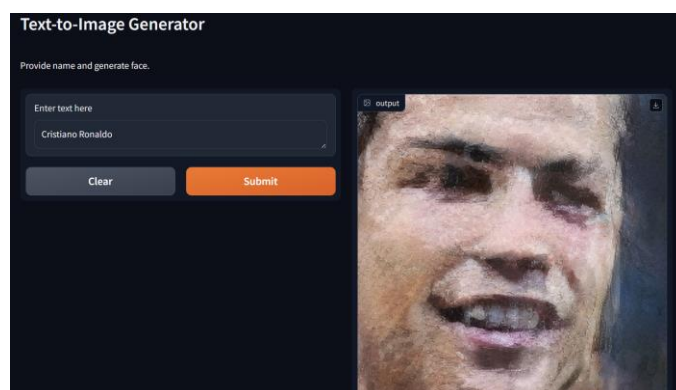
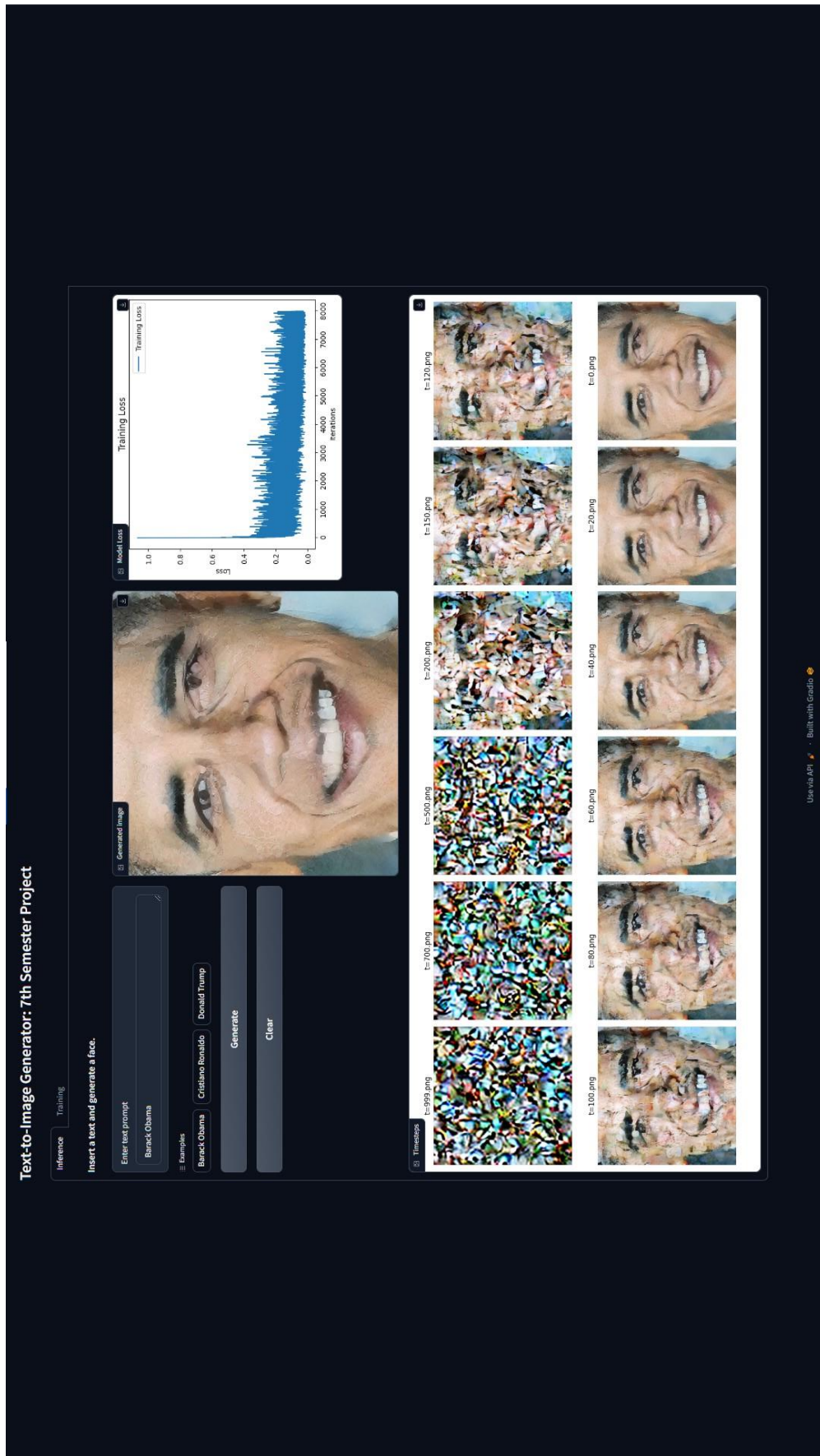


Figure 19: System Test - II

APPENDIX-II





Text-to-Image Generator: 7th Semester Project

Inference Training Scratch Results

Train a Text-to-Image Generator by inserting following Hyperparameters.

Epochs	Batch Size	Learning Rate
1000	20	0.001
Loss Function	Optimizer	Latent Folder Location
<input checked="" type="radio"/> MSE <input type="radio"/> MAE	<input checked="" type="radio"/> Adam <input type="radio"/> SGD	./data/face/images
Preprocessing	Data Folder Location	
<input type="radio"/> Yes <input checked="" type="radio"/> No		
Train		Cancel
Textbox		

Use via API  · Built with Gradio 

Text-to-Image Generator: 7th Semester Project

Inference Training Scratch Results

Results Generated by Scratch Model.

