

# LAB #7

## Prompt Engineering (Stable Diffusion) (10 pts)

**Due by Oct 16<sup>th</sup> Mon 4pm**

Stable Diffusion is a generative AI tool that lets you create or edit various different types of images. There are text to image or image to image methods.

This lab provides hands-on experience using Stable Diffusion and explores the concept and hands-on practices of prompt engineering.

For this and other Stable Diffusion labs, we need to have Python to 3.10 version and install a few extensions (eg. Controlnet and Roop).

### [Setting up for the Stable Diffusion Labs]

The following steps in paperspace.com site will upgrade Python to 3.10 version, install the extensions and UI:

1. Log into Paperspace.com
2. Goto notebooks
3. Select a free GPU & 6hr (or less)
4. Turn on the 'Advanced options'.
5. Enter the following in 'Container Name':  
**cyberes/gradient-base-py3.10:latest**
6. Click 'Start Notebook'
7. Upload the 'StableDiffusionUI\_automatic1111.ipynb'
8. Run all cells
9. Under the module named 'Launch the WebUI,' you will find a URL such as  
**http://37uw51098weoiruwieur92skjfd984lsj.gradio.live (see the image next page)**
10. Now you are in Stable Diffusion Web UI (an excellent tool for text2img and img2img)
11. On the top menu, click 'Extensions,' we will use **controlNet** and **Roop** extensions.
12. Click the 'List' button to find controlnet and roop extensions from the list
13. Click it to install (do one at a time)

14. Click 'txt2img' back to main UI page

15. In the extensions tab go to the installed tab and then click on the "apply and quit" button. You then close the window and run the cell to open the GUI again

16. Again, under the module named 'Launch the WebUI' (see enclosed screen capture) you will find a URL such as <http://37uw51098weoiruwieur92skjfd984lsj.gradio.live>

Launch the WebUI

Run this block to launch the WebUI. You will get a link to nn.gradio.app, that's your WebUI. Follow it.

See [shared.py](#) to view the code for the launch args. There's a lot of good info in there about exactly what the args do. If you aren't a programmer, [here's the wiki](#).

Troubleshooting

- If you have any issues, try restarting the kernel.
- EOFError: Ran out of input: probably means you ran out of storage space and the model [ckpt] file wasn't downloaded completely. Try cleaning up your files. There are some helpful scripts in the Tools section below.
- The file may be malicious, so the program is not going to read it: means the program encountered unexpected data in the model file (the technical term is 'pickle'). Merging models can cause this. You can disable this feature by setting `[disable_pickle_check]` to True in the settings block.
- Try updating your notebook using the block in the Tools section below.
- If you're still having issues, delete `stable-diffusion-webui` and reinstall.

```
[ ]: try:
    store = model_storage_dir_repo_storage_dir activate_vformers activate_deepdanbooru activate_nevram disable_pickle_check gradio_port gradio_auth ui_themes insecure_extension_access gradio_queue
    test = model_storage_dir_repo_storage_dir activate_vformers activate_deepdanbooru activate_nevram disable_pickle_check gradio_port gradio_auth ui_themes insecure_extension_access gradio_queue
except BaseException as e:
    print("There is an issue with your variables.")
    print("Please go back to the first block and make sure your settings are correct, then run the cell.")
    print(f"Error: {e}")
    import sys
    sys.exit(1)

from pathlib import Path
local_path(Path(repo_storage_dir, "stable-diffusion-webui"))

# Code to set the options you want as defined in the very first block
arg = "--vformers" if activate_vformers else ""
dd_arg = "--deepdanbooru" if activate_deepdanbooru else ""
nvram_arg = "--nevrasm" if activate_nevram else ""
pickle = "--disable-safe-unpickle" if disable_pickle_check else ""
port = f"--port {gradio_port}" if gradio_port else "--share"
auth = f"--gradio-auth {gradio_auth}" if gradio_auth else ""
theme = f"--theme {ui_themes}" if ui_themes else ""
insecure_extension_access = "--enable-insecure-extension-access" if insecure_extension_access else ""
queue = "--gradio-queue" if gradio_queue else ""

# Launch args go below:
python webui.py {v_arg} {dd_arg} {nvram_arg} {pickle} {port} {auth} {theme} {queue} --enable-insecure-extension-access

2023-10-18 22:52:41.254 - roop - INFO - roop v0.8.2
2023-10-18 22:52:41.339 - roop - INFO - roop v0.8.2
Loading weights [6e0b161889] from /storage/stable-diffusion/stable-diffusion-webui/models/stable-diffusion/v1-5-pruned-emaonly.safetensors
/storage/stable-diffusion/stable-diffusion-webui/extensions/sd-webui-roop/scripts/faceswap.py:38: GradioDeprecationWarning: Usage of gradio.inputs is deprecated, and will not be supported in the future, please import your component from gradio.components
img = gr.inputs.Image(type="pil")
/storage/stable-diffusion/stable-diffusion-webui/modules/gradio_extensions.py:25: GradioDeprecationWarning: optional parameter is deprecated, and it has no effect
res = original_image_unit(eaf, args, "images")
/storage/stable-diffusion/stable-diffusion-webui/extensions/sd-webui-roop/scripts/faceswap.py:55: GradioDeprecationWarning: Usage of gradio.inputs is deprecated, and will not be supported in the future, please import your component from gradio.components
upscaler_name = gr.inputs.Dropdown()
/storage/stable-diffusion/stable-diffusion-webui/extensions/sd-webui-roop/scripts/faceswap.py:74: GradioDeprecationWarning: Usage of gradio.inputs is deprecated, and will not be supported in the future, please import your component from gradio.components
model = gr.inputs.Dropdown()
2023-10-18 22:52:42.786 - ControlNetFastLoad - INFO - Load ControlNet FastLoad Filter on iskenote=True and accessLevel=1
Running on local UI
Creating model from config: /storage/stable-diffusion/stable-diffusion-webui/configs/v1-inference.yaml
Running on public UI - http://37uw51098weoiruwieur92skjfd984lsj.gradio.live
This share link expires in 72 hours. For free, transient hosting and GPU upgrades, run 'gradio deploy' from Terminal to deploy to Spaces (https://huggingface.co/spaces)
Startup time: 21.5s (import torch: 8.6s, import model: 0.8s, setup paths: 1.8s, initialize shared: 0.3s, other imports: 0.5s, setup codeformer: 0.1s, load scripts: 0.4s, create ui: 1.0s, gradio launch: 1.9s).
Applied attention optimization: vformers...
Model loaded in 8.6s (load weights from disk: 6s, create model: 0.6s, apply weights to model: 3.3s, apply half(): 0.7s, calculate empty prompt: 0.5s).

[ ]: !pip install protobuf==3.20.2

Export Generations
This block will rename and compress the outputs with zip max compression. It expects you to have log/ and outputs/ in //notebooks/stable-diffusion-webui/.
```

17. Back to the 'txt2img' main UI page

18. You may background the screen by adding the following code (`/?__theme=dark`) in the web browser address.

19. Now you are ready to start the lab.

## Prompt Engineering Lab

20. Based on Dr. Choi's lecture, think about a target image that you and your partner want to create. Both people have to agree. Make sure to choose an image that will utilize all the lesson points.

21. For each incremental prompt, create and save a few images (you may use the batches setting to how many images)
22. Open a MS Word and create a table that captures all your prompt-generated incremental images with a brief description for each prompt. Your final submitting table would have the series of images, their corresponding prompts and descriptions.