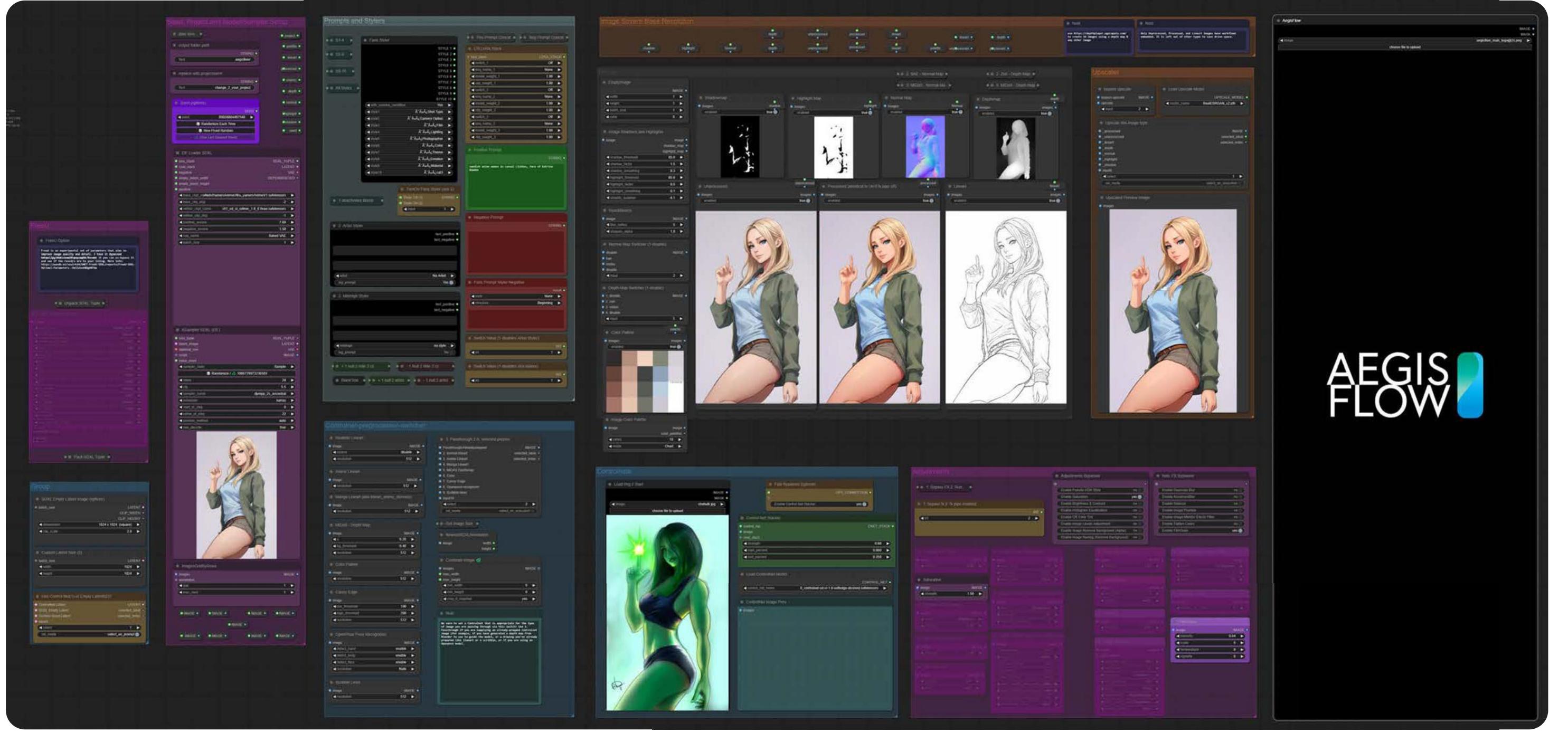


AEGIS FLOW

ComfyUI workflows for creative professionals,
for use with SD 1.x and SD XL



Hi! This is my personal workflow that I created to enable me to use generative AI tools on my own art and on my job as a working artist. Hopefully, it can help you too.

This isn't intended to be "the workflow to end all workflows." It isn't intended to perform endless inpainting, etc. It may grow to that, but for now, I had simple goals for this tool:

- To reimagine my previous work
- To test ideas quickly without having to fully render new images
- To generate multiple related assets in a "project" style workflow, and then automatically name and associate them with each other in a project folder.

This workflow is Donationware. I think folks should be compensated for art, but I also agree that art should be for everyone. This workflow is completely dependent on the hard work of others making art & generative AI work, whom I believe you should support as well in whatever method you are able.

Distribute this workflow as you wish, and if you find it useful, support the folks that make your life easier and more creative. That includes continuing to expand your own technique... AI should be a tool, not a replacement for your own point of view, artistic vision, or message.

If you don't want to deal with the intricacies of setting everything up, downloading checkpoints, etc, I do sell a portable installation of this workflow and all the software pre-installed on a 256gb flash drive for \$60 shipped. It will run on any Windows machine capable of running Stable Diffusion. That price covers my time, shipping and the cost of the drive---the software remains governed by the licenses of each individual component of the workflow. I'm also working on a Linux compatible version.

Visit me on Gumroad at
<https://majorstudio.gumroad.com/l/aegisflowUSB> to purchase one,
or support my ongoing efforts at majorstudio.gumroad.com/l/members.

Don



Aegisflow is a workflow that uses the amazing ComfyUI interface for Stable Diffusion, and a variety of extensions to that platform.

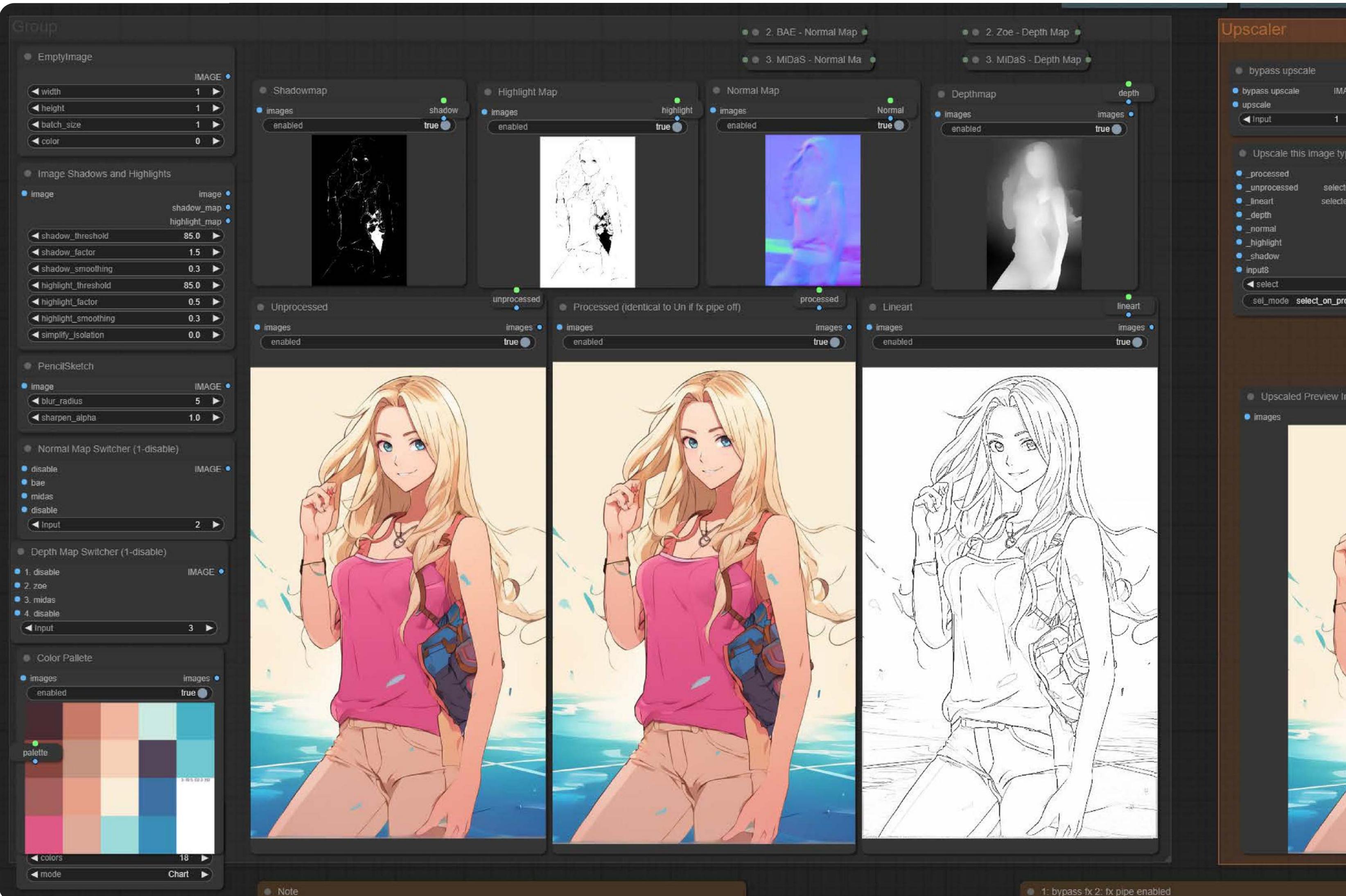
Aegisflow can take an image you provide which guides, via controlnet, along with stylers, LoRAs, and of course the familiar “prompts,” to create an image. This is then turned into several variant images that can be used in other programs (Photoshop or Blender, for example) to help you with your creative process.

The end output of Aegisflow consists of:

- a “raw” image. This is the AI’s interpretation of your inputs before undergoing other processing.
- a “processed” image, made from the raw image after being passed through the adjustment and FX banks.
- a “lineart” image, consisting of a lineart projection of the original image that could be used in several manners within AegisFlow or external programs.
- highlight and shadow extrapolations.
- extrapolated normal and depth maps that attempt to give depth processing for use in 3D platforms or texturing programs.
- a color palette of averaged colors in the image, in whatever number of colors you specify.
- upscaled image(s). One can select any of the previous images to be upscaled using the technique of your choice.

Of these images, the workflow is included within the unprocessed, processed, lineart, and upscaled images. For the purpose of saving space, other images are saved without the embedded workflow information.

These images are saved in a folder that is named to your specification in the AegisFlow system. The foldername uses just the name of the project, while images inside will use the projectname joined with the seed number and the image type.

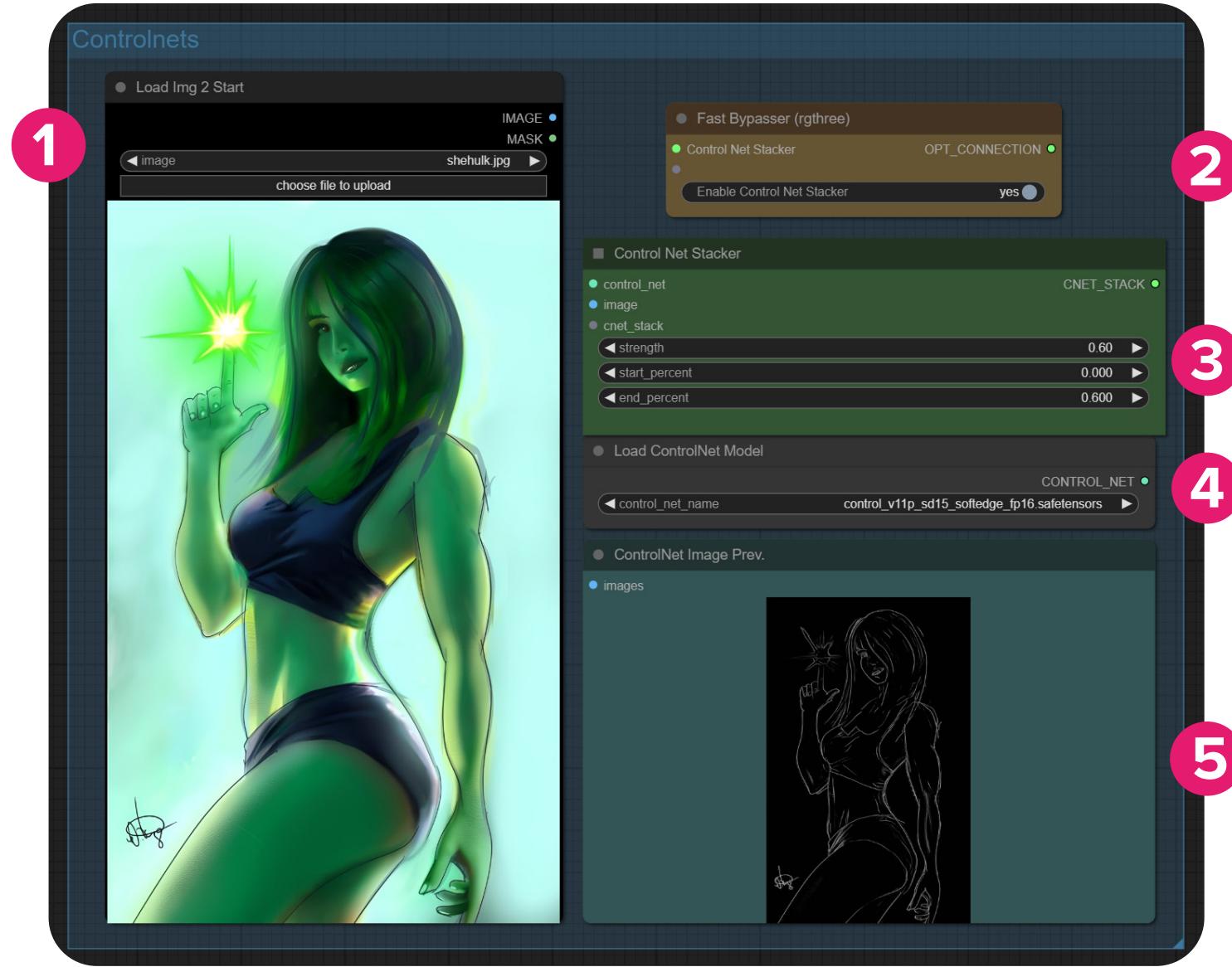


What does AegisFlow do? What does it produce?

AEGIS
FLOW

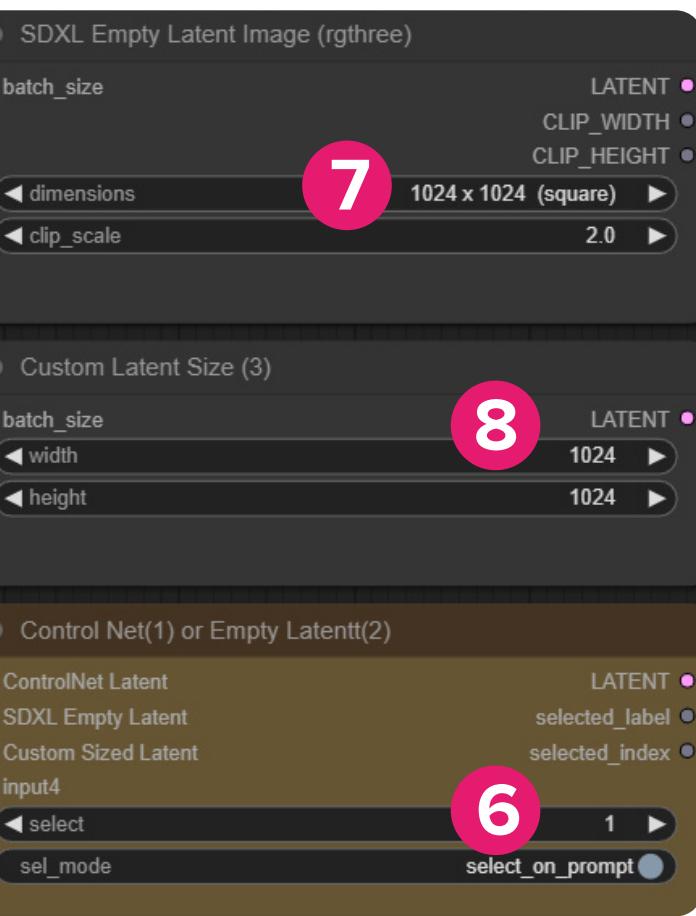
Step 1: Set Up Your Controlnet Image

Firstly, while using image guidance is not required (you can prompt text only), the main idea of this workflow is to use your art as the AI's guidance. Here, we'll start with loading a controlnet image. Here, I'm using my "Gamma Girl" drawing & I'm using it to guide towards a woman in casual clothes.



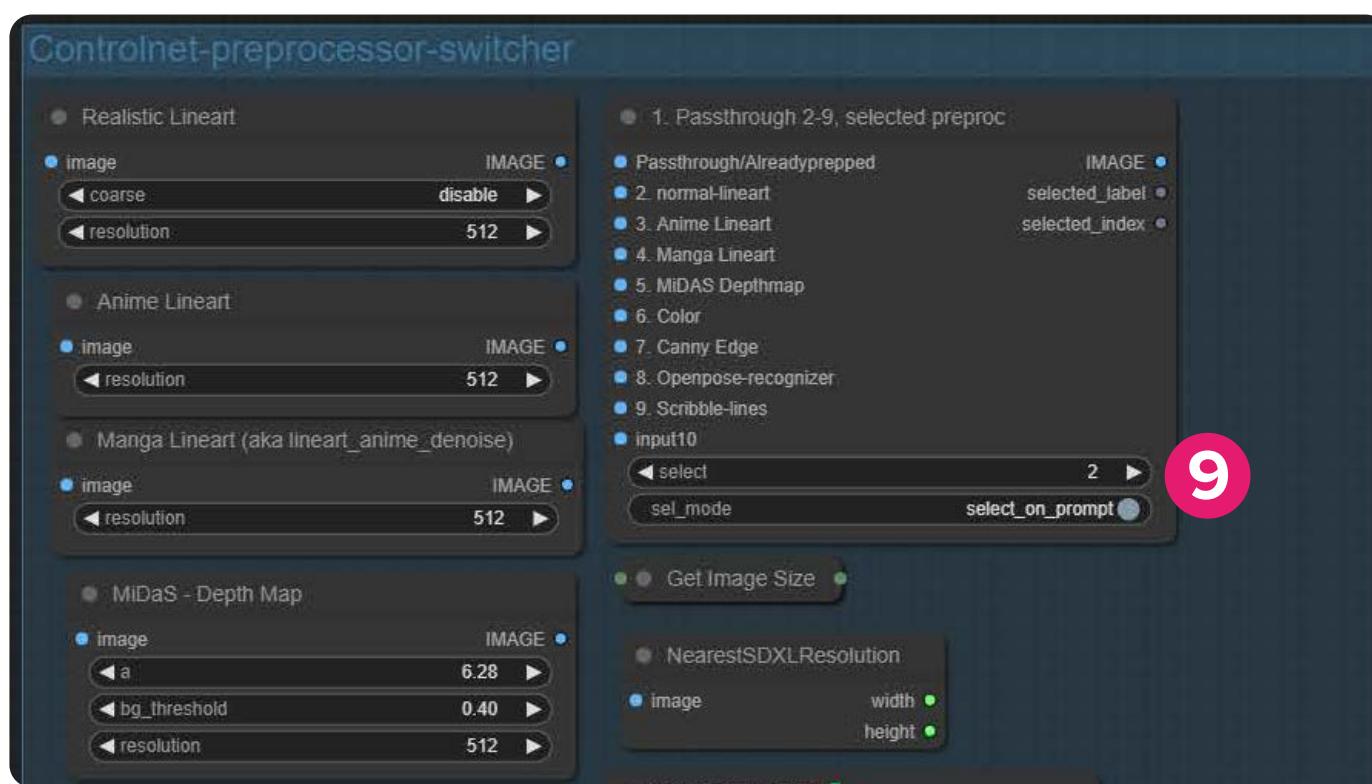
- 1) This is where you upload your image. It will be resized and cropped to the nearest SDXL standard size.
- 2) Should you wish to not use controlnet guidance, you can bypass it. The image size will still be used as the latent.
- 3) These are controlnet settings. The top number tells the system how closely to follow guidance. 0 would effectively turn off guidance as if you put in no image at all. Numbers near 1 will enforce guidance heavily. 0.4 to 0.7 are decent. Varying this and the sampler CFG (Step 3) will help you narrow in on the right combo for the desired outcome.
- 4) This is the type of controlnet that will interpret the guidance.
- 5) This is the result produced by selections 1-9 in the group to the left called *controlnet preprocessor switcher*.

If you want to not use a control image, there is a switch to the left of the loader/sampler (6). By default, it is set to 1, which uses the controlnet image for the latent size. If set to 2, the latent image is replaced by an SDXL-compatible empty latent image as specified in (7). If set to 3, an empty latent of the size you specify in *custom latent size* is used (8).



Step 2: Choose how to Process your Input Image

Next, you'll select the proper preprocessor for the image you are using as input (9). If it is a hand drawing like this, then using 2, 3, or 4 might work well. 1 (*passthrough*) might be appropriate if, for example, you are passing in a pre-made Openpose stick figure or a depth map generated in a 3D program. There are examples in the Appendix that visualize some of these choices, but this guide isn't designed to be an exhaustive overview of controlnet operations. For more information, consult YouTube University :).



Step 3: Setup your Loader and Sampler

Now we'll move to the far left of the workflow, where we will select our project name, seed generation, checkpoint/model, VAE type, and our various sampler options.



in **output folder path**, you can make a subfolder that gets created in your ComfyUI outputs directory.

Type in a ***name appropriate for your project***. Here, we might use 'swedish woman anime' or whatever.

Select a **seed**. I like 'new fixed random,' personally.

Choose a **model**, **refiner** (for XL based models) and a **VAE** if needed.

Sampler settings are for you to decide.

Steps determines how long the AI works on the image. Just like real life, more isn't always better. 20-25 steps is usually enough.

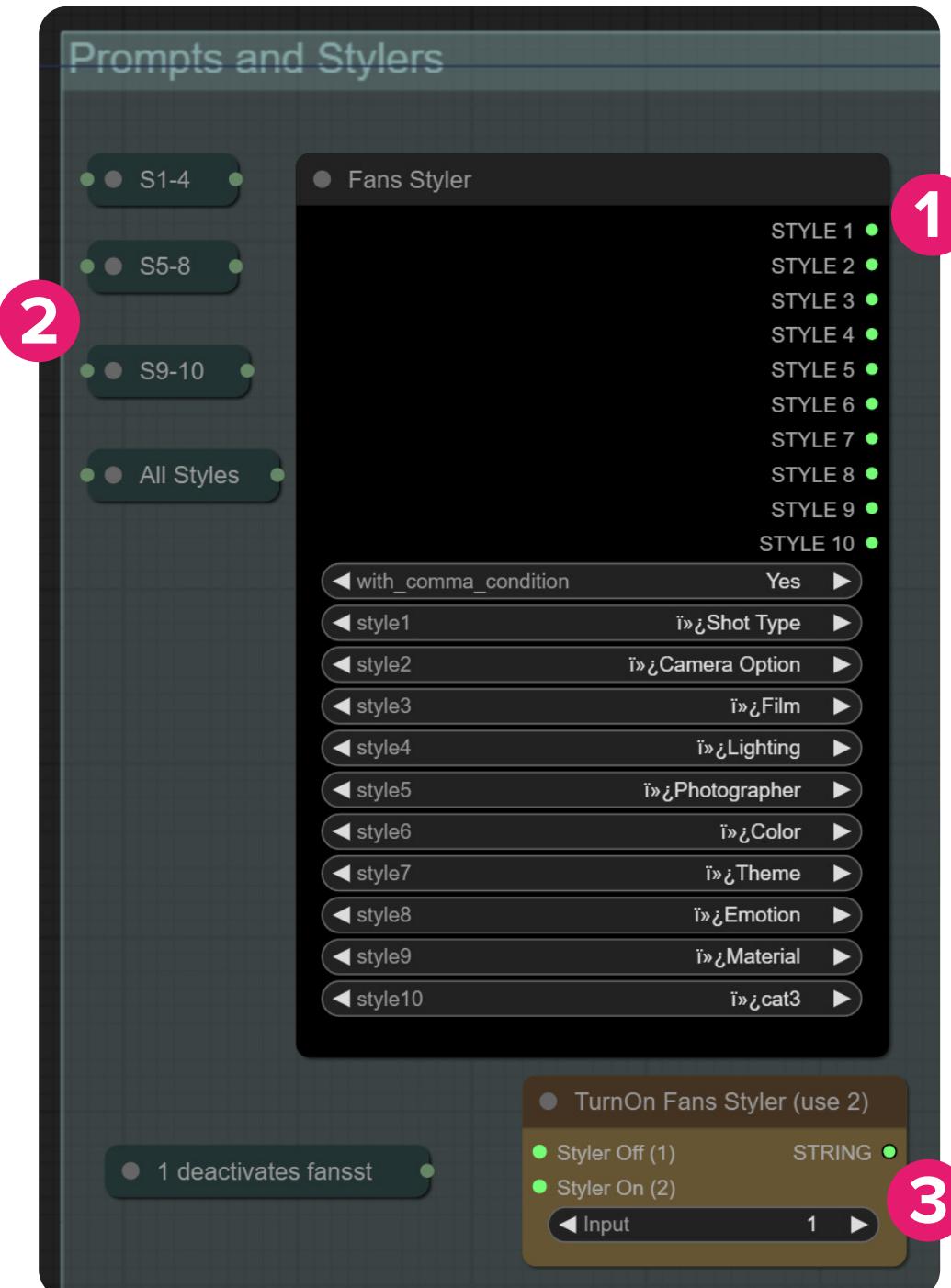
Cfg determines how much the AI will have freedom over the final look. Consult YouTube U for generalized Stable Diffusion knowledge.

Step 4: Set Up (or Disable) Stylers

AegisFlow has some styling components built into the workflow. Each can be used alone or in combination with each other, along with your own Positive and Negative prompts (this is also called “conditioning”).

Fans Styler

A popular add-on that allows you to apply a variety of styles through drop-down menus (1). The switch in yellow-brown at the bottom turns the styler on and off (3). Throughout Aegisflow, you will see “rolled up” nodes and reroute nodes (2). If a node or a connector is rolled up, don’t mess with it unless you know what you are doing. If you break something, just reload a known good generation from your history or a recent generation rather than trying to fix it.



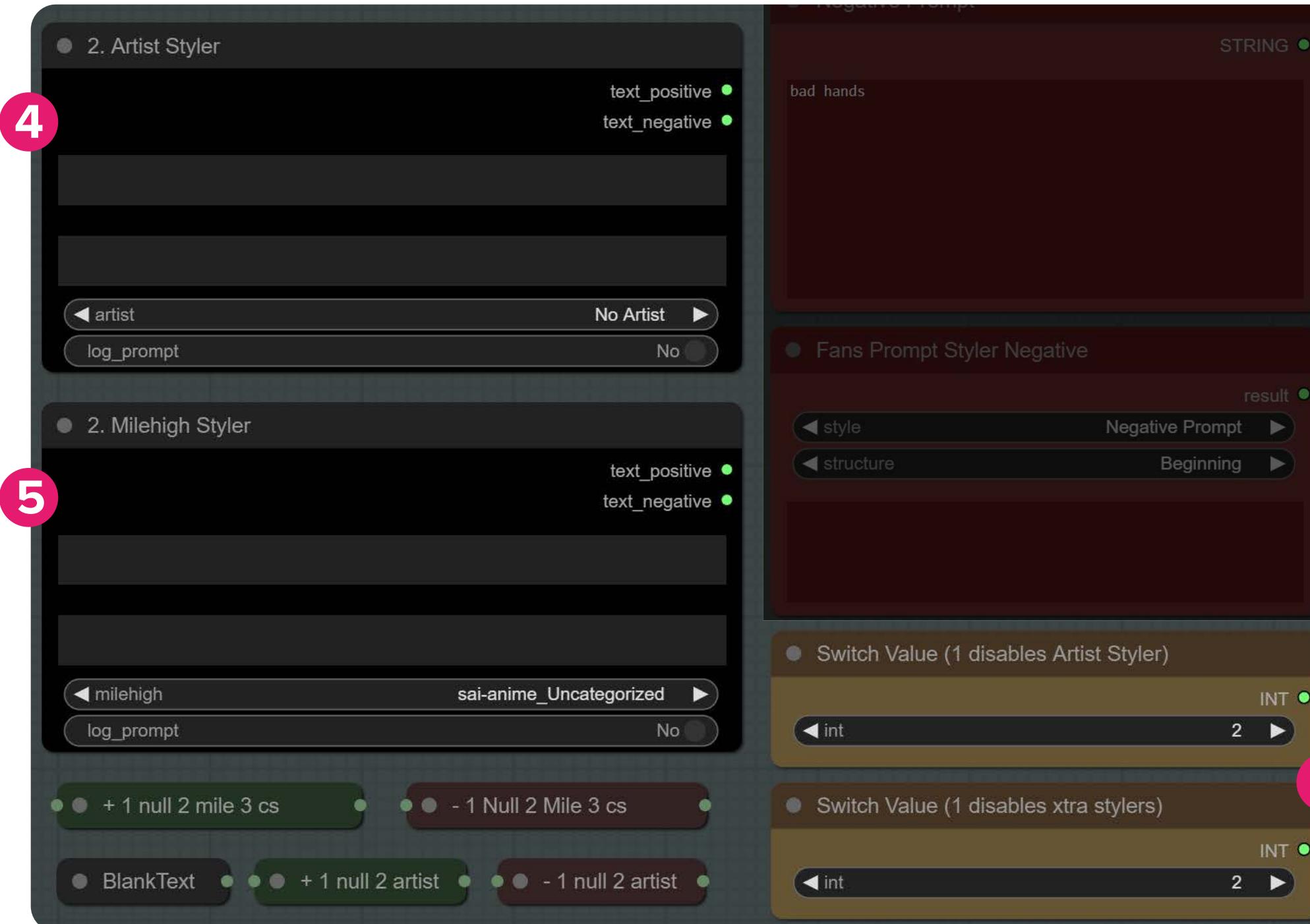
Artist Styler (4)

Another add-on that applies conditioning, the artist styler tries to emulate a huge variety of styles from artists and art movements. This can be both used and abused, TBH, and I would suggest to use these styles sparingly and in conjunction with your own work---but I’m not your mom. In the end it’s your call how to ethically use these tools.

Milehigh Styler (5)

Similar to ArtistStyler but more focused on emulating techniques than artistic styles. As with all AI art, the process is iterative and experimental.

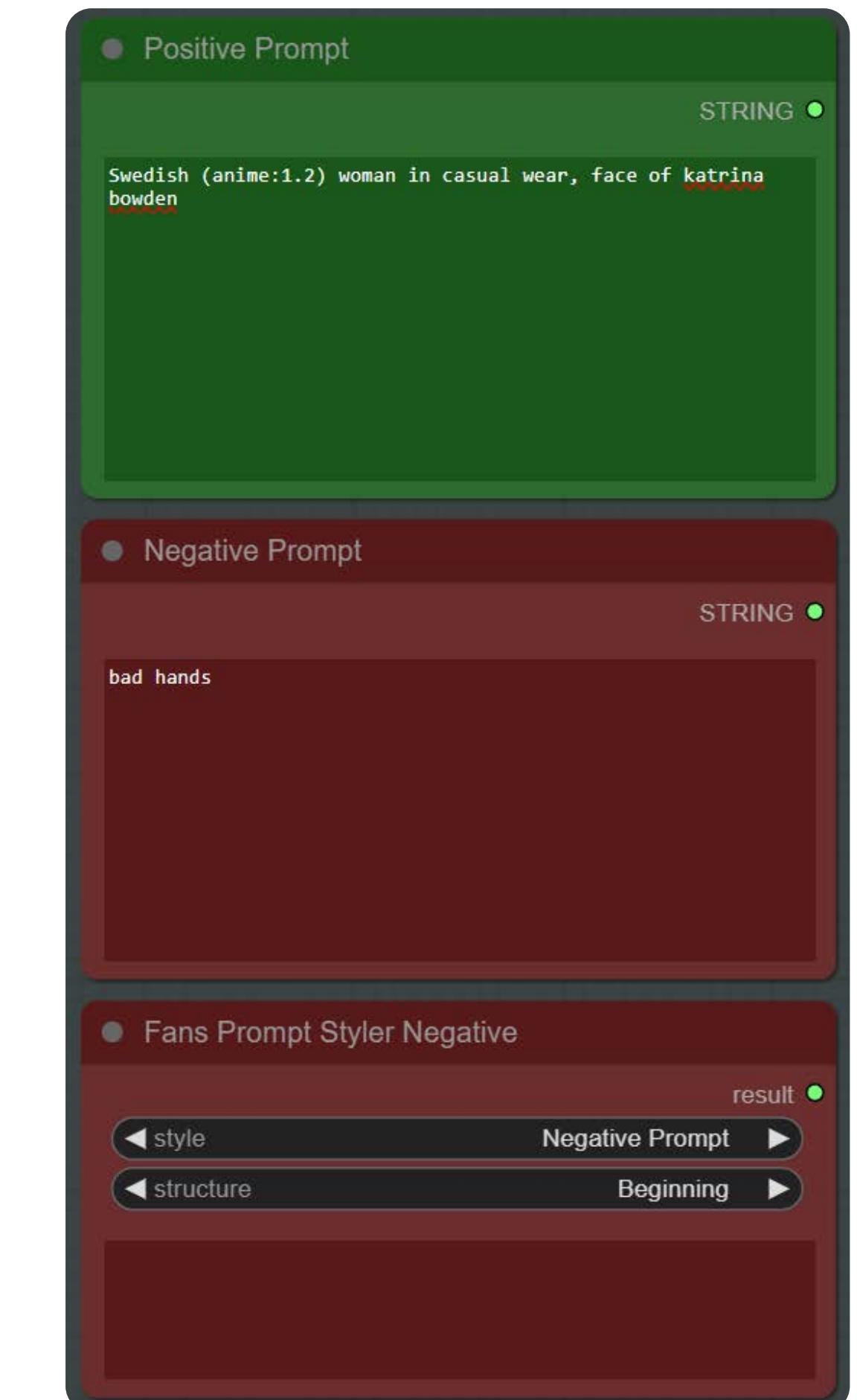
As before, and throughout AegisFlow, The switches in yellow-brown at the bottom turns the stylers on and off (6). Use 1 as the value to disable, or 2 to enable.



Step 5: Input your text prompts

Prompts

Here is where you put in your positive prompt (what you want to see), your negative prompts (what you don’t want to see) and can select from some pre-existing negative prompts developed by members of the community.



Step 6: Add in LoRAs (optional)

LoRAs are “guides” for the AI that help generate an image. A LoRA creator trains an overall look that will apply to the whole image. Sites such as Civit.Ai and HuggingFace, amongst others, have large collections of LoRAs designed to produce things from the broad to the hyper-specific.

Some LoRAs are general-purpose and can be used merely by selecting it and giving it values that define how much influence the LoRA should have (similar to controlnets). Other LoRAs may require one or more “trigger” phrases that tell the AI to use the LoRA. With the USB version of AegisFlow, i’ve named included LoRAs according to their triggers, like so: *(trigger words)_loraname.safetensors*.

It’s important to remember that LoRAs are specific to the type of model being used. SD 1.5 LoRAs won’t influence SD XL output (and in fact will likely throw a “dimension mismatch” error). The same is true in reverse. It’s best to label your LoRAs or put them in a folder according to the version of SD to which they can be applied (as I have).

At the right are 8 images with differing LoRAs applied, using the same seed and conditioning.

LoRAs are placed inside the models\loras folder of ComfyUI.

For those that don’t know, Katrina Bowden played Cerie on 30 Rock, one of my favorite shows ;).



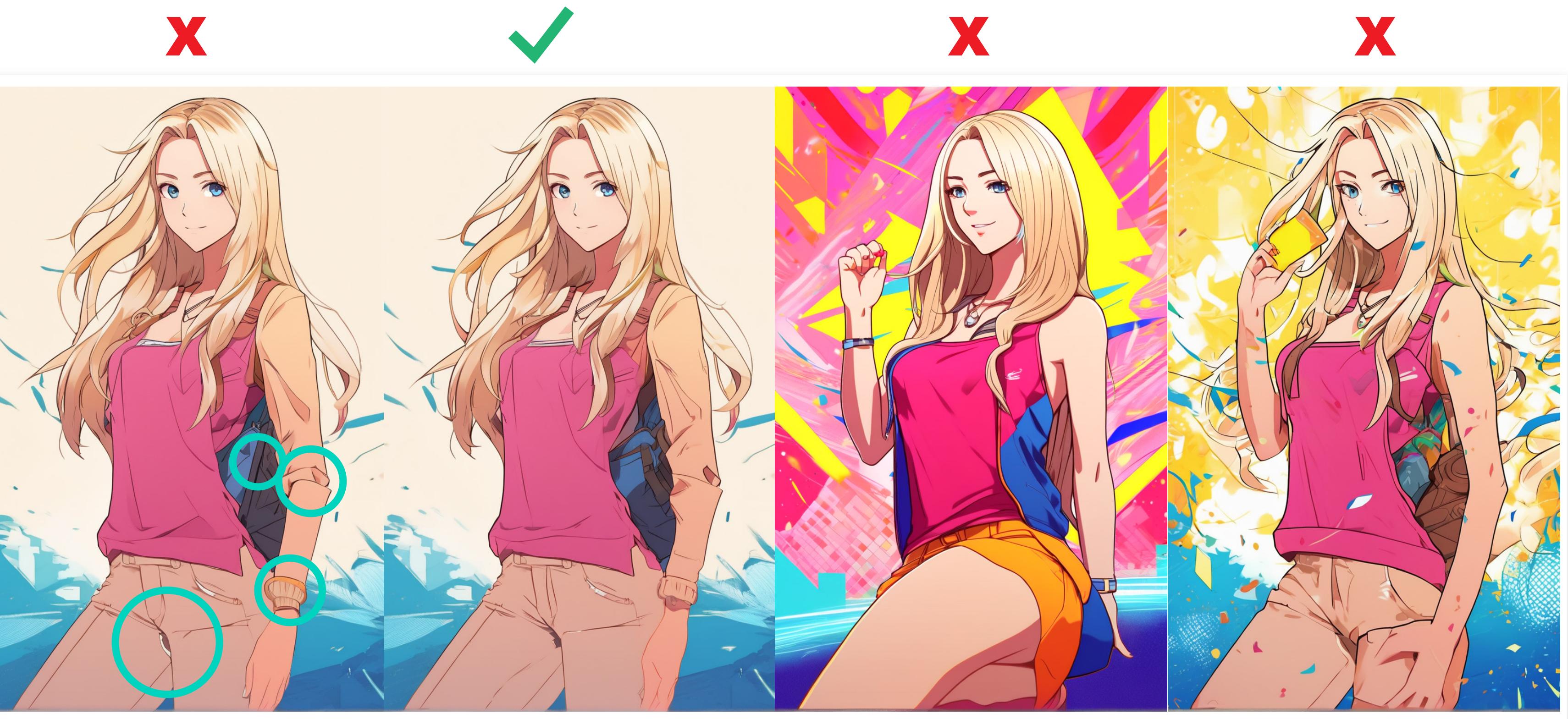
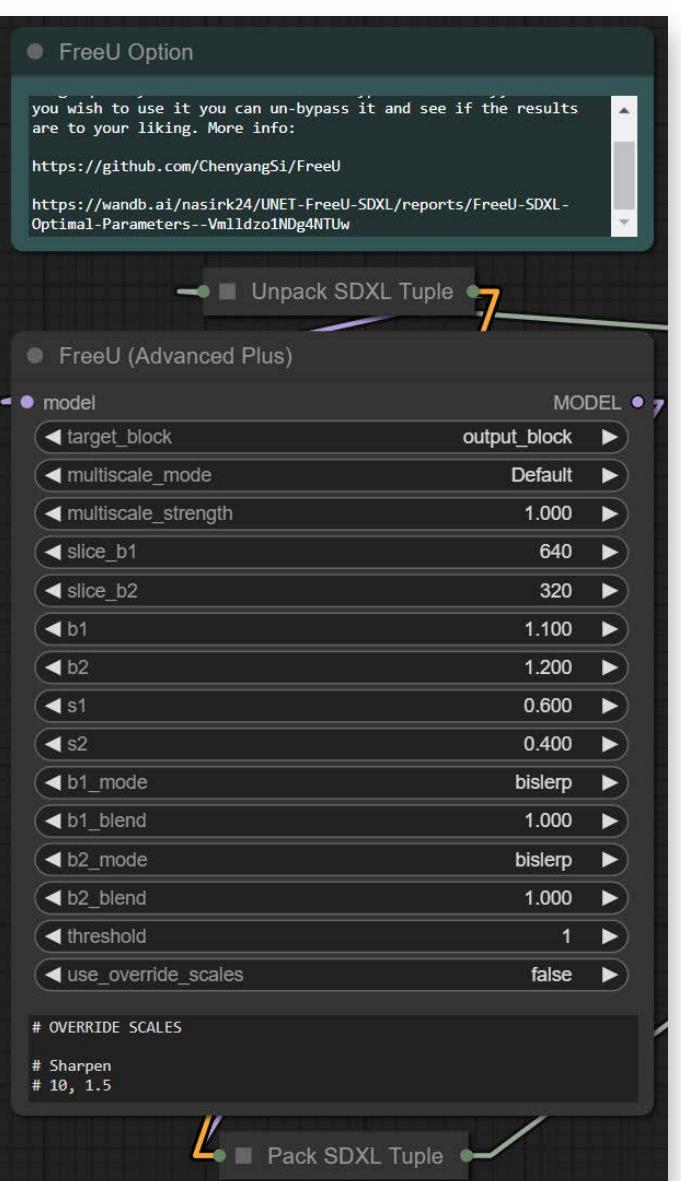
Step 8: Experiment with FreeU (VERY optional)

FreeU is a fairly recent development. Using “math,” and that’s as far as I can go there, it aims to improve SD performance and output.

Like many things in SD, the results are somewhat image dependent and small changes in parameters can lead to large changes in the images produced. (see right).

You’ll find this bypassed by default and to the left of the Loader/Sampler section. Right click and click “bypass” to toggle it on.

Give it a try, and see how it works for you. Initial reports seem to suggest it works best on SD 1.5 models. The series of tubes known as the internet has recommended values based on SD version (1.4, 1.5, 2.x, and XL)



No FreeU, base generation.
See artifacts circled and
compare with next...

Using FreeU on Middle Block,
Subtle Improved Quality,
artifacts fixed

Using FreeU on Output Block,
Decreased aesthetic quality &
massive changes.

Using FreeU on Input Block,
Decreased aesthetic quality &
massive changes.

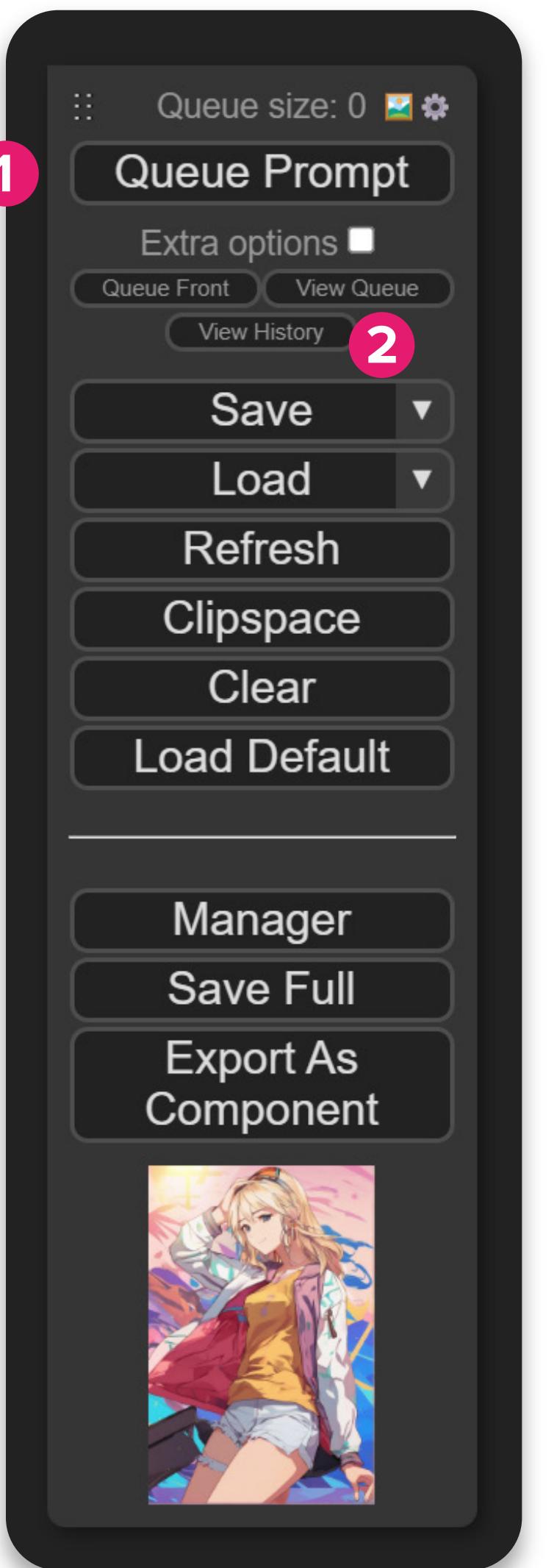
Step 10: Generate and Output!

We're ready! Time to press the "queue prompt" button in the manager (1). Once you do, This will happen:

1. Your input image will be processed by your chosen controlnet preprocessor (or passed through).
2. The Controlnet model will load.
3. The Main model/checkpoint and refiner (if needed) will load. This can take a long time--sometimes 10 minutes for SDXL-- so be patient during the first generation. This is triple-true if you are using the CPU instead of a GPU.
4. The sampler will take your conditioning (prompts, controlnet data, etc) and start generating your image.
5. Aegisflow will then produce the various offspring images (shadowmap, processed, depthmap, lineart, etc.). If you've enabled the adjustment/FX pipeline, and have any enabled, those will execute and pass the results to the *Processed* image (between *Unprocessed* and *Lineart*).
6. It will send all these images to a section of nodes that saves these images out to the specified folder/filenames.
7. If upscaling is enabled, your chosen upscaler will load and the image you've selected will be upscaled.
8. Your upscaled image will be saved, as "projectname_seed_imagetype_upscaled.png"

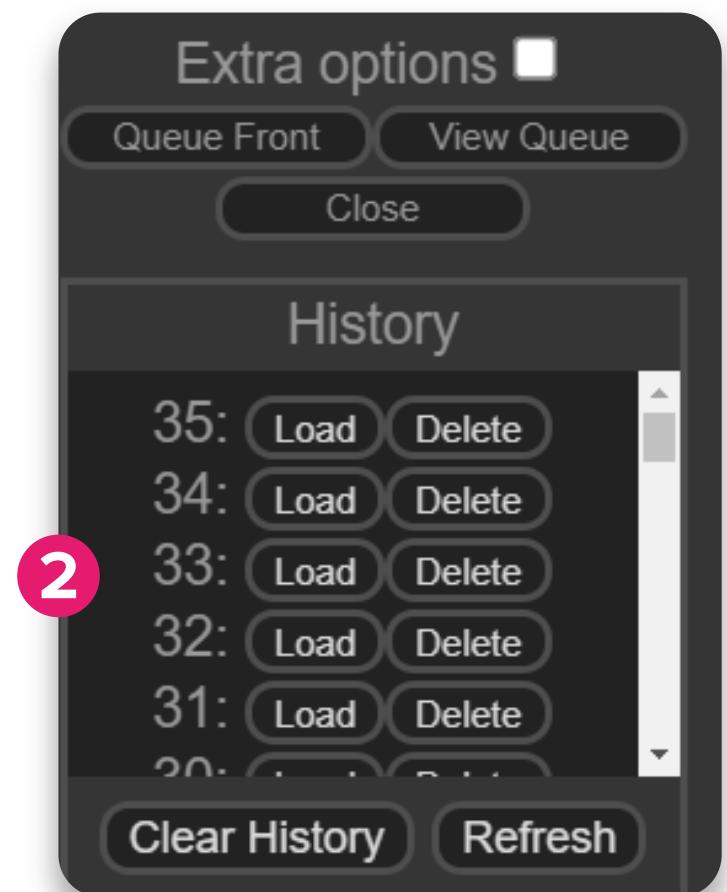
That's all! Note that if you change the image conditioning/prompts, but not the project name or seed, it will overwrite previous images. This is so that you don't fill your drive with bad images as you tune the prompt.

AegisFlow isn't supposed to be an image mill, churning out thousands of Waifus or robot-skull pictures. It's a more considered approach, where you select and iterate to get the result you need.



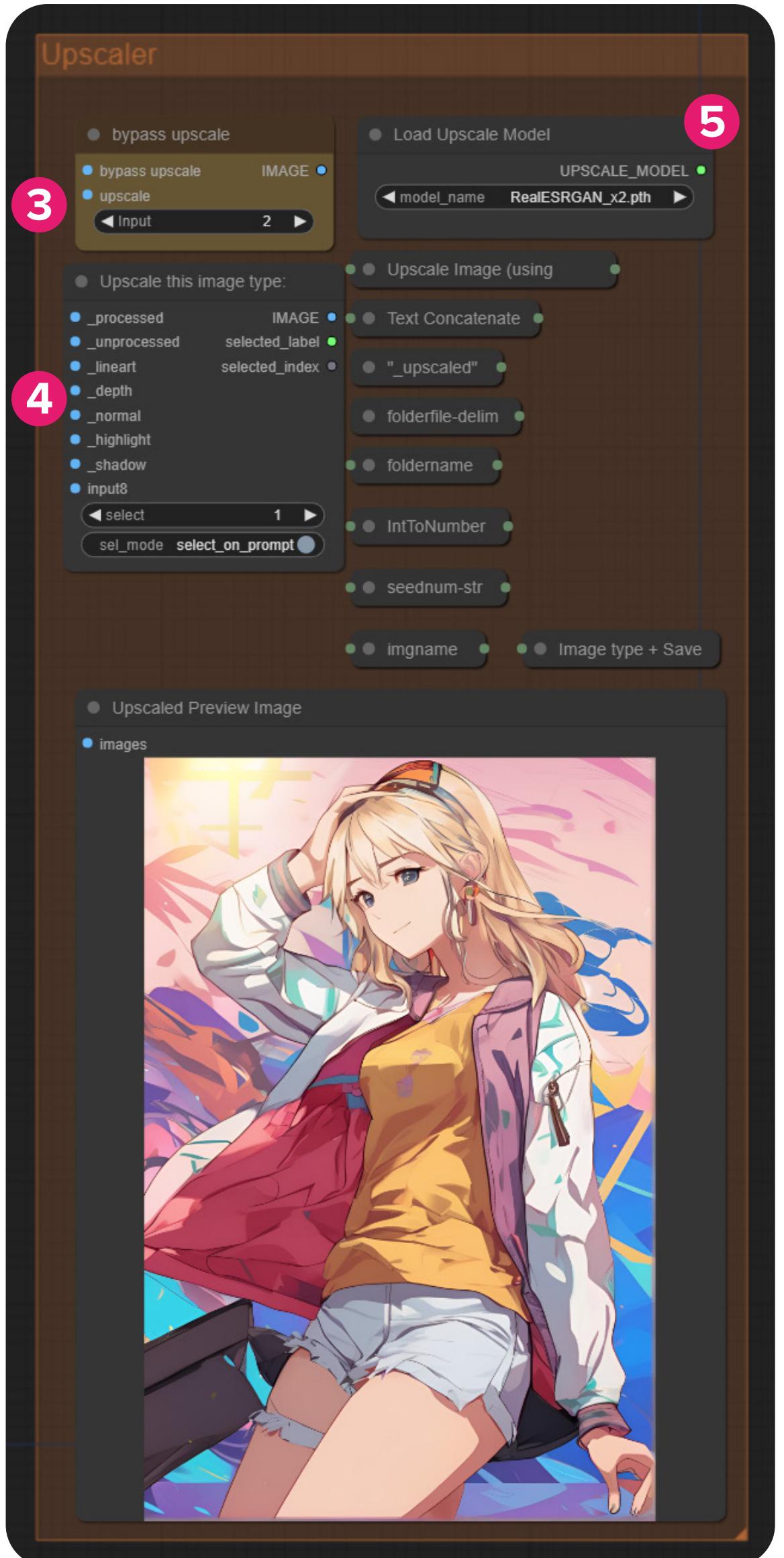
If you want to save a generation but keep the seed, make sure you change the projectname.

Of course, ComfyUI keeps a queue of past jobs as long as the task is running, so if you forget, load a previous generation from the history (2) and change the project name, then re-run the generation of that image.



Finally, for upscaling, you need to set the upscaler switch to on (3), choose an image type from the selector (4) and choose an upscaling model(5). If you're using the same settings and seed, and press *Queue Prompt*, then the workflow will take your selected image type and upscale it without needing to perform all the previous steps.

Note: If the upscaler is bypassed, you will still see an image in the "upscaled Preview," but the image will not be actually scaled. Only when the upscaler is enabled will you produce an actual upscaled image.



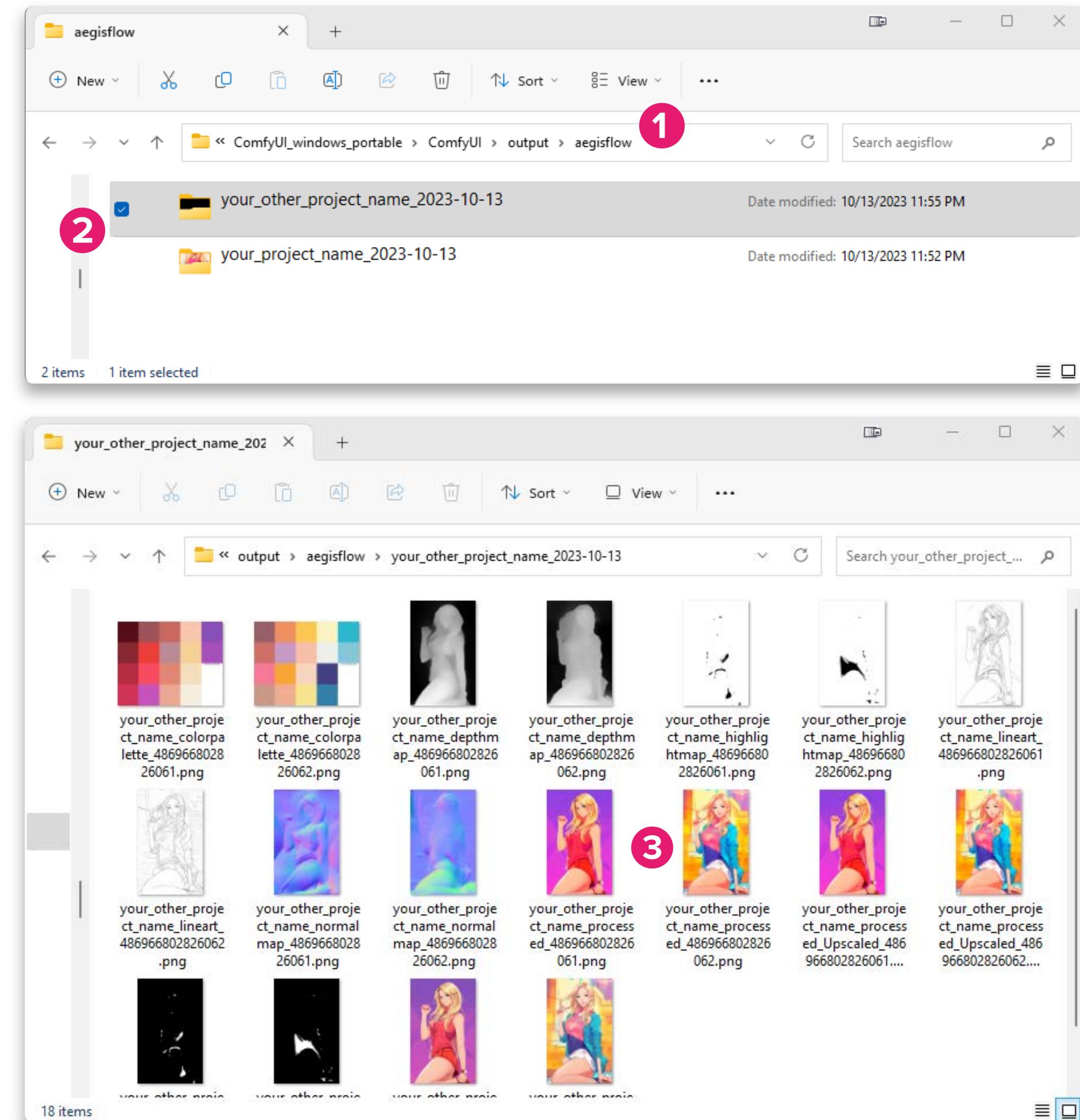
File Structure

Shown here on Windows, but it would be similar on most systems. The main directory or directories would be set in step 1, as would the project name. In this case, I've set the main directory to "AegisFlow" (1), and the projects are located underneath this directory (2).

Inside the project folder will be the saved files, arranged by seed numbers.

Of course, you can set this up however you'd like. Maybe you have a game that you're conceiving two characters for. You might make main directories called "Jack" and "Jill," and under those, maybe you make projects like "jack_winter_gear" or "Jill_scuba_outfit."

The idea is to plan it out a bit in your mind, and then you can arrange things as granularly as you'd like. Should you ever want to come back to an image, take one of the workflow-enabled images (processed, unprocessed, or lineart) and drag it into your comfyUI window to load the workflow. The seed you used will be populated back in. Generate, or make changes and generate, and you're back to working that concept. Easy!

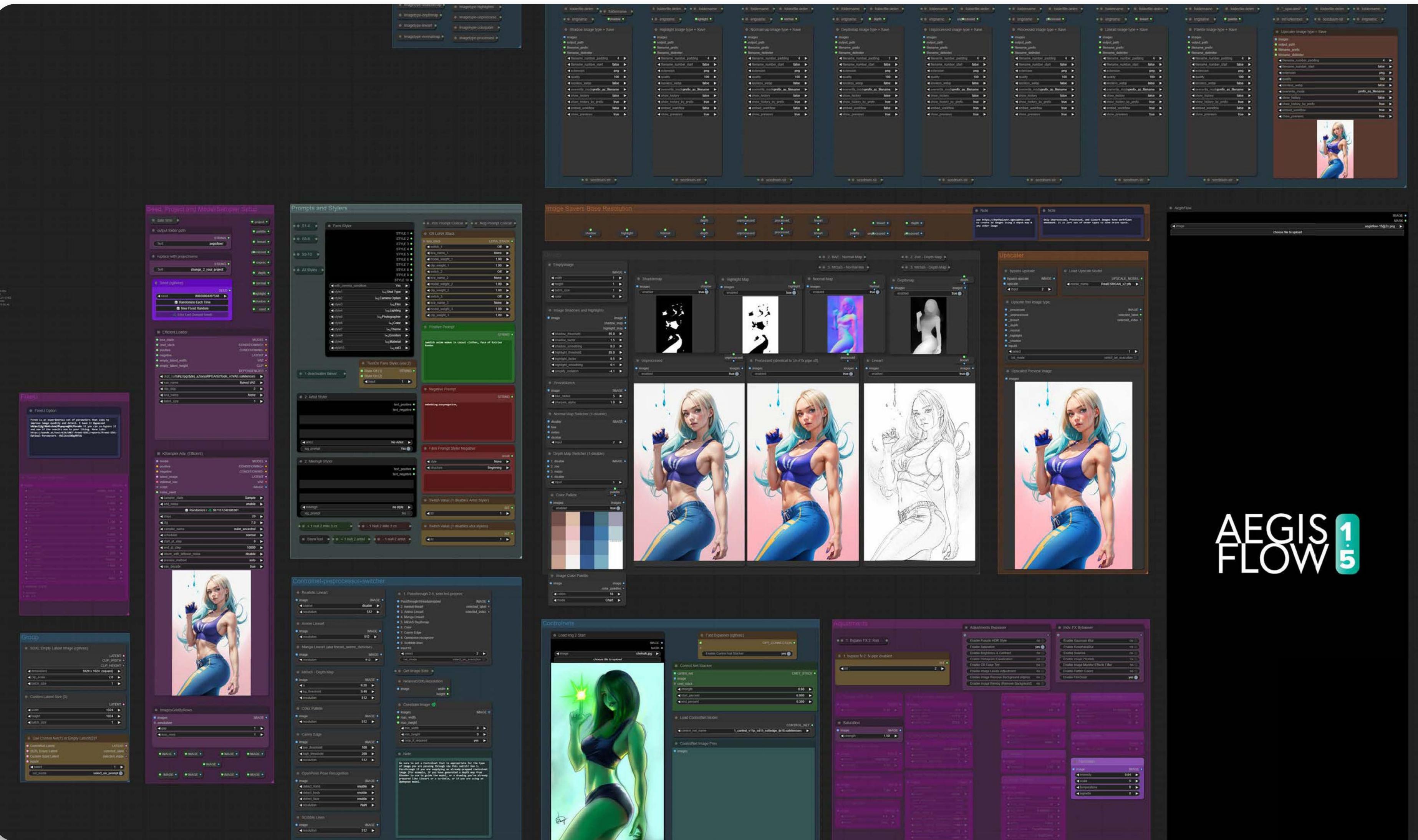


A special Workflow for SD 1.5

SD XL is the new hotness, but that doesn't mean SD 1.5 is old & busted. Quite the opposite; it has good support and massive amounts of trained checkpoints available so you need to be able to use it too. The issue is that SDXL and 1.5 are built and work differently.

AegisFlow 1.5 to the rescue. This is a workflow that uses 1.5 compatible loaders and samplers so that you can fire up SD 1.5 models and use them in the same basic way as AegisFlow XL, without needing to make lots of configuration changes.

In fact, if desired you could have two custom installations of ComfyUI, and then only include compatible models and LoRAs etc in each, avoiding errors (only one could be run at a time on most systems though).



What about SD 1.5?

**AEGIS
FLOW** 1.5

AegisFlow 1.5 and XL on Physical Media

Don't have time to configure your ComfyUI installation to run AegisFlow? Don't want to spend countless hours getting all the proper SD checkpoints and dependencies? Is your time worth around \$2 an hour?

If so, you might consider purchasing a portable install on a 256gb USB thumbdrive. To be crystal clear, you CAN do all this yourself. You can track down missing dependencies, LoRAs, ControlNets, et al and download 200gb or so yourself, and install them via Git and in other ways.

But if you have a bit of extra cash and not much time, I can provide a pre-configured drive that will work on any Windows machine that meets the requirements of Stable Diffusion and Comfy UI. This is generally a fast PC with an NVIDIA GPU of a recent generation (1080 GTX series at bare minimum, 3xxx RTX series and above recommended). Older cards won't have enough VRAM to run, although one can run SD on CPU—slowly.

Check out MajorStud.io/aegisflow for info on how to get a drive sent out for \$60 shipped in the USA.

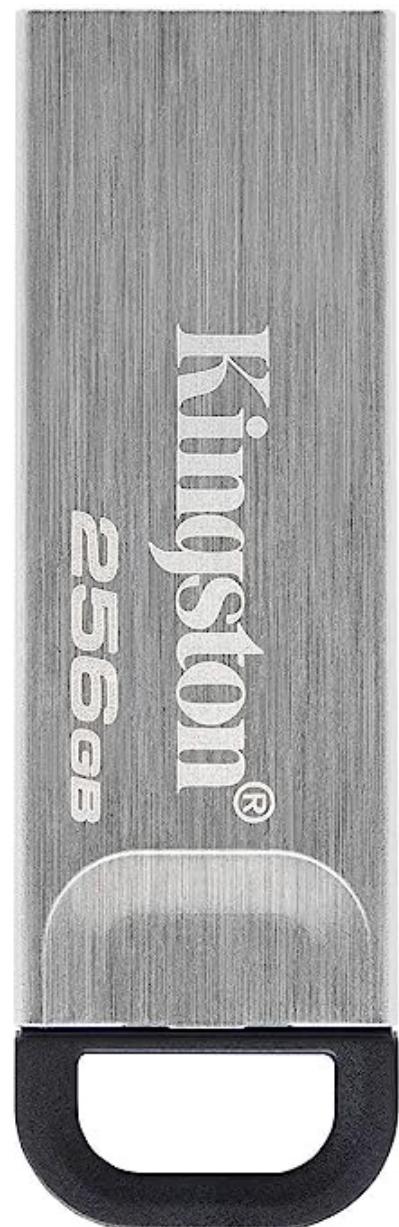
Once you have the drive, you can run it off that drive directly, and/or transfer it to another and keep the original as a backup (recommended).

What does it include?

The drive has over 200 GB of Stable Diffusion resources as well as AegisFlow 1.5/XL and other toolset workflows for specific, discrete tasks. It includes many plugins and add-ons, as well as a series of test images for experimentation with controlnets and more.

All of this is arranged so that the models and LoRAs are organized by SD version.

I use drives from Sandisk, Samsung, PNY Kingston and other name brand manufacturers based on availability.



Legalese

This workflow and some of the test images are things that I have personally created. The rest is the product of the folks in the FOSS community some are listed under acknowledgements explicitly or implicitly..

I am providing this workflow as-is and without warranty for use in or for any specific purpose, and the software required (stable diffusion checkpoints, add ons, ComfyUI, Python, et al) are provided as-is and are governed by the licensing agreements from those creators.

I recommend highly that you use powerful tools like generative AI responsibly and ethically, but I am not in a position to nor do I have the ability to view, direct, or determine what you use this software for. If you use it to do anything against the law, that is 100% on you. My suggestion? Don't be gross and don't be a jerk. It will be better for you, the AI Art community, and the world in general.

A non-exhaustive list of things you should not do is generate images of real-world humans in ways that cause them harm in any way, real or perceived, generate imagery that could be considered illegal in your area, generate images as mis- or disinformation, attempt to impersonate, defame, or harm other persons, and other bad stuff. You're not dumb, and you know what bad is, or at very least what people consider bad. Use common sense.

Non-Legalese, in closing

I had a good time making this, and I hope it can help you in some real way. Whether that is in concept art, generating “stock” imagery, in therapeutic efforts, or simply for enjoyment I hope that this helps you produce something meaningful to you. Never stop learning, and remember that a tool is only as good as the person using it.

Use AI, don't let it use you. Use it to increase your output, polish your work, rapidly explore new ideas, and generate assets that can be combined with other assets and your own art to make new and cool stuff. Use it to poke at new frontiers.

Don't use it as a crutch :). Learn the fundamentals of composition, form, color, and more so that you can guide the AI towards what your vision is.

Above all, enjoy it, and enjoy life. Be well.

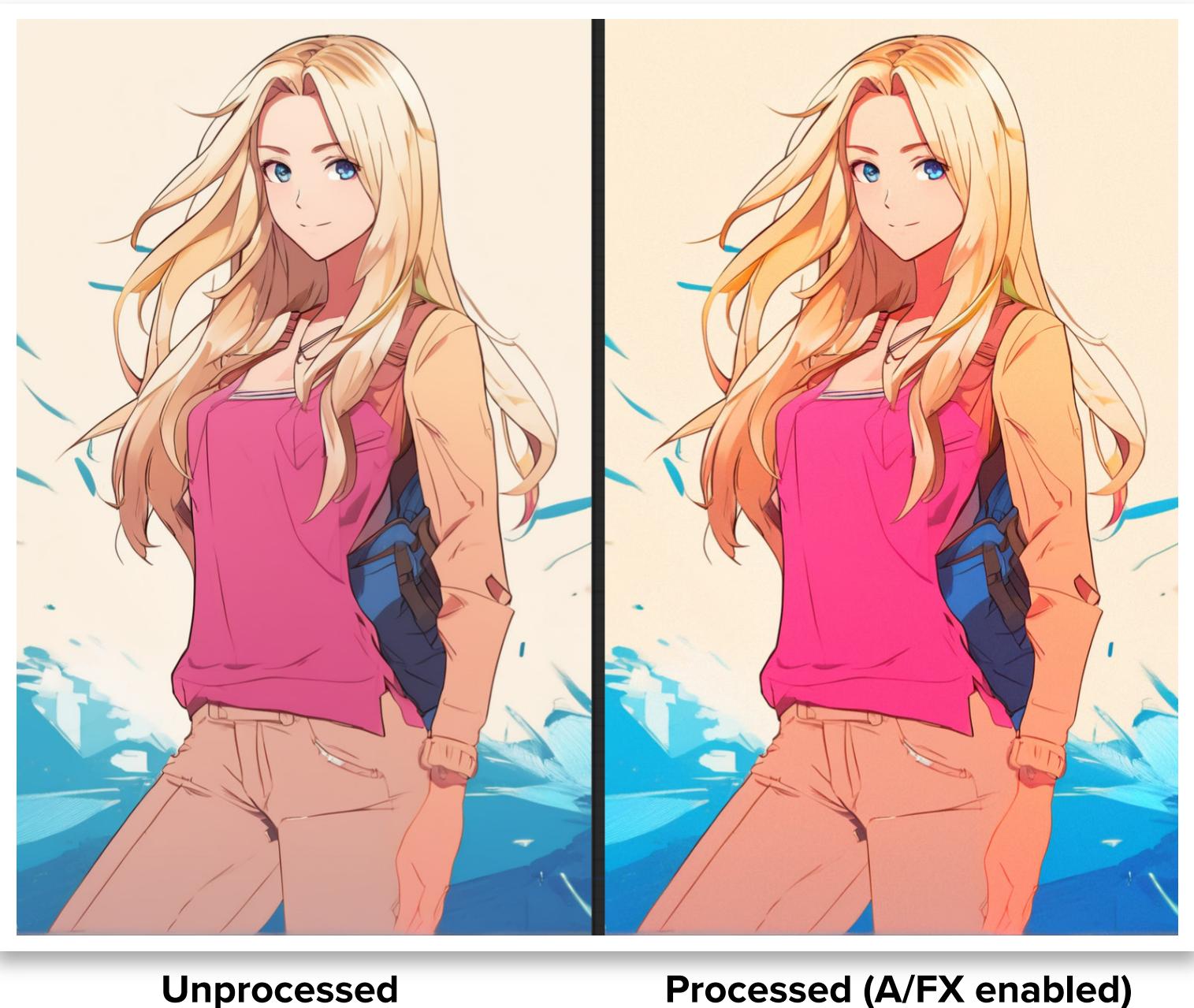
Don

Don@majorstud.io

Step 9: Adjustments & FX (Optional)

There are a number of image adjustments that can be used to tune your image after generation, similar to retouching in programs like Lightroom or Photoshop. These can be tweaked without having to remake the entire image each time, so feel free to experiment. The yellow-brown switch (1) can be set to 1 to bypass the A/FX pipeline, or set to 2 to enable it. Then, you will use the Bypassers (2) to turn on or off individual effects or adjustments.

In the example shown here, we've turned on the pipeline and enabled saturation and filmgrain to add some light noise to the image.



Adjustments

● ● 1: Bypass FX 2: Run

● 1: bypass fx 2: fx pipe enabled

INT ● 1 2 ●

● Adjustments Bypasser

- Enable Pseudo HDR Style no
- Enable Saturation yes
- Enable Brightness & Contrast no
- Enable Histogram Equalization no
- Enable CR Color Tint no
- Enable Image Levels Adjustment no
- Enable Image Remove Background (Alpha) no
- Enable Image Rembg (Remove Background) no

● Indv. FX Bybasser

- Enable Gaussian Blur no
- Enable KuwaharaBlur no
- Enable Solarize no
- Enable Image Pixelate no
- Enable Image Monitor Effects Filter no
- Enable Flatten Colors no
- Enable FilmGrain yes

● Pseudo HDR Style

● image IMAGE ●

● intensity 0.49 ●

● Saturation

● image IMAGE ●

● strength 1.50 ●

● Brightness & Contrast

● image IMAGE ●

● mode brightness ●

● strength 0.50 ●

● Histogram Equalization

● image IMAGE ●

● strength 1.00 ●

● CR Color Tint

● image IMAGE ●

● strength 0.4 ●

● mode cool ●

● Image Levels Adjustment

● image IMAGE ●

● black_level 0.0 ●

● mid_level 127.5 ●

● white_level 255.0 ●

● Image Remove Background (Alpha)

● images images ●

● mode background ●

● threshold 19 ●

● threshold_tolerance 2 ●

● Image Rembg (Remove Background)

● images images ●

● transparency false ●

● model u2net ●

● post_processing false ●

● only_mask false ●

● alpha_matting false ●

● alpha_matting_foreground_threshold 240 ●

● alpha_matting_background_threshold 10 ●

● alpha_matting_erode_size 10 ●

● background_color none ●

● Gaussian Blur

● image IMAGE ●

● strength 1.00 ●

● KuwaharaBlur

● image IMAGE ●

● blur_radius 3 ●

● method mean ●

● Solanze

● image IMAGE ●

● threshold 0.50 ●

● Image Pixelate

● images images ●

● color_palettes ●

● pixelation_size 164 ●

● num_colors 32 ●

● init_mode k-means++ ●

● max_iterations 100 ●

● dither False ●

● dither_mode FloydSteinberg ●

● color_palette_mode Brightness ●

● reverse_palette False ●

● Image Monitor Effects Filter

● image image ●

● mode TV Distortion ●

● amplitude 5 ●

● offset 10 ●

● Flatten Colors

● images images ●

● number_of_colors 5 ●

● FilmGrain

● image IMAGE ●

● intensity 0.04 ●

● scale 5 ●

● temperature 0 ●

● vignette 0 ●

Adjustments and Effects Section

Acknowledgements

As mentioned in the beginning, AegisFlow couldn't exist without the hard work of many in the Free Open Source Software (FOSS) community. The Software included with the physical drive, and/or that you obtain yourself to allow you to use this workflow, is governed by the following licenses:

GPL 3.0

https://en.wikipedia.org/wiki/GNU_General_Public_License

The MIT License

<https://opensource.org/license/mit/>

Creative OpenRail ML License

<https://www.ykilcher.com/license>

(There is some concern RE this license and its usage restrictions):

https://www.youtube.com/watch?v=W5M-dvzpzSQ&ab_channel=YannicKilcher

The concerns are mostly around “being a jerk” since usage restrictions are in place to attempt to shield Stability.AI from liability if their model is used nefariously.

Python Spftware Foundation

Help build and maintain the Python programming language
<https://www.python.org/psf-landing/>

Stability.AI

The Makers of Stable Diffusion
<https://stability.ai/>

ComfyAnonymous

The Maker of ComfyUI
<https://github.com/comfyanonymous/ComfyUI>

SargeDP

Addon maker and maker of another fine workflow
SargeSDXL
<https://github.com/SargeDP>

Cubic

The maker of Comfyui_SimpleMath
https://github.com/cubiq/ComfyUI_SimpleMath

Fannovel16

maker of comfyui_controlnet_aux
https://github.com/Fannovel16/comfyui_controlnet_aux

jamesWalker55

maker of comfyui-P2Idgan
<https://github.com/jamesWalker55/comfyui-p2Idgan>

EvanSpearman

creator of ComfyMath
<https://github.com/evanspearman/ComfyMath>

ChenyangSi

Co-Maker of FreeU
<https://github.com/ChenyangSi/FreeU>

ltdrdata

The Maker of ComfyUI Manager
<https://github.com/ltdrdata/ComfyUI-Manager>

Derfuu

Addon maker of Derfuu_ComfyUI_ModdedNodes
https://github.com/Derfuu/Derfuu_ComfyUI_ModdedNodes

pfaeff

Addon maker of pfaeff-comfyui
<https://github.com/Pfaeff/pfaeff-comfyui>

Gregor Adams

Maker of the addon “FailFast”
<https://github.com/failfa-st/failfast-comfyui-extensions>

As well as any maker that distributes addons for ComfyUI accessible in the ComfyUI manager, and artists around the world...Thank you all!

Special Thanks



Examples

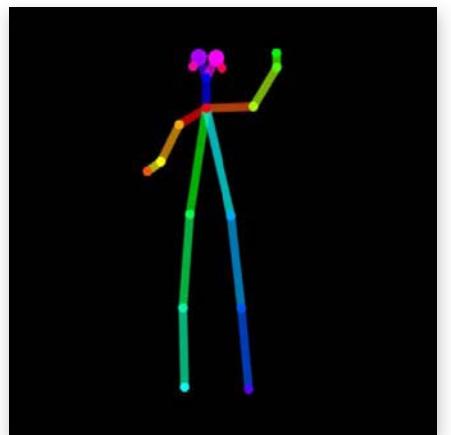
When using controlnets, the choice of what preprocessor to use (or to use one at all) is dependent on the image you are going to load into the workflow. Here's some examples of different usages of preprocessors (left) and different controlnets (right):



Premade
Depthmap,
Use 1 (Passthrough)



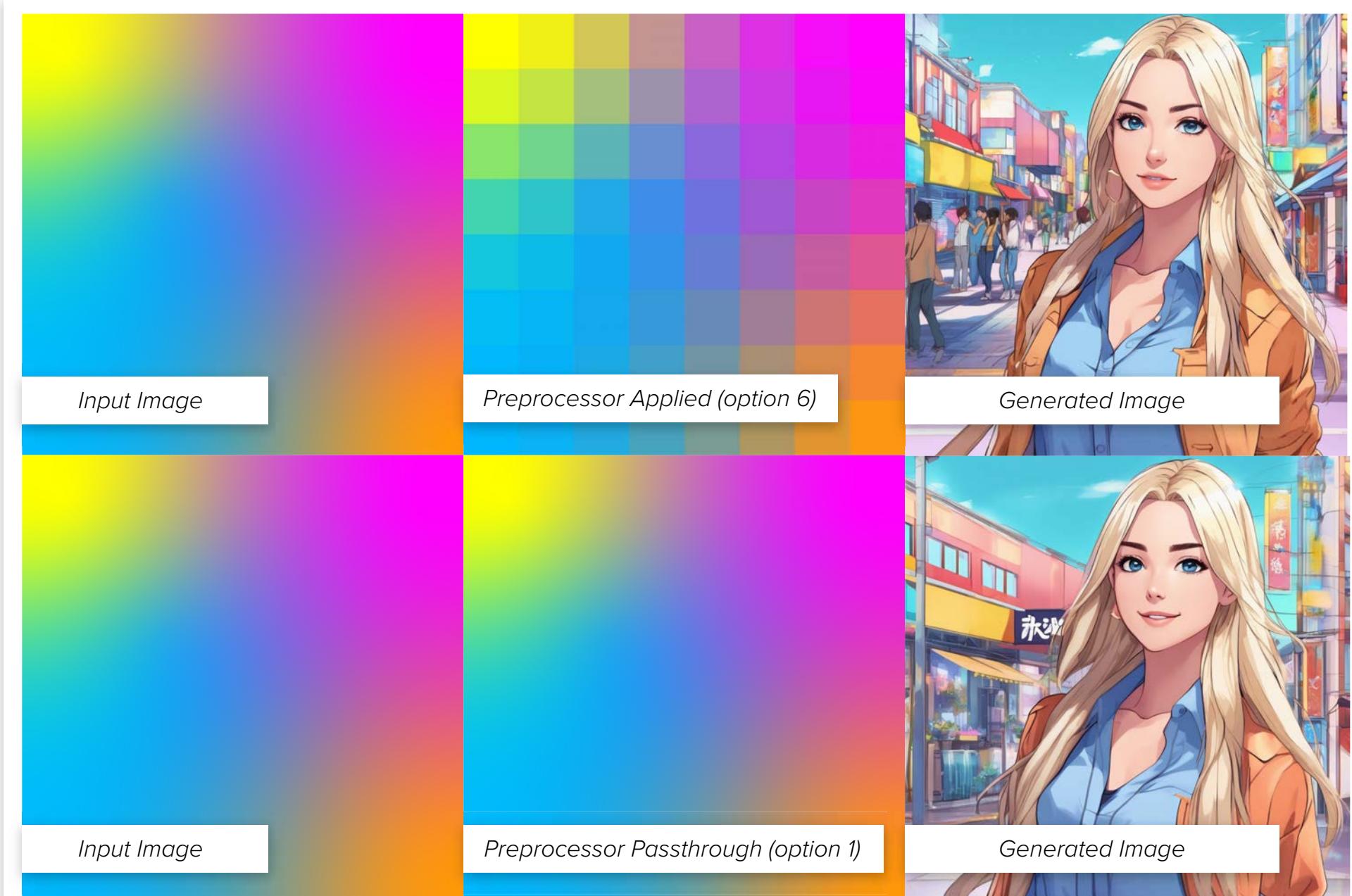
Sketch
Try 2-4, 7



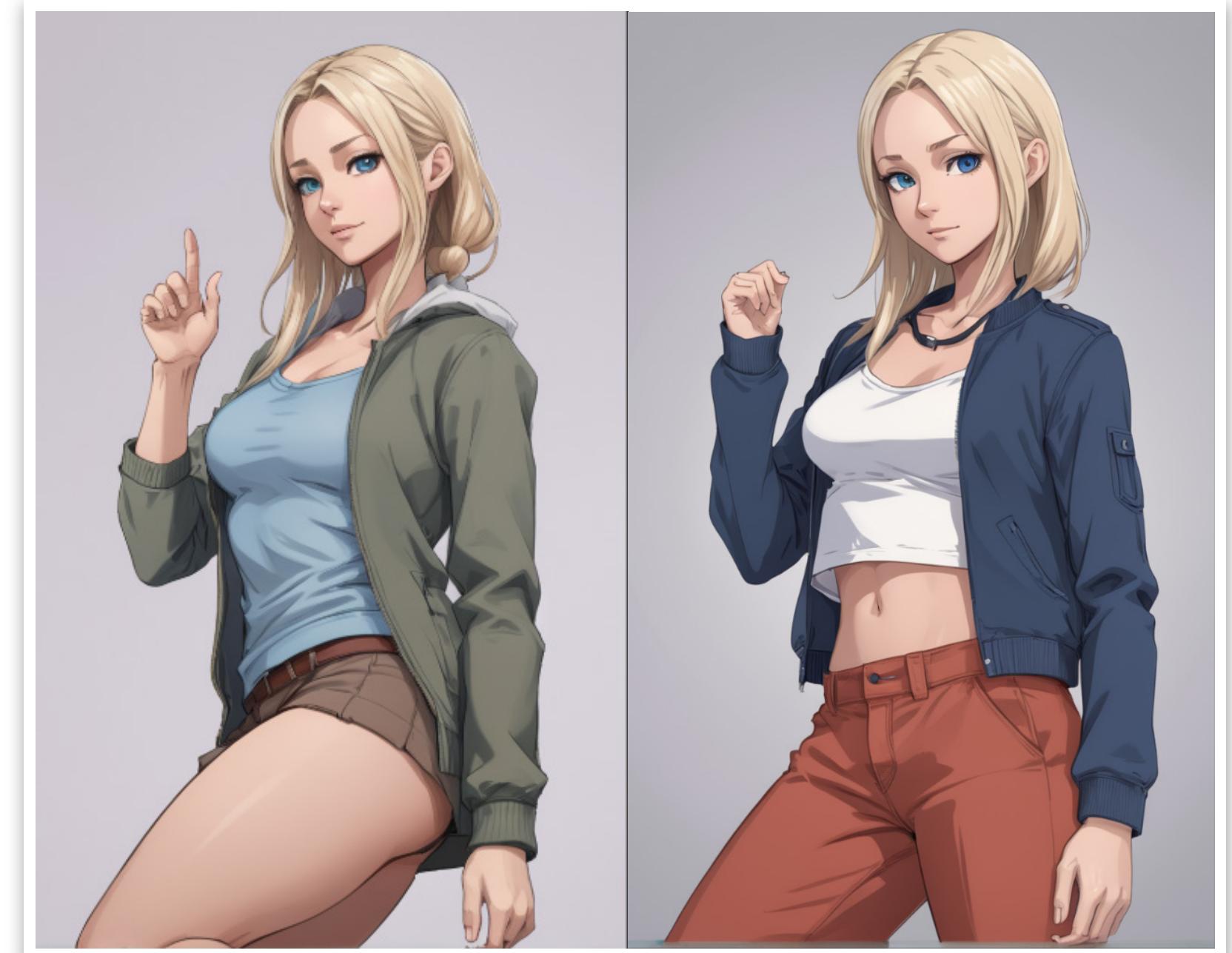
Premade
OpenPose
Use 1



Scribble
Use 9



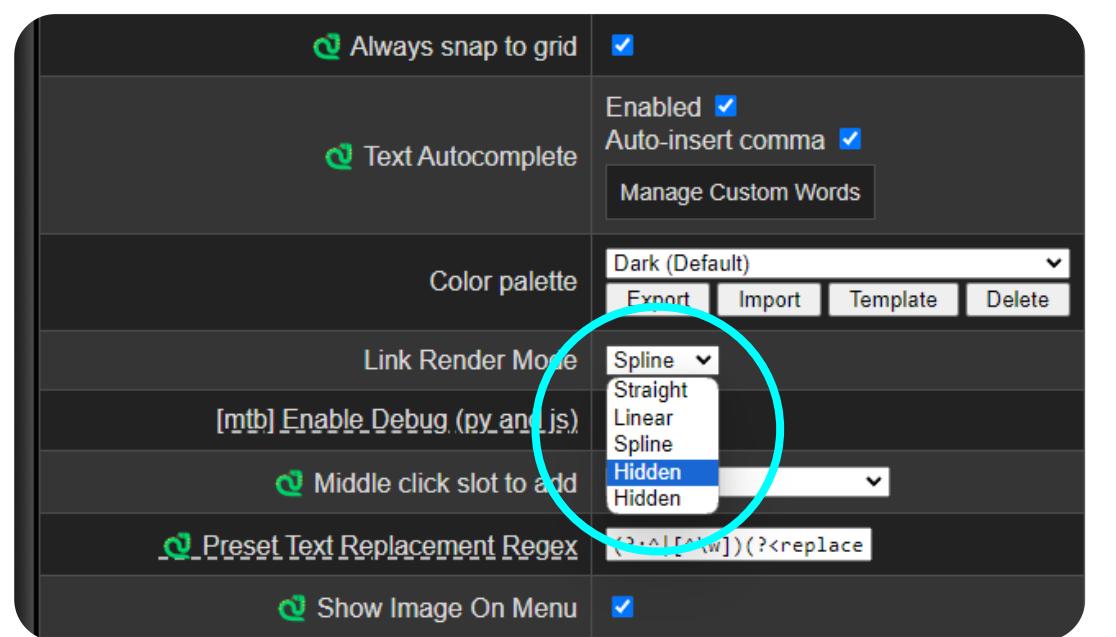
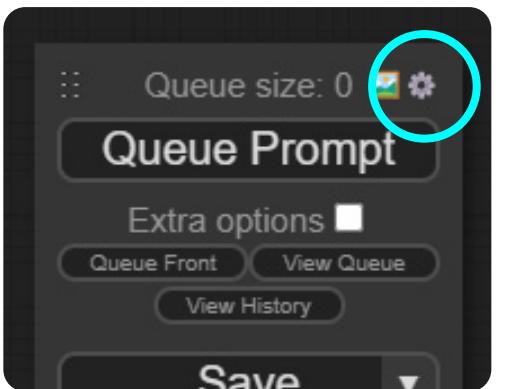
Color Input, Comparing Color (6) and Passthrough (1) Experiment.



Using Controlnet-sd-xl-1.0-softedge-dexined vs.
Control-Lora-Sketch-Rank256, all other settings identical

The wires inside the machine

Unless I am working behind the scenes, I don't have spaghetti enabled. For this reason, if you turn the node connections (spaghetti) on with the manager...



...then you'll see a mess of backend wires. They all connect to the right things, but they are definitely NOT pretty. Personally, while I am a fan of beauty, order, and the like, I think that this is a scenario where it makes less sense to strive for it.

For me, this is a tool, not an experiment, and so I tried to lay out the workflow for usability over back-end packaging. If you want to extend or change AegisFlow, you might need to first unfreeze all, unpin all nodes, and unlock all groups, and then move items around to figure out what goes to what. Have at it, but make sure you keep original backups!



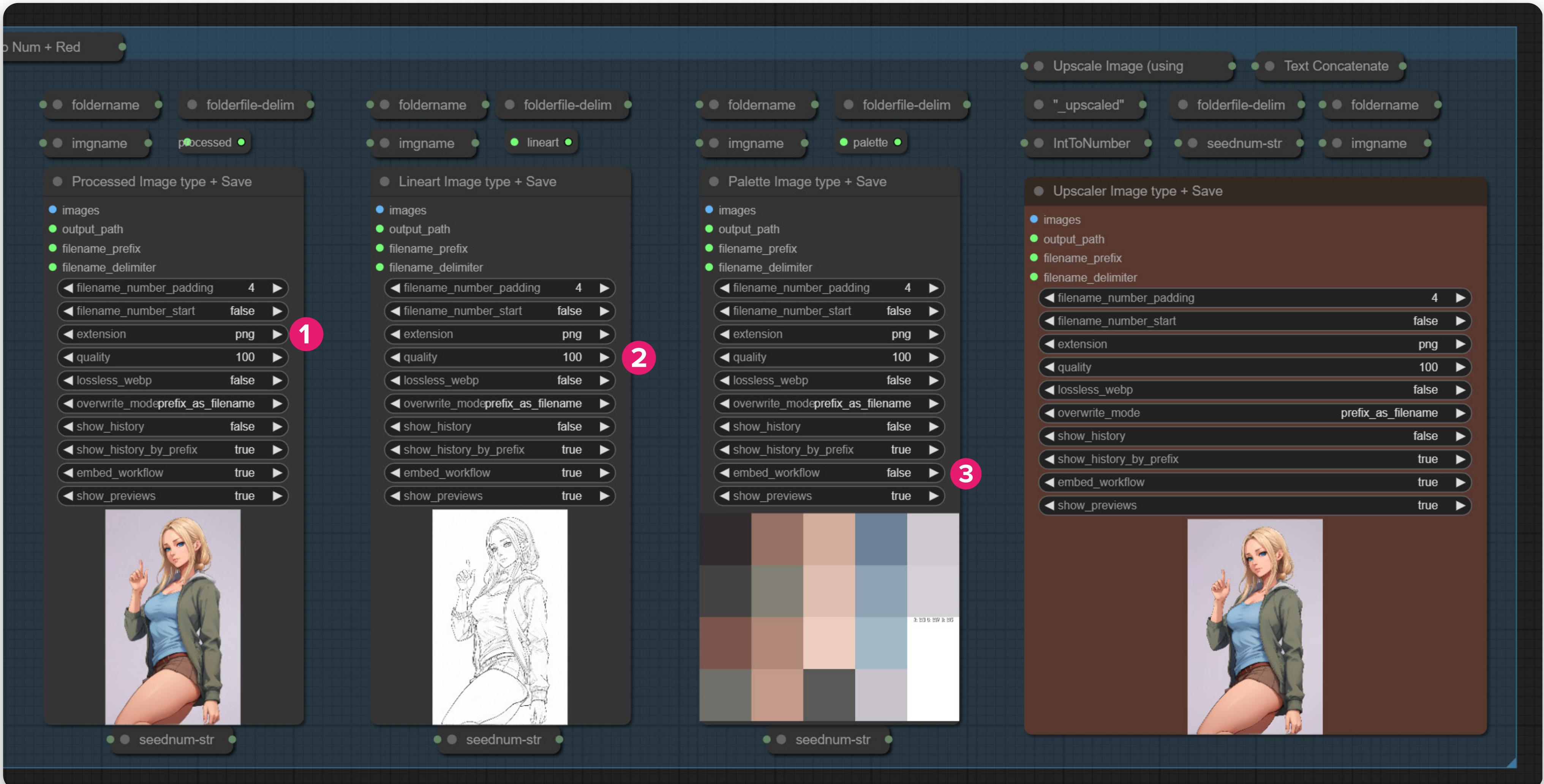
AEGIS
FLOW X
L

Appendix B: Backend information

AEGIS
FLOW

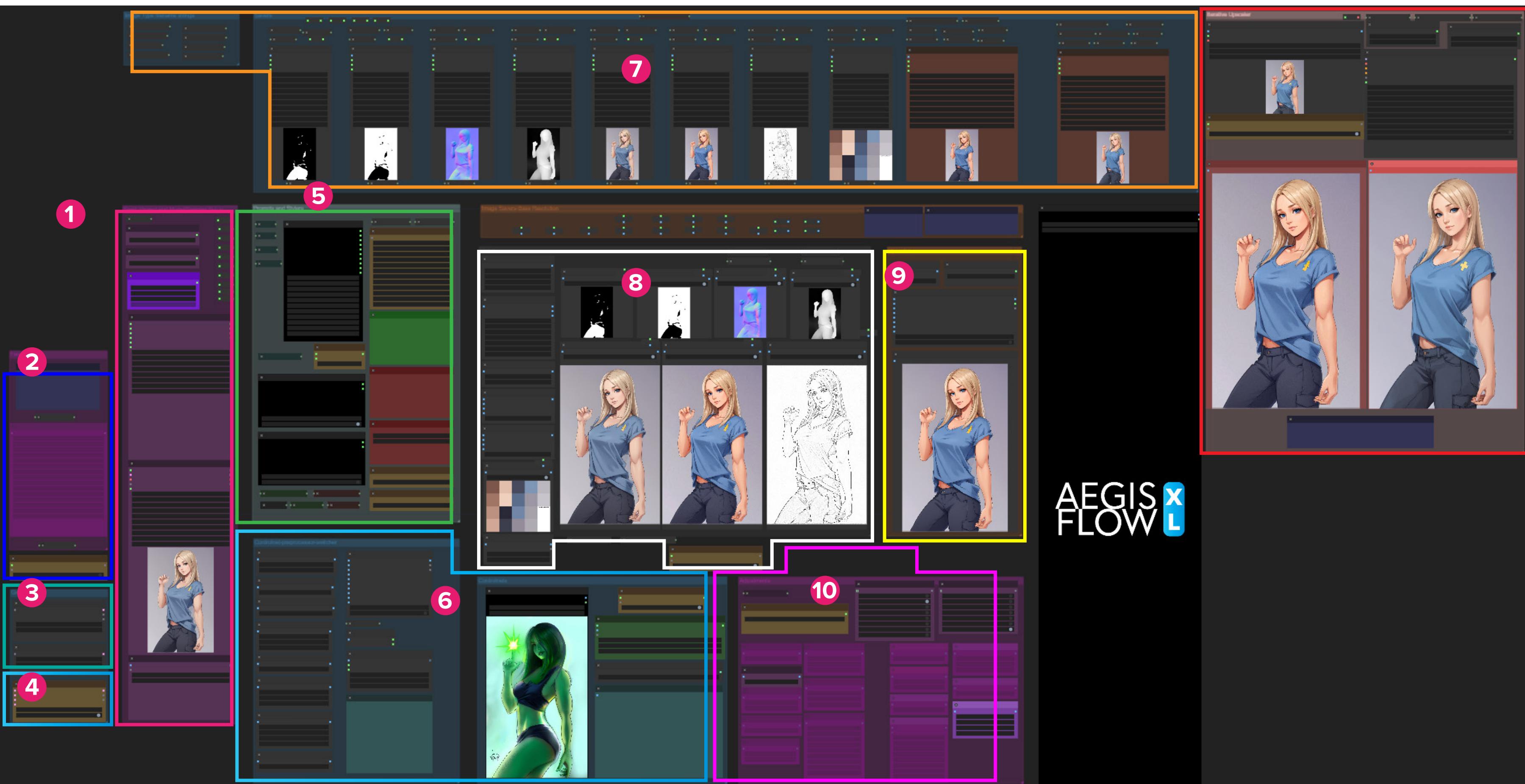
File Saving: The section you don't really need to look at much :)

The “Saving” group is at the top of the workflow. You won’t need to touch it really, unless you want to modify the following: 1) change the file type. 2) change the quality level of the exported image 3) change whether to embed the workflow in an image. There’s really not a lot to change here, but of course if you want to you’re welcome to customize AegisFlow to your liking if you have the time and skill.



Appendix C: Filesaving Area

- 1 Project Folder Setup, Seed, & Loader/Sampler
- 2 Optional FreeU settings
- 3 Alternate Latent Size options (use yellow switch)
- 4 Switch to choose latent size source
- 5 Styles and Text Prompting
- 6 ControlNet Controls and Input Image, Switch to enable or disable ControlNet Influence
- 7 Image Saving Automation
- 8 Various image settings and output previews
- 9 Upscaler section
- 10 Adjustments and Effects
- 11 Iterative Upscaler



I've got an error and one of my nodes is outlined in purple. What gives?

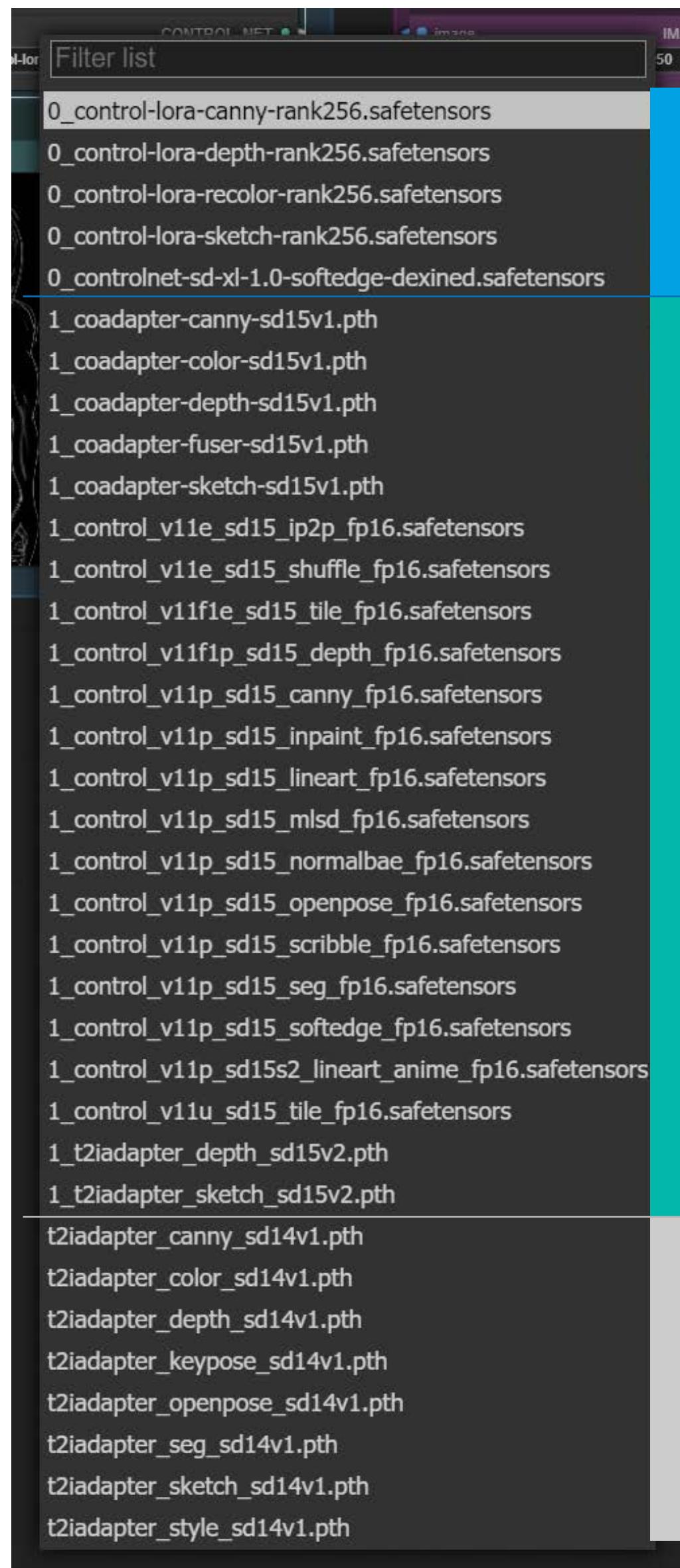
If you are using AegisFlow in its native state, that is, you haven't tried to modify it or add to it, then the most likely reason for this is one of the following:

1. You have tried to use an SDXL model in an SD 1.5 workflow, or vice-versa. You likely are seeing a "size mismatch" or a "dimension error" Because Stable Diffusion before and after XL works differently internally, things that are built for one version won't work with the other. This will lead to either:

- Errors
- An eldritch horror of an image
- complete sd lockup, requiring that you restart your ComfyUI instance.

2. This also counts for controlnet models. That means that if you try to use an SDXL controlNet inside the AegisFlow 1.5 workflow, it will fail with an error at the sampler with a "NoneType' object has no attribute" error. See the image at right for which controlnets are for 1.x and which are used for XL.

Fatal Errors on startup might indicate your computer doesn't have the capacity to run ComfyUI/SD. That is more of an issue with SD needing advanced systems than anything wrong with the workflow. Also, you may see some random warnings reported in the cmd window, but they don't always affect operation.



SDXL Compatible

SD 1.5 Compatible

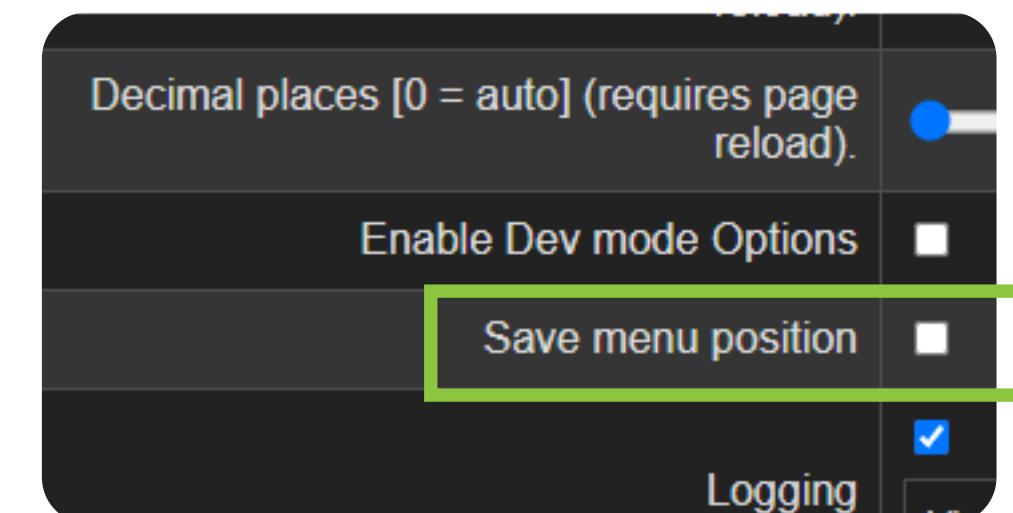
SD 1.4 Compatible

```
Error occurred when executing KSampler_Adv_(Efficient):
'NoneType' object has no attribute 'shape'
File "D:\ComfyUI\windows_portable\comfy\execution.py", line 152, in recursive_execute
    output_data = recursive_execute(input_data, input_data)
File "D:\ComfyUI\windows_portable\comfy\execution.py", line 82, in get_output_data
    return self._get_node_data(node, input_data, self.INPUT_TYPE, allow_interrupt=True)
File "D:\ComfyUI\windows_portable\comfy\execution.py", line 100, in _get_node_data
    result = self._execute_node(node, input_data, self.INPUT_TYPE, allow_interrupt=True)
results.append(extract_obj(node, func(**self._dict(input_data, 1)))
```

Scary errors like this are often caused by selecting and SDXL controlnet when a 1.5 model is being used. Another common cause is a missing resource (a missing model, controlNet, LoRA or similar.)

Missing manager menu

If the manager menu sometimes disappears. I've discovered this is due to the menu being positioned in CSS offscreen. By default, ComfyUI places the manager on the right side of the screen, but then the manager position is saved and is maybe saved in a place that becomes inaccessible in a new window. I've found the solution to this issue is to clear this checkbox:



If your manager is already lost, increase browser magnification until you see it again, then clear this box in the manager settings. Reload your ComfyUI browser windows and you should see the menu back on the right side.

Lastly, if a generation takes an insane amount of time and seems hung, first check the cmd window to see if anything is downloading. If there's no activity, consider restarting ComfyUI and trying again. Sometimes this just happens, although it is pretty rare.

I do this for fun, but my family eats to stay alive.

For those that want to help out, and can, I would love to see some ongoing support via Gumroad's subscription feature.

In exchange, I will post new WIP builds of AegisFlow to my Patreon page for members to try out prior to them being released via major updates, and you'll be able to help guide its development with feature suggestions, etc.

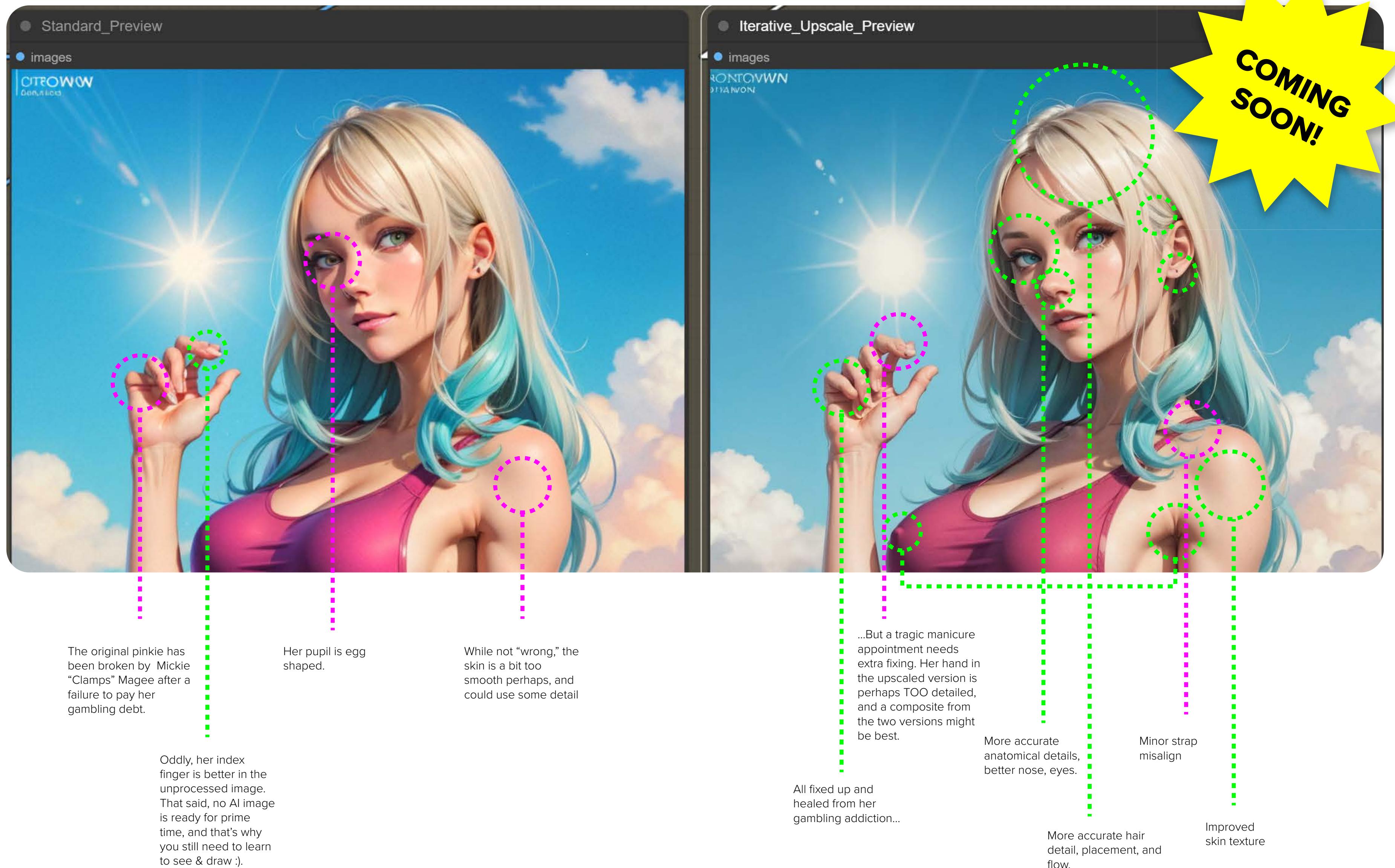
I'm a professional working artist with over 25 years in the design industry (primarily graphic design) for companies like HBO, the Department of Defense, Microsoft and numerous tech companies. If that experience is helpful, to you I can answer your design-related questions (and probably others) in the members area.

I'll also place art (some AI related, many not) in the members section that you can download and learn from.

I'm not looking for huge commitments here—there is no \$500/month **MEGASUPPORTER** tier that allows you to rename my kids—Just \$1 and \$5 based on your desire and ability.

At right, this is an iterative upscaler built into a version I am working on right now. Check out the (mostly) improved image!

<https://majorstudio.gumroad.com/l/members>



Appendix F: Become a member for early access additions to AegisFlow



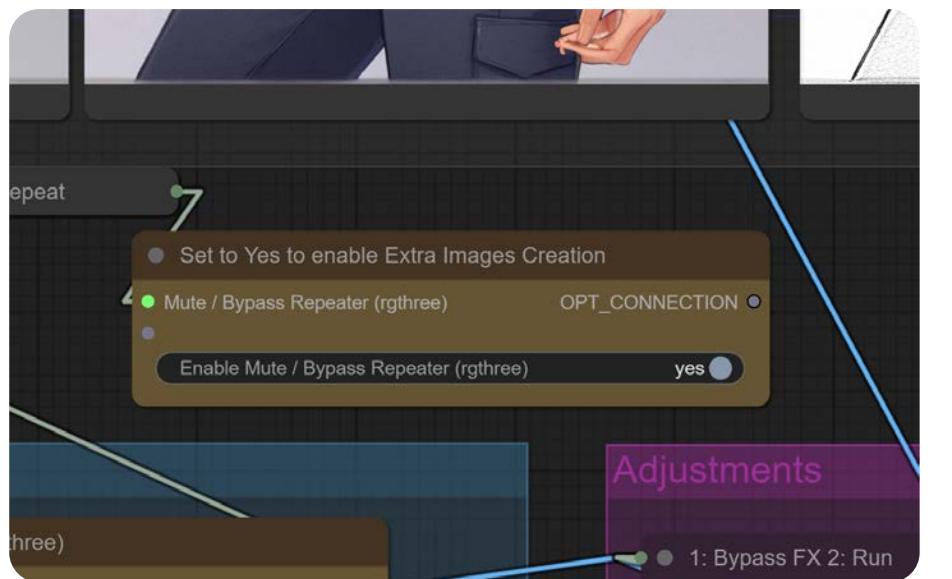
MAJOR
S T U D . I O

Step 7: Extra Images & Iterative Upscale

Extra Images toggle switch

Added into v1.3, There is a new switch that can turn off extra image generation and also a switchable Iterative upscaler that adds detail and generally improves upscaled images. As an example, you might wish to initially disable both of these so that you can get a seed and settings setup that looks good, then re-enable them and re-queue (with the same seed and settings) to get the extra images from your chosen image.

This switch is located just below the “Processed” image preview and above the adjustment and controlnet areas.



Iterative Upscale

The iterative upscaler is a special upscaling workflow that takes your standard upscaled image and then applies a multi-stage upscale to improve image fidelity.

1. The Upscale Factor is the multiplier for the final image size.
2. How many steps it will use to get to that scale.
3. Enables or disabled the Iterative Upscale and its image saving node
4. Choses between using the original + conditioning (1) or adding in the additional conditioning(2) from 6.
5. Choses between using the original - conditioning (1) or adding in the additional conditioning(2) from 7.
6. Additional Positive conditioning
7. Additional Negative Conditioning
8. Provides a hook that gradually changes the denoise to target_denoise as the step progresses. (from Comfy Impact Pack documentation)
9. Provides a hook that gradually changes the cfg to target_cfg as the step progresses. (from Comfy Impact Pack documentation)
10. An upscaler is provided that converts latent to pixels using VAEDecode, performs upscaling, converts back to latent using VAEEncode, and then performs k-sampling. This upscaler can be attached to nodes such as 'Iterative Upscale' for use. (from Comfy Impact Pack documentation)
11. Original Preview (left) and iterative preview (right)

Enable Extra Image Switch and Iterative Upscale

