

Recommended Hardware

NVIDIA GPU >11GB VRAM

Dependencies

A1111/Forge

Download and extract this archive somewhere:

https://github.com/lllyasviel/stable-diffusion-webui-forge/releases/download/latest/webui_forge_cu121_torch231.7z

Best to place this near the root of a drive letter, no long paths, no spaces. Though this software should tolerate such things, it is not recommended to count on that.

Models

Then download these files, and place in the 'webui\models\Stable-diffusion' subdirectory.

noobaiXLNAIXL_epsilonPred05Version.safetensors

From either of these links (HuggingFace should not require login to download):

https://huggingface.co/nnnn1111/models/resolve/main/noobaiXLNAIXL_epsilonPred05Version.safetensors

<https://civitai.com/models/833294?modelVersionId=968495>

animagine-xl-3.1.safetensors

<https://huggingface.co/cagliostrolab/animagine-xl-3.1/resolve/main/animagine-xl-3.1.safetensors>

LORA/LYCORIS

Now download this file and place in the 'webui\models\Lora' subdirectory (yes, you may have to create the directory).

FEGBA-Sprites-illus_Fp.safetensors

<https://civitai.com/models/949388/fire-emblem-gba-sprite-style-illustrious?modelVersionId=1062945>

Usage – Update and Run the WebUI

Double click the relevant batch files, in order. You should get a web browser tab opened to the A1111/Forge interface.

update.bat

run.bat

Immediately click the ‘XL’ or ‘SDXL’ UI option in the top-left corner. This will give additional settings for SDXL models.

Usage – Resolution

Resolution is taken as a cue much stronger than text prompts. If you use a resolution matching images the community tagged as photos, you will usually get photorealistic results.

For clip art and icons, the recommended resolutions to generate are:

1024x1024

Usage – Terms

To get good images, you will need to add appropriate tags, as well as a detailed ‘natural language’ description of your object, to craft a text prompt.

The original SDXL 1.0 base model was trained on pairs of images and natural language captions. From there, the community has curated pairs of images and captions of ordered comma delimited ‘tag’ keywords de-facto standardized as ‘Danbooru’ tags.

Thus, these community models can be directed to combine very specific results from using the standardized tag keywords in the standardized order, yet also mix in some natural language concepts, such as relatively obscure objects and devices.

Some of these de-facto standardized tags/keywords relevant to clip art and icons, in the standardized order, are available at:

https://github.com/mirage335-special/iconArt/blob/main/_doc/usage-imageGeneration/terms.txt

Recommended parameters are also included.

Usage – Automatic Generation of ‘Natural Language’ Descriptive Terms

To better describe the objects you want to include in your clip art, such as historical style, an LLM assist prompt is available at:

https://github.com/mirage335-special/iconArt/blob/main/_doc/usage-imageGeneration/LLM_assist_prompt.txt

This prompt has been tested with a derivative of the Llama 3.1 8b LLM model with excellent results, but may also be usable enough with ChatGPT, etc.

Usage - Models

For clip art, use animagine-xl-3.1 .

For icons, use noobaiXLNAIXL_epsilonPred05Version with the FEGBA-Sprites-illus_Fp LORA/LYCORIS .

Design - How it Works – Low Bits, Sampler, Schedule, Steps, CFG, Batch

Stable Diffusion models begin with a random noise image. Yes, as dense a mix as you can get of random dots of red, green, blue, and everything in between.

That random noise is refined to get closer and closer to the concepts that are associated with the text prompt, based on the training from pairs of images, and captions. In the case of modern community models, additional training has been done with a de-facto standardized strict but vast set of keywords/tags, to improve control.

Low Bits

Less precise arithmetic can be more ‘creative’, overfitting less, and introducing more obscure concepts. More precise arithmetic can better control such defects as incorrect numbers of fingers. Diffusion in Low Bits setting controls the data type – integer, floating point, etc.

This cannot be taken for granted, there is not a ‘best’ answer, and not always a ‘these are better for this purpose’. It is often necessary to generate images from different arithmetic precisions to get a decent selection to choose from.

Fortunately, there are only a few useful options, all of them fp16 LORA . Lower precision LORA is of no known benefit and should never be used.

bnp-fp4 (fp16 LoRA)

fp8_e5m2 (fp16 LoRA)

fp8_e4m3fn (fp16 LoRA)

Sampler, Schedule, Steps

Sampler

Sampler algorithms are **either ancestral** (deriving an image from many previous image steps, *plural*), **or not** (deriving an image by refining only one previous image step, *singular*).

Most of the sampler algorithms available are very obsolete, kept around to recreate and modify historical examples of good images. Most others generate very slightly different, but more often defective, images (eg. duplicate fingers), or just randomly different results (ie. as if a different random seed had been used).

Fortunately, there are just two useful samplers, one ancestral, one not. An **ancestral sampler is usually better**, but can require more steps, and not all models are compatible with an ancestral sampler.

Euler a (ancestral)

DPM++ 2M SDE

Schedule, Steps

From one image step to the next, noise can be added, changed, etc, in varying amounts. Whether more noise is added at later steps, earlier steps, middle steps, or some more complicated mixing algorithm is used, depends on the schedule.

Turbo – Minimizes the number of steps needed (as **few as 12 steps**), saving time generating many images. Useful to **quickly test improving the text prompt** or generating many candidate images to select a few good images.

Karras – Often decent images with a small number of steps (maybe 20-40 steps).

Polyexponential – Can generate the best images with very large number of steps (120-150 steps). Computationally intense, slow, but **often very worthwhile in difficult cases**. Let the computer do the work for you if your prompt is already good.

CFG – Classifier Free Guidance

Higher numbers, towards 7 and greater, creates images following the prompt much more strictly, but can also cause such issues as less variety of concepts, and very bad color saturation. Lower numbers, towards 5 or less, can create images more likely to ignore instructions entirely.

Models can differ somewhat in the recommended CFG value range. Usually 5-7 is recommended for most models.

With a good prompt, 7 is usually best.

Batch

Increasing batch size above 1, can usually improve parallelism, better saturating the capabilities of the GPU, producing more images more quickly.

Multiple batches can be used to increase the total number of images generated beyond that.

Alternatives

Research Engine

A Large-Language-Model project similarly intended for development purposes exists, adding an automatically installed 'Llama-augment' model, which can be much more robust, as well as OpenWebUI and a metasearch engine usable by the AI.

For MSWindows, installation is very straightforward, including installing 'extendedInterface' with the 'extlface.exe' installer, instructions are available at:

<https://github.com/mirage335-special/kit-researchEngine/blob/88c97d734e66e85e8c8c42946d1ba3fbb4251e95/Overview.txt>

For Unix/Linux, the installation is automatic, available as part of ubdist/OS .

<https://github.com/soaringDistributions/ubDistBuild>

Scripts to automate this installation are in the neighboring subdirectories along with the MSWindows installation instructions.

Forge Fork

It is possible to splice a directory from a fork of A1111/Forge, to re-enable video and 3D object generation from images.

<https://github.com/Panchovix/stable-diffusion-webui-reForge>

While the results are very impressive, this can be set up as standalone server, and online AI video generation services may be even more impressive. Just getting good images and maybe inpainting is already the tricky part of the process, so it is more reasonable to use something more automatic for subsequent steps.

Download Scripts

Downloading the needed files can be done more automatically, from GitHub repository and release files, using the scripts from the iconArt repository. This works with the MSWindows platform as well

if 'extendedInterface' is already installed – use the 'extlface.exe' file from the 'internal' release to install.

<https://github.com/mirage335-colossus/extendedInterface/releases>

<https://github.com/mirage335-special/iconArt>

_fetch_iconArt.bat

These scripts, and possibly also the model files with Stable Diffusion software, are normally already included with ubdist/OS , available as a raw disk/USB image, a bootable BD/USB ISO, and as an automatically installable MSW/WSL2 filesystem.

<https://github.com/soaringDistributions/ubDistBuild>

PNG to JPG Converter

Most image conversion software will NOT preserve the metadata saved with the PNG images.

An image conversion tool is available at:

https://github.com/mirage335-special/iconArt/tree/main/_lib/png_jpg_convert

For MSWindows, 'extendedInterface', installed by an 'internal' release of the 'extlface.exe' installer, includes the necessary python dependencies already.

<https://github.com/mirage335-colossus/extendedInterface/releases>

Right click on the directory with the images to convert, 'Open with ubcp-bash', and use the python script appropriately. This script may be included in PATH with future 'extendedInterface' builds.

Updated A1111/Forge

Packages that have already been 'run' on various computer platforms (ie. certain GPUs, specific laptops, etc), may be available. These may or may not be useful for such purposes as 'LiveISO' distribution, and to preserve the needed python dependencies from upstream breakage.

<https://github.com/soaringDistributions/iconArt-build/releases/>