

This guide will walk you through setting up and running GFPGAN, with some optional automation at the end.

Requirements:

Python ≥ 3.7

PyTorch ≥ 1.7

Recommended:

Nvidia card with CUDA cores (note step 1)

Anaconda Python (Alternative: Miniconda)

Windows (for automation)

Pride

A heart of stone

Step 1. Installing Python/PyTorch

(If you already have a functioning setup of Python/PyTorch you can skip this step. Please note that I will be using Anaconda for my Python install, so some of the references in later steps might not exactly match if you use a different install.)

Download [Anaconda](#) and install it. Make sure you note your installation directory when you do so, as this will be handy later.

Run the Anaconda prompt as administrator and install [PyTorch](#) by entering one of the following commands. If you use the second command, it will work for all systems, but won't leverage CUDA cores and will run using vulkan, which will result in much slower performance if you have a GPU with CUDA cores. If you have an Nvidia GPU that isn't archaic, you probably have CUDA cores and should use the first one.

- CUDA GPUs¹ should use `pip3 install torch torchvision torchaudio --extra-index-url https://download.pytorch.org/whl/cu116`.
- Otherwise use `pip3 install torch torchvision torchaudio`.

Step 2. Install Git

You can install Git for Anaconda using the following command:

```
conda install -c anaconda git
```

¹ <https://developer.nvidia.com/cuda-gpus>

Step 3. Clone GFPGAN

Open the Anaconda prompt and move to the folder that you want to save GFPGAN in.

Below is an example of changing drive (in this case, d: to move to the D: drive) and folder (cd testing stuff).

A screenshot of the Anaconda Prompt window. The title bar reads "Administrator: Anaconda Prompt (Anaconda3)". The command history shows: (base) C:\WINDOWS\system32>d: (base) D:\>cd "testing stuff" (base) D:\testing stuff>.

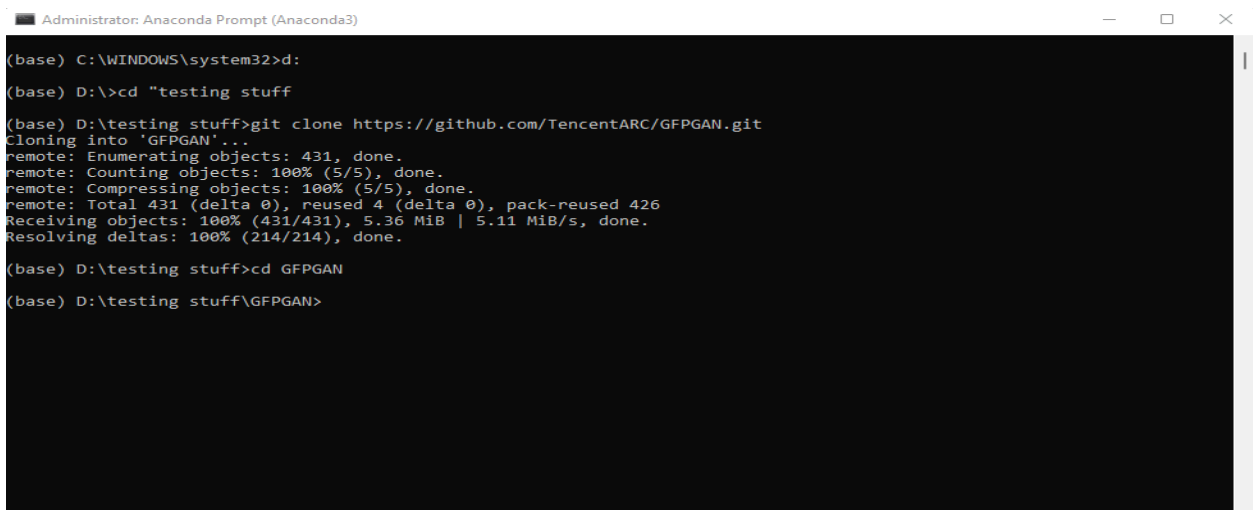
```
(base) C:\WINDOWS\system32>d:
(base) D:\>cd "testing stuff"
(base) D:\testing stuff>
```

Clone the git repo using the following command:

```
git clone https://github.com/TencentARC/GFPGAN.git2
```

Use the following command to enter the GFPGAN folder (regardless of where you installed it):

```
cd GFPGAN
```

A screenshot of the Anaconda Prompt window. The title bar reads "Administrator: Anaconda Prompt (Anaconda3)". The command history shows: (base) C:\WINDOWS\system32>d: (base) D:\>cd "testing stuff" (base) D:\testing stuff>git clone https://github.com/TencentARC/GFPGAN.git (base) D:\testing stuff>cd GFPGAN (base) D:\testing stuff\GFPGAN>.

```
(base) C:\WINDOWS\system32>d:
(base) D:\>cd "testing stuff"
(base) D:\testing stuff>git clone https://github.com/TencentARC/GFPGAN.git
Cloning into 'GFPGAN'...
remote: Enumerating objects: 431, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 431 (delta 0), reused 4 (delta 0), pack-reused 426
Receiving objects: 100% (431/431), 5.36 MiB | 5.11 MiB/s, done.
Resolving deltas: 100% (214/214), done.
(base) D:\testing stuff>cd GFPGAN
(base) D:\testing stuff\GFPGAN>
```

² You will need to authenticate with GitHub to do this. If you have not cloned a Git repository before, it will pop up a window and ask you to authenticate. I recommend using the browser authentication option. You may need to create an account but once that's done you can re-run the command while signed in to automatically authenticate using your browser.

Step 4. Installing prerequisites and developing GFPGAN

GFPGAN has prerequisites that can be installed using the following commands.^{3 4}

```
pip install basicsr
pip install facexlib
pip install realesrgan
pip install -r requirements.txt
```

We also need to develop GFPGAN, but that's as easy as running setup.py.

```
python setup.py develop
```

Step 5. Installing models

You must also install models and save them to the .../GFPGAN/experiments/pretrained_models folder. There are two ways to do this.

- Download the models manually and move the file into the models folder manually.
[V1](#) (optional), [V1.2](#) (optional), [V1.3](#)⁵
- Use the following command to download the v1.3 model:

```
wget
https://github.com/TencentARC/GFPGAN/releases/download/v1.3.0/GFPGANv1.3.
pth -P experiments/pretrained_models
```

Step 6. Running GFPGAN

Now that model(s) setup is complete, you can run GFPGAN. The command to run GFPGAN is as follows.⁶

```
python inference_gfpgan.py -i inputs/whole_imgs -o results -v 1.3 -s 2
```

Here's what that command does:

python - tells the prompt to run using python- this is because inference_gfpgan.py is a python script, so it tells it to run that script with python.

Inference_gfpgan.py - tells python to run that script

-i designates the input folder- in theory you could change this (for example if you wanted to grab all images in a different folder) but for now let's leave it as inputs/whole_imgs

³ realesrgan is used to upscale backgrounds and is optional but recommended

⁴ Note that you will need to be in the GFPGAN folder to use pip install -r requirements.txt. If you have closed the Anaconda prompt between the last step and now, you will need to navigate back to the GFPGAN folder.

⁵ To compare the models, see <https://github.com/TencentARC/GFPGAN/blob/master/Comparisons.md>. Note that if you do not install V1.3 you can still use GFPGAN, but since V1.3 is the default model, it will require you to always specify the model.

⁶ Note you will need to run this from the Anaconda Prompt or activate python to do this.

-o designates the output folder, which I recommend leaving as results. With the default path of “results” it will put the completed restored files into .../GFPGAN/results/restored images. If that folder doesn’t exist, don’t worry- it will be created when you run it.

-v specifies the model to be used- 1.3 is the default so this portion is optional.

-s specifies the upsampling scale- basically, it will double the size of the sampled section (and, if realesrgan is installed, the rest of the image as well).

There are more options,⁷ but these are the ones you need to know.

If you place an image in the .../GFPGAN/inputs/whole_imgs folder, it will process when you run the command above and automatically place the completed file in the

.../GFPGAN/results/restored images folder.

GFPGAN switches according to their Git repository for reference purposes:

-h	show this help
-i input	Input image or folder. Default: inputs/whole_imgs
-o output	Output folder. Default: results
-v version	GFPGAN model version. Option: 1 1.2 1.3. Default: 1.3
-s upscale	The final upsampling scale of the image. Default: 2
--bg_upsampler	background upsampler. Default: realesrgan ⁸
--bg_tile	Tile size for background sampler, 0 for no tile during testing. Default: 400
--suffix	Suffix of the restored faces
--only_center_face	Only restore the center face
--aligned	Input are aligned faces
--ext	Image extension. Options: auto jpg png, auto means using the same extension as inputs. Default: auto

⁷ <https://github.com/TencentARC/GFPGAN>

⁸ It appears to use RealESRGAN_x2plus for the model. If it hangs while downloading this, try running your prompt as administrator or manually installing the file.

Automation⁹

Now, that's great, but you still need to open the Anaconda prompt each time, navigate to your GFPGAN folder, and then run that command (or whatever command you want, such as for a different model of GFPGAN) to process your images, and that's a lot of work. What if we could, you know, just click a button and make it work? Well, with the wonders of a .bat file, you can.

Note that the final step of the process, while commented, is destructive, and if you're anxious about that, you can delete everything below the step that actually runs GFPGAN.

Open Notepad

Paste in the following and update paths in <> accordingly:¹⁰

```
call <wherever you installed Anaconda to in step
1>\Anaconda3\condabin\activate.bat
:: This will make it so that the command to run GFPGAN doesn't say "Where is
python? I don't know what python is. I'm going to die now."
:: uncomment the next line if you installed on a drive other than your system
drive
:: <driveletter>:
:: if you're running the .bat from anywhere *BUT* your GFPGAN folder, you will
need to uncomment the next step and make them navigate to where you put GFPGAN.
If you're running from your GFPGAN folder, I'd still use these steps just to be
sure that it doesn't try to run in your Anaconda install folder and throw a
fit.
:: cd "<path to folder where GFPGAN is located>"
python inference_gfpgan.py -i inputs/whole_imgs -o results -v 1.3 -s 2
::Uncomment the three below lines to automatically delete the input images. If
you only use .pngs or .jpgs, you can delete the other line without issue. Note
that this is a destructive step so you may want to leave this commented, just
remember to remove the input files before running again or it will reprocess
them.
::cd "<where you put GFPGAN>\inputs\whole_imgs"
::forfiles /M *.png /C "cmd /c del @file"
::forfiles /M *.jpg /C "cmd /c del @file"
```

Save the .bat file as something you will be able to recognize.

When you click on this .bat, it will automatically process the files from the GFPGAN input folder and put them in the standard output folders, so it's basically just skipping the step of manually telling it to do what you did in step 6.

⁹ Since I use a .bat you will need to use Windows or modify appropriately for your OS. forfiles is also only going to work in CMD/powershell on Windows, to my knowledge.

¹⁰ So, for example, for "<path to folder where GFPGAN is located>", you would use something like "D:\gfpgan\GFPGAN"