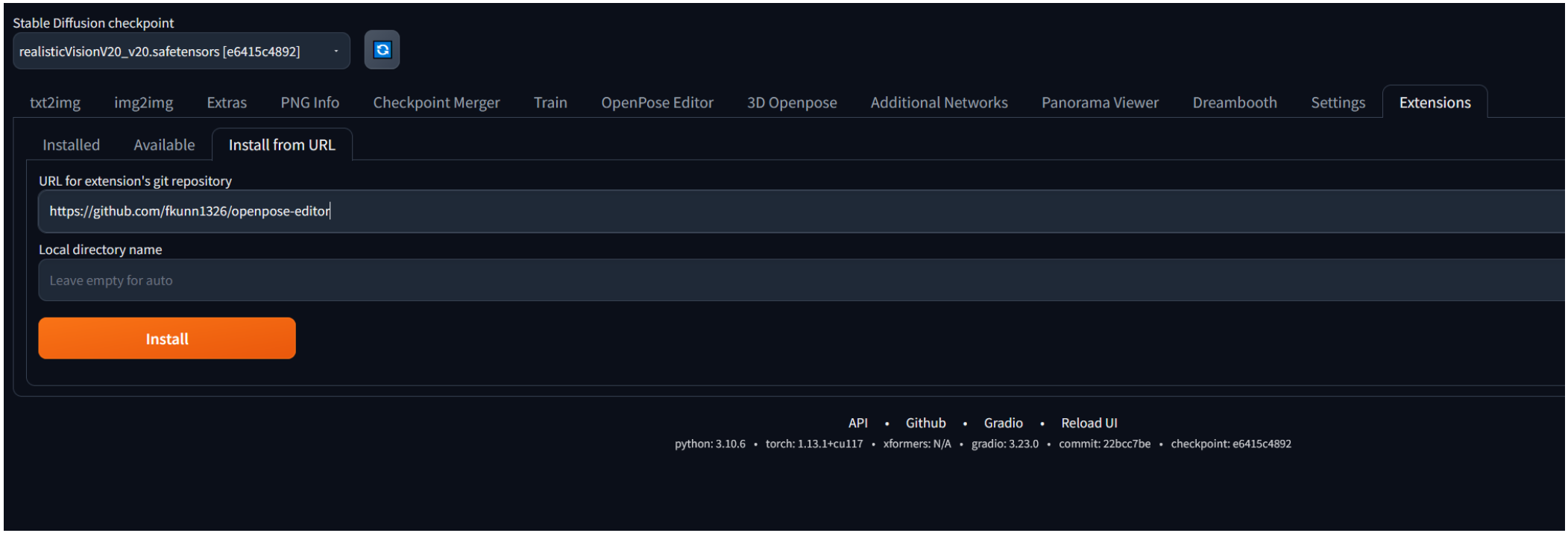
# Install Open Pose Editor:

Paste this Link „ *https://github.com/fkunn1326/openpose-editor* „ under Extras into install from URL




Then click on „check for Update“ and restart UI.
there is also a more complex 3d openpose mode, but the one shown above is more than enough.

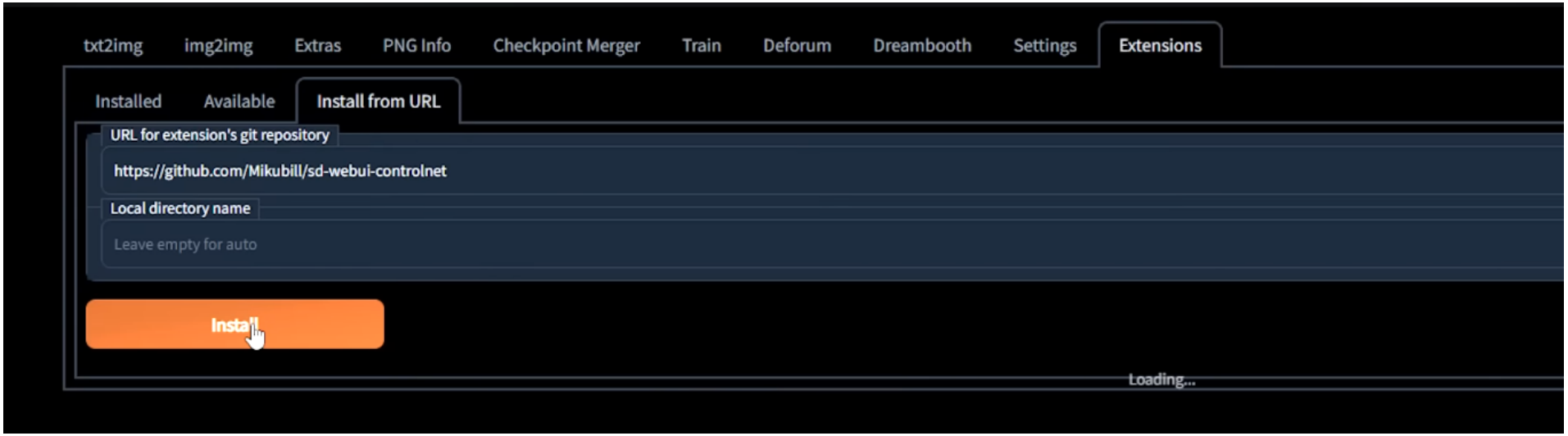
Fast Tutorial:
https://www.youtube.com/watch?v=5z71oxf8kh4

# Installing Control Net

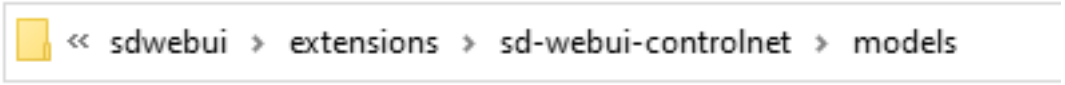
First download the models found following the link:
https://civitai.com/models/9251?modelVersionId=11004
Open the Comand Window (type cmd) and after locating your SDWEBUI Folder type:
*pip install opencv-python*

```
C:\sdwebui>pip install opencv-python
```

Then you have to install thsi link from URL : https://github.com/Mikubill/sd-webui-controlnet




then“Apply and Restart UI“
Afterwards copy all your Control net files you downloaded ind the beginning into this Folder



## Combine 2 Control Nets

https://www.youtube.com/watch?v=cNIHZInV3mg&t=223s

## Open source text Ai:

https://www.youtube.com/watch?v=w41-MUfxIWo

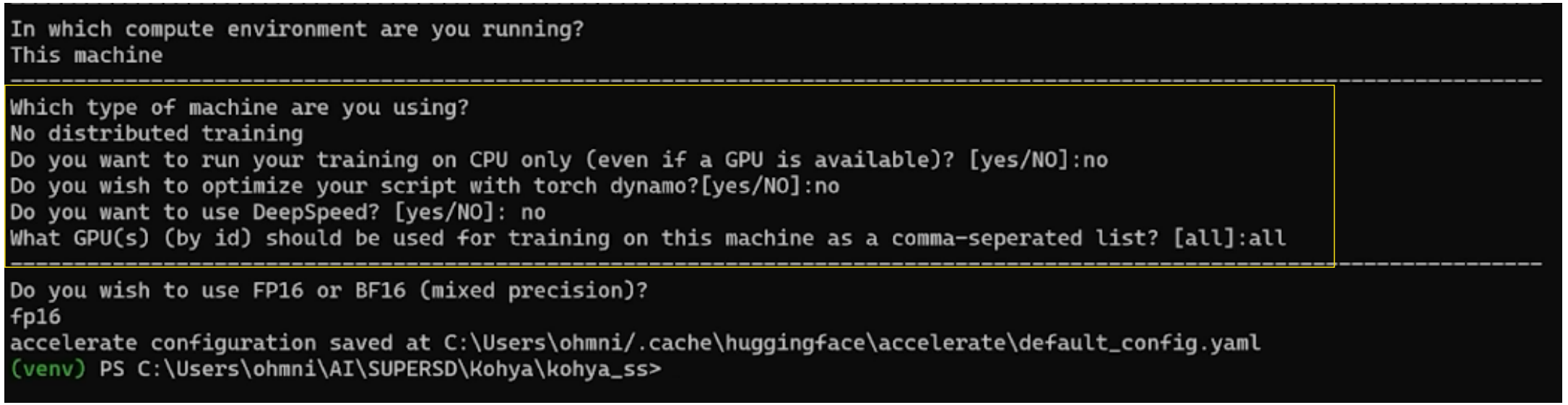
We are using an extra piece of software to create LORAs for Stable diffusion, ist a similar process that Lensa.ai uses to produce your Portraits in different Styles and Outfits.

Choose A Foto - scale it with BIRME to 512 x 512 or 768 x 768, then under „Train“ -- Preprocess Images -- use Blip for caption (Input und Output folder.)

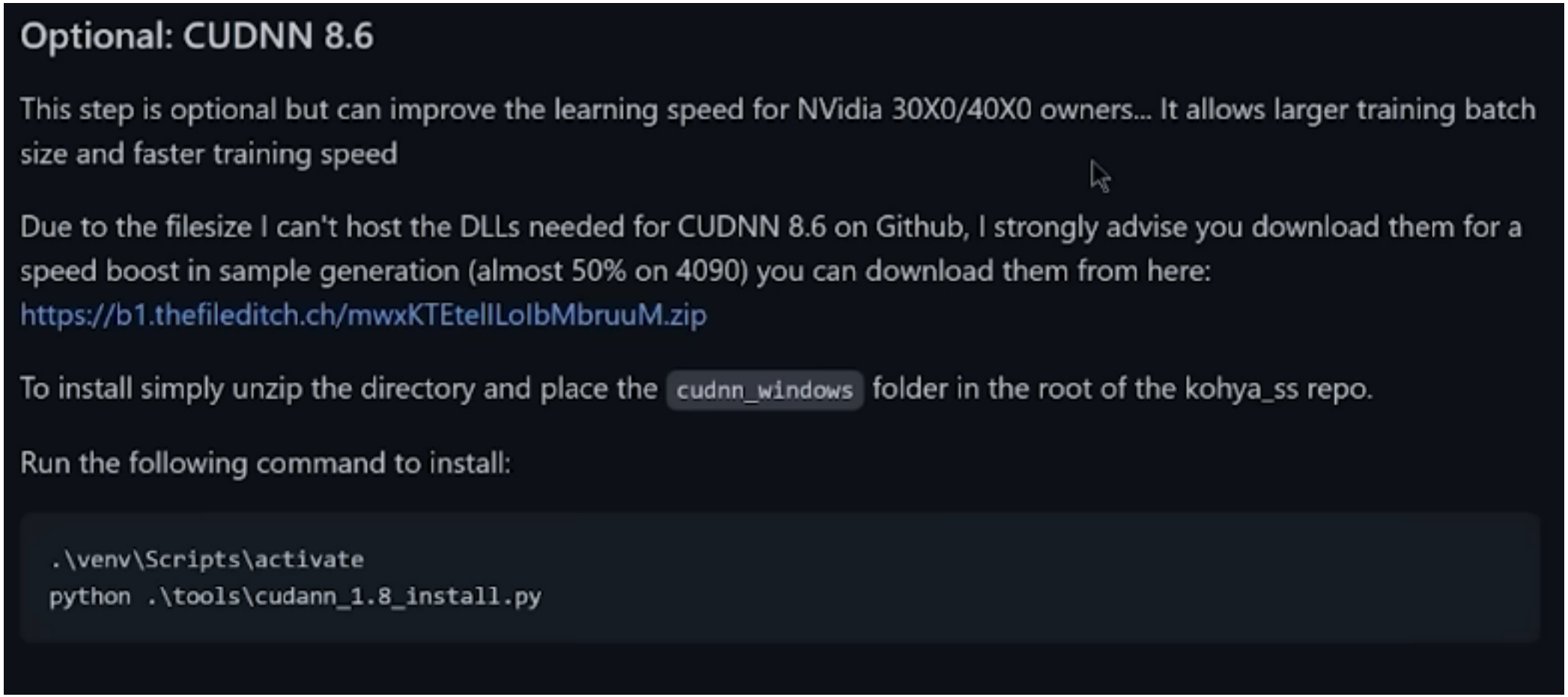
Now in a more detailed way:

Go to https://github.com/bmaltais/kohya_ss copy the following comand „ git clone https://github.com/bmaltais/kohya_ss.git
cd kohya_ss
.\setup.bat „

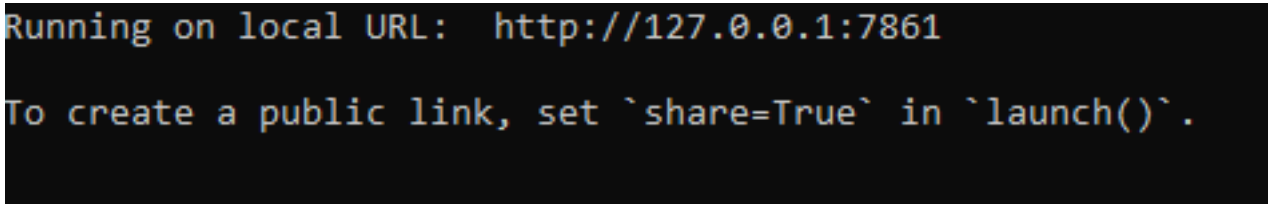
and run it in the terminal (cmd). Wait a few minutes and then answer the questions as below:



If you have a certain Graphics card you can do the next step

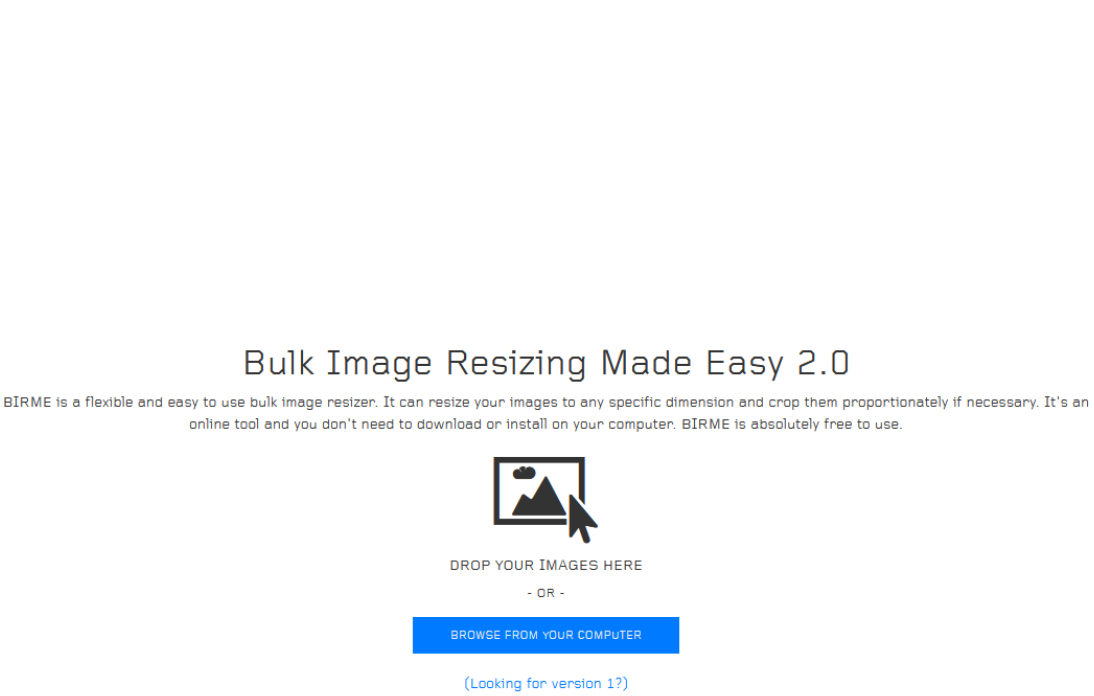


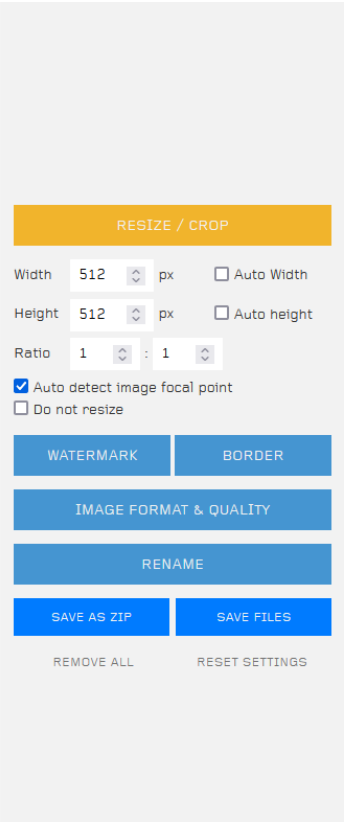

afterwards go to the Kohya_ss folder and start gui.bat




Like with Stable Diffusion paste this URL in your browser and go on.

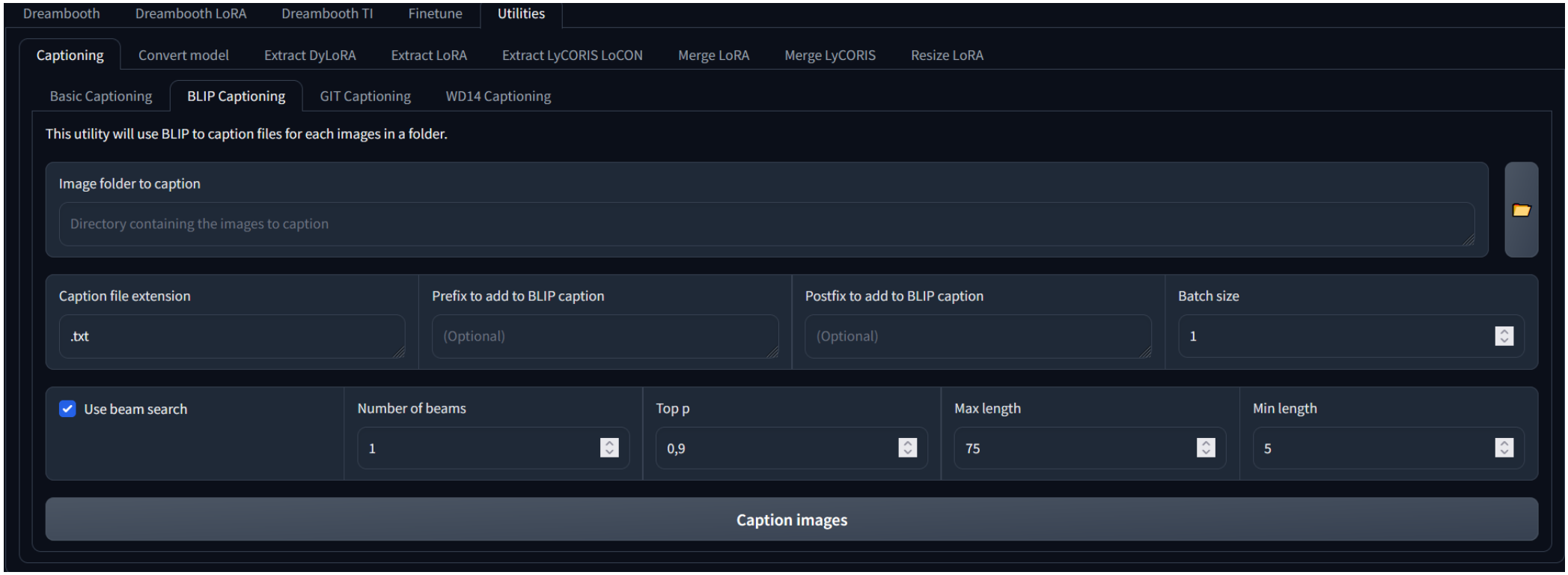
Now it comes to choiosing the images you want to train your Lora with. anything goes, if you are using google us „large“ in image search.
Then go to Birme.net and scale the selected images to 512x512 or 768 x 768



Now it comes to choiosing the images you want to train your Lora with. anything goes, if you are using google us „large“ in image search.
Then go to Birme.net and scale the selected images to 512x512 or 768 x 768.

Go  to the Kohya Gui to create automated Caption for each image:

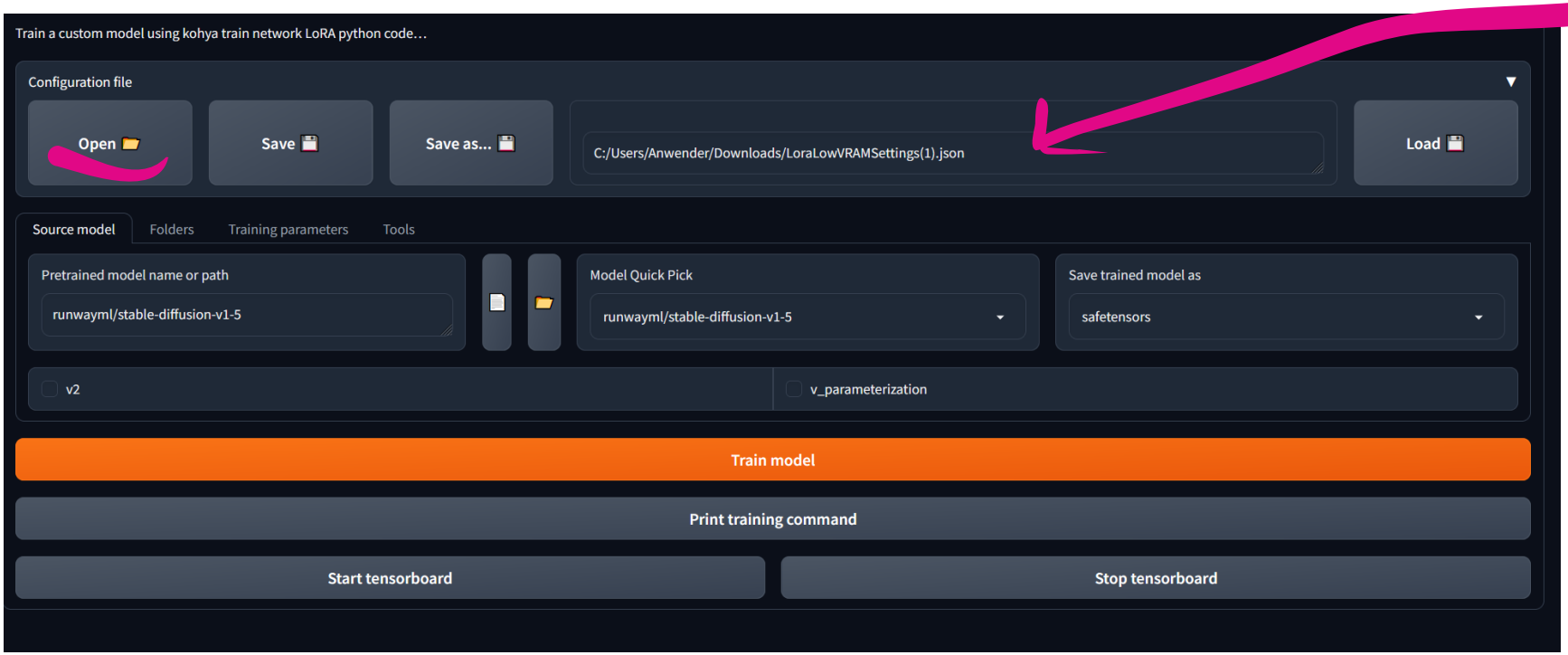



Afterwards open the caption files and input your Trigger Word for the Lora (something very specific Like Hansiiii or Corrrrggiii) and you could even improve  the automated Caption manually.

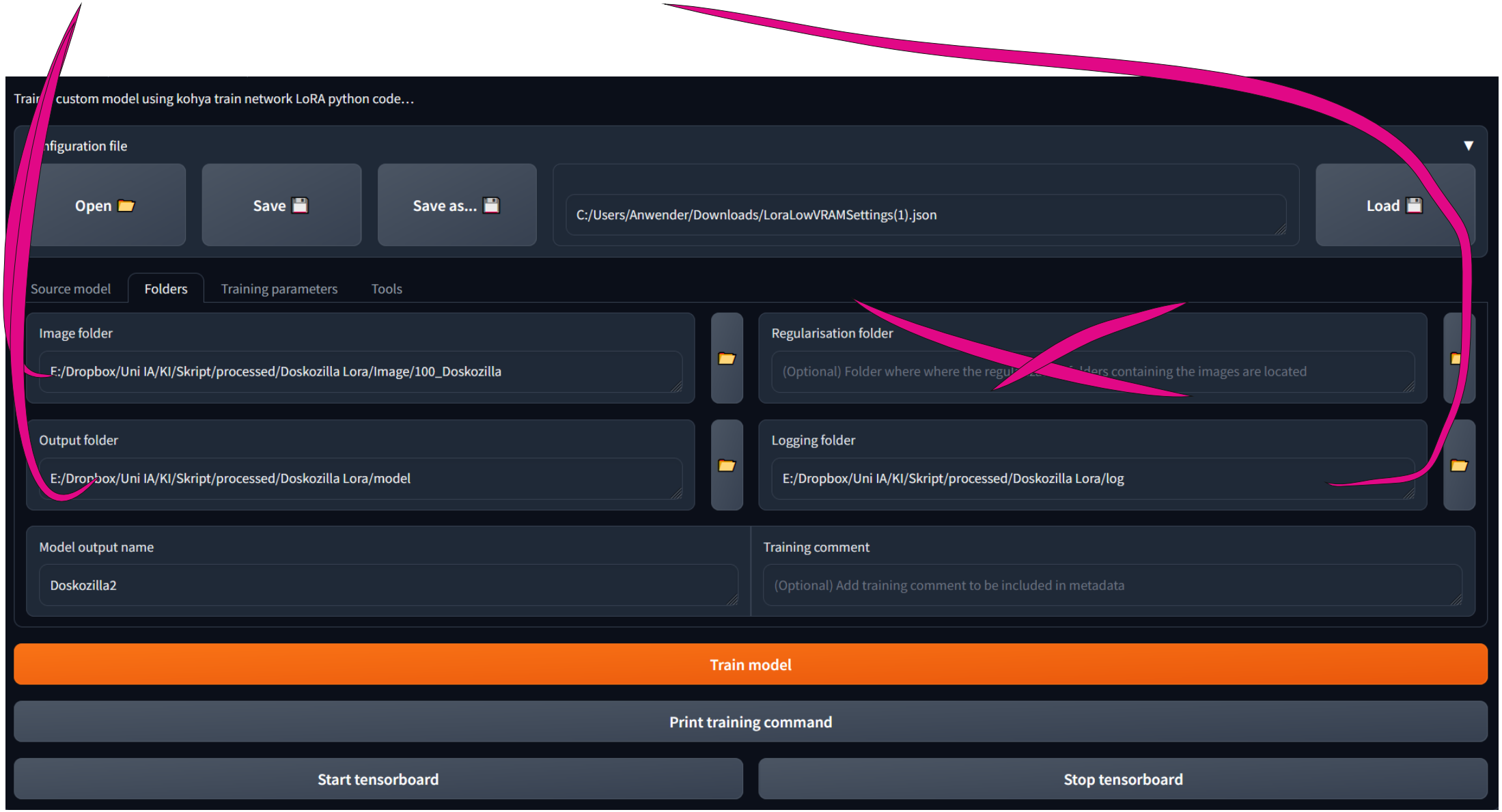
Now ist time to create 3 Folders: Image
Log
Model

In the „Image Folder“ we need a new Folder with a number in its name.
The number comes from 1500 (minimum trainingsteps) divided by the amount of images we have (but the number should be at least 100)


Finally training can begin:
At first we need 1 of 2 diff settings files:
https://mega.nz/file/ndFRwTIB#-T-6AKXtK8Xa8VZGzNAjARNUyGTj2qEYSLp7zePDOIs
https://mega.nz/file/zFNjlJAL#uB2uTAvcqLohSUzYBgtuBcAMt4Jnclg6jVV5YE4s0F4



Link the corresponding folders and you can train right away



LoRA type

Standard

LoRA network weights

{Optional) Path to existing LoRA network weights to resume training

Train batch size 1

Epoch

1

Save every N epochs

1

Caption Extension

.txt

Mixed precision

fp16

Save precision

fp16

Number of CPU threads per core 2

Seed

1234

Cache latent

Learning rate

0.0001

LR Scheduler

constant

LR warmup (% of steps)

0

Optimizer

AdamW8bit

Optimizer extra arguments

(Optional) eg: relative_step=True scale_parameter=True warmup_init=True

Text Encoder learning rate

5e-5

Unet learning rate

0.0001

Network Rank (Dimension) 128

Network Alpha 128

alpha for LoRA weight scaling

Max resolution

The maximum resolution of dataset images. W,H

512,512

mit guter GPU kann man das au f 768 x768 erhöhen

Stop text encoder training 0

After what % of steps should the text encoder stop being trained. 0 = train for all steps.

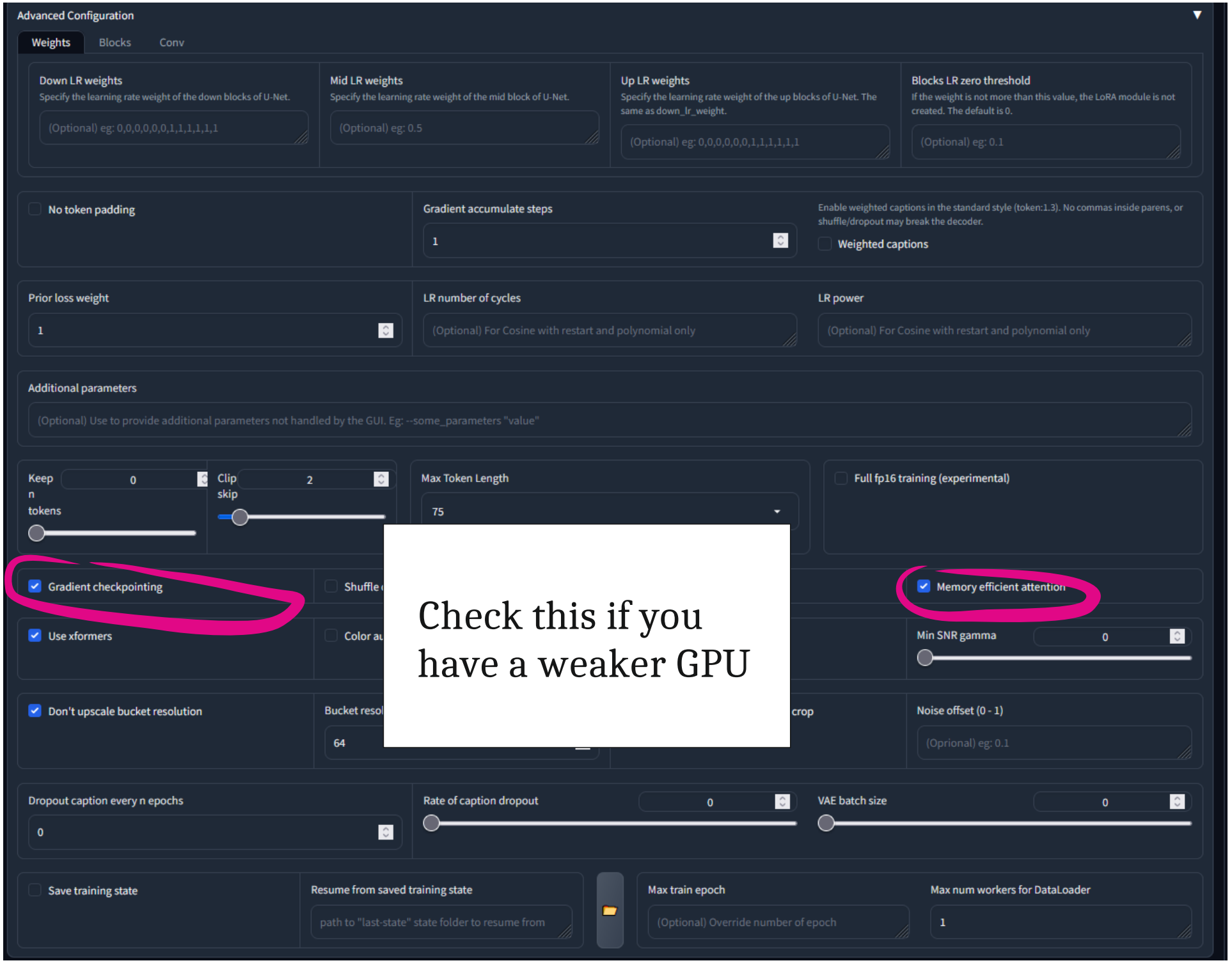
Allow non similar resolution dataset images to be trained on.

Enable buckets

Advanced Configuration

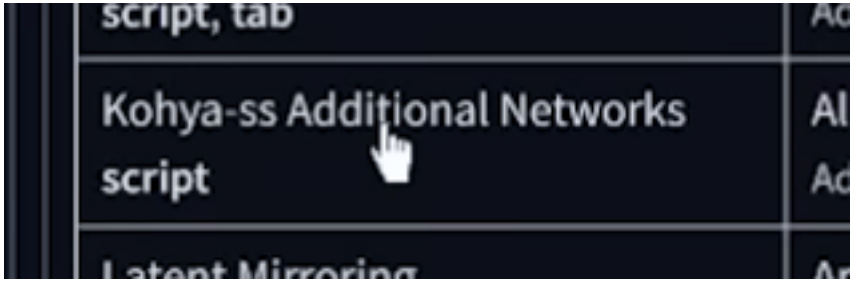
Sample images config

Train model

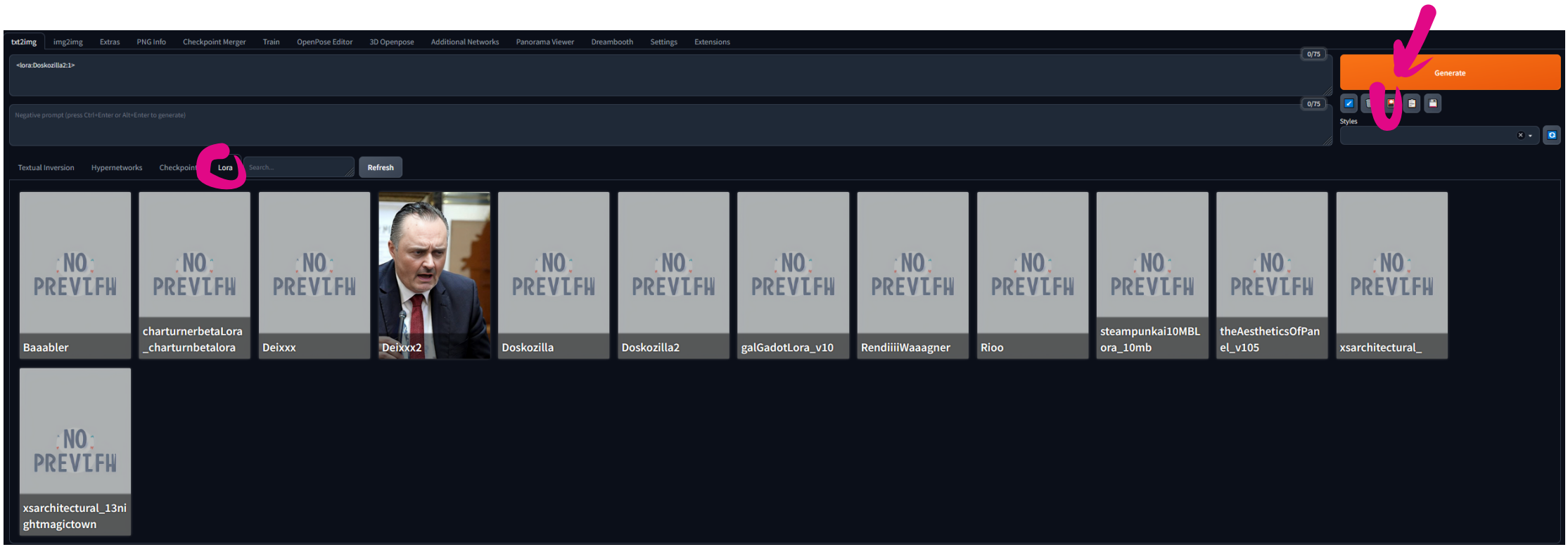

Advanced Configuration
Weights
Blocks
Conv
Down LR weights
Specify the learning rate weight of the down blocks of U-Net.
(Optional) eg: 0,0,0,0,0,0,1,1,1,1,1,1
Mid LR weights
Specify the learning rate weight of the mid block of U-Net.
(Optional) eg: 0.5
Up LR weights
Specify the learning rate weight of the up blocks of U-Net. The same as down_lr_weight.
(Optional) eg: 0,0,0,0,0,0,1,1,1,1,1,1
Blocks LR zero threshold
If the weight is not more than this value, the LoRA module is not created. The default is 0.
(Optional) eg: 0.1
No token padding
Gradient accumulate steps
1
Enable weighted captions in the standard style (token:1.3). No commas inside parens, or shuffle/dropout may break the decoder.
Weighted captions
Prior loss weight
1
LR number of cycles
(Optional) For Cosine with restart and polynomial only
LR power
(Optional) For Cosine with restart and polynomial only
Additional parameters
(Optional) Use to provide additional parameters not handled by the GUI. Eg: --some_parameters "value"
Keep n tokens
0
Clip skip
2
Max Token Length
75
Full fp16 training (experimental)
Gradient checkpointing
Shuffle
Memory efficient attention
Use xformers
Color au
Min SNR gamma
0
Don't upscale bucket resolution
Bucket reso
64
crop
Noise offset (0 - 1)
(Oprional) eg: 0.1
Dropout caption every n epochs
0
Rate of caption dropout
0
VAE batch size
0
Save training state
Resume from saved training state
path to "last-state" state folder to resume from
Max train epoch
(Optional) Override number of epoch
Max num workers for DataLoader
1
Check this if you have a weaker GPU

Go to the Loras mdoel folder und copy the file to the following folder:


Now one last extension is needed
In SD Extras go to Extension - Load from Adress, load it and then activate



Now you can load your Loras, choose them and you will see your trigger wqord appearing.
The number signifies the strength of your special loo, mostly ist advise to decrease it to 0.9- to get the best results.
Xou can also combine different Loras but the sum of the Lora weight numbers always has to add up to 1. (e.g.: 0.3, 0.3 and 0.4)




Now you can load your Loras, choose them and you will see your trigger wqord appearing.
The number signifies the strength of your special loo, mostly ist advise to decrease it to 0.9- to get the best results.
You can also combine different Loras but the sum of the Lora weight numbers always has to add up to 1. (e.g.: 0.3, 0.3 and 0.4)

https://www.youtube.com/watch?v=70H03cv57-o&t=659s

# Process of building a good prompt

## Iterative prompt building

You should approach prompt building as an iterative process. As you see from the previous section, the images could be pretty good with just a few keywords added to the subject.

I always start with a simple prompt with subject, medium, and style only. Generate at least 4 images at a time to see what you get. Most prompts do not work 100% of the time. You want to get some idea of what they can do statistically.

Add at most two keywords at a time. Likewise, generate at least 4 images to assess its effect.

## Using negative prompt

You can use an universal negative prompt if you are starting out.

Adding keywords to the negative prompt can be part of the iterative process. The keywords can be objects or body parts you want to avoid (Since v1 models are not very good at rendering hands, it's not a bad idea to use "hand" in the negative prompt to hide them.)

https://stable-diffusion-art.com/prompt-guide/#Keyword_weight

You can adjust the **weight** of a keyword by the syntax `(keyword: factor)`. `factor` is a value such that less than 1 means less important and larger than 1 means more important.

For example, we can adjust the weight of the keyword `dog` in the following prompt

> **dog**, autumn in paris, ornate, beautiful, atmosphere, vibe, mist, smoke, fire, chimney, rain, wet, pristine, puddles, melting, dripping, snow, creek, lush, ice, bridge, forest, roses, flowers, by stanley artgerm lau, greg rutkowski, thomas kindkade, alphonse mucha, loish, norman rockwell.




Increasing the weight of `dog` tends to generate more dogs. Decreasing it tends to generate fewer. It is not always true for every single image. But it is true in a statistical sense.

# () and [] syntax

(*This syntax applies to AUTOMATIC1111 GUI.*)

An equivalent way to adjust keyword strength is to use `()` and `[]`. `(keyword)` increases the strength of the keyword by a factor of 1.1 and is the same as `(keyword:1.1)`. `[keyword]` decrease the strength by a factor of 0.9 and is the same as `(keyword:0.9)`.

You can use multiple of them, just like in Algebra... The effect is multiplicative.

> (keyword): 1.1
> ((keyword)): 1.21
> (((keyword))): 1.33

Similarly, the effects of using multiple `[]` are

> [keyword]: 0.9
> [[keyword]]: 0.81
> [[[keyword]]]: 0.73

https://stable-diffusion-art.com/prompt-guide/#Keyword_weight

Keyword blending

(This syntax applies to AUTOMATIC1111 GUI.)

You can mix two keywords. The proper term is prompt scheduling. The syntax is

[keyword1 : keyword2: factor]

factor controls at which step keyword1 is switched to keyword2. It is a number between 0 and 1.

For example, if I use the prompt

Oil painting portrait of [Joe Biden: Donald Trump: 0.5]

for 30 sampling steps.

That means the prompt in steps 1 to 15 is

Oil painting portrait of Joe Biden

And the prompt in steps 16 to 30 becomes

Oil painting portrait of Donald Trump


The factor determines when the keyword is changed. it is after 30 steps x 0.5 = 15 steps.

The effect of changing the factor is blending the two presidents to different degrees.

You may have noticed Trump is in a white suit which is more of a Joe outfit. This is a perfect example of a very important rule for keyword blending: The first keyword dictates the global composition. The early diffusion steps set the overall composition. The later steps refine details.

https://stable-diffusion-art.com/prompt-guide/#Keyword_weight

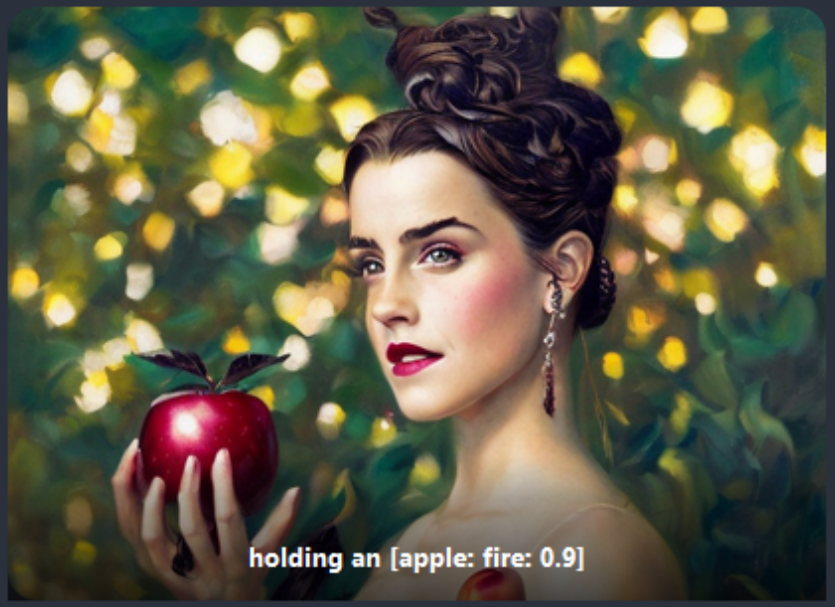
## Blending faces

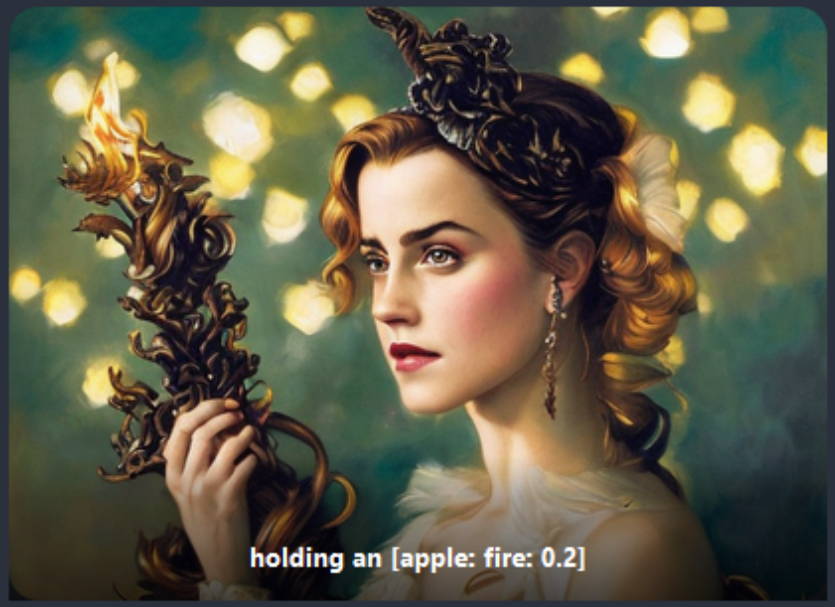
A common use case is to create a new face with a particular look, borrowing from actors and actresses. For example, [Emma Watson: Amber heard: 0.85], 40 steps is a look between the two:

When carefully choosing the two names and adjusting the factor, we can get the look we want precisely.

## Poor man's prompt-to-prompt

Using keyword blending, you can achieve effects similar to prompt-to-prompt, generating pairs of highly similar images with edits. The following two images are generated with the same prompt except for a prompt schedule to substitute `apple` with `fire`. The seed and number of steps were kept the same.




The factor needs to be carefully adjusted. How does it work? The theory behind this is the overall composition of the image was set by the early diffusion process. Once the diffusion is trapped in a small space, swapping any keywords won't have a large effect on the overall image. It would only change a small part.

> castle, **blue sky background**

> **wide angle view** of castle, blue sky background

By adding more describing keywords in the prompt, we narrow down the sampling of castles. In We asked for **any image of a castle** in the first example. Then we asked to get only those with a blue sky background. Finally, we demanded it is **taken as a wide-angle photo**.

The more you specify in the prompt, the less variation in the images.

## Attribute association

Some attributes are strongly correlated. When you specify one, you will get the other. Stable Diffusion generates the most likely images that could have an unintended association effect.

Let's say we want to generate photos of women with **blue eyes**.

> a young female with **blue eyes**, highlights in hair, sitting outside restaurant, wearing a white outfit, side light

Blue eyes

What if we change to brown eyes?

> a young female with **brown eyes**, highlights in hair, sitting outside restaurant, wearing a white outfit, side light

Brown eyes

Nowhere in the prompts, I specified ethnicity. But because people with blue eyes are predominantly Europeans, Caucasians were generated. Brown eyes are more common across different ethnicities, so you will see a more diverse sample of races.

Stereotyping and bias is a big topic in AI models. I will confine to the technical aspect in this article.

# Embeddings are keywords

Embeddings, the result of textual inversion, are nothing but combinations of keywords. You can expect them to do a bit more than what they claim.

Let's see the following base images of Ironman making a meal without using embeddings.

Prompt: iron man cooking in kitchen.

Style-Empire is an embedding I like to use because it adds a dark tone to portrait images and creates an interesting lighting effect. Since it was trained on an image with a street scene at night, you can expect it adds some blacks AND perhaps buildings and streets. See the images below with the embedding added.

Style-Empire is an embedding I like to use because it adds a dark tone to portrait images and creates an interesting lighting effect. Since it was trained on an image with a street scene at night, you can expect it adds some blacks AND perhaps buildings and streets. See the images below with the embedding added.

Prompt: iron man cooking in kitchen **Style-Empire**.

Note some interesting effects

- The background of the first image changed to city buildings at night.
- Iron man tends to show his face. Perhaps the training image is a portrait?

So even if an embedding is intended to modify the style, it is just a bunch of keywords and can have unintended effects.

Embeddings explained:

# Effect of custom models

Using a custom model is the easiest way to achieve a style, guaranteed. This is also a unique charm of Stable Diffusion. Because of the large open-source community, hundreds of custom models are freely available.

When using a model, we need to be aware that the meaning of a keyword can change. This is especially true for styles.

Let's use Henry Asencio again as an example. In v1.5, his name alone generates:

Using DreamShaper, a model fine-tuned for portrait illustrations, with the same prompt gives

It is a very decent but distinctly different style. The model has a strong basis for generating clear and pretty faces, which has been revealed here.

So make sure to check when you use a style in custom models. van Gogh may not be van Gogh anymore!