

## Task 5 - Visual Generative AI - New It-Jim Logo



Our [current logo is pretty cool](#), but we decided that we want to try something more fancy. Something is missing... Maybe a little bit of glitter? Anime girls? Oh, why don't you decide that? We entrust you with this important task: using open-source models, generate a new It-Jim logo using the existing one as a reference. You might want to use some cool interfaces for this: ComfyUI, Fooocus. Automatic1111 will probably be the most suitable if you have never had experience with Stable Diffusion before. Also, you should go beyond the simple img2img pipeline and add some additional conditioning models to increase control over the results.

**Note:** Stable Diffusion models are quite hungry for video RAM, so you'll need at least 6 GB of VRAM, perfectly 8+. If you don't have a CUDA-capable device, use GoogleColab for training.

Deliverable:

- Several variations of generated images;
- Complete report of the steps, where we can see the models you used, selected inference parameter and some intermediate results;
- Workflow JSON in case you selected ComfyUI as a working environment (respect).

## Solution

### Business Understanding

The task involves generating a new logo for It-Jim, a company that desires a more modern, visually appealing, and potentially innovative logo design. The task requires

using open-source models and additional conditioning models to enhance the control over the results. The primary goal is to explore different prompts and parameters to create a variety of logos that meet the company's aesthetic and branding requirements.

### **Compliance with requirements:**

- Use the existing It-Jim logo as a reference.
- Implement the logo generation using self written web ui using gradio. Unfortunately, I did not have time to dive into web interfaces you suggested, so I intended to use what I knew.
- Go beyond the simple img2img pipeline by adding additional conditioning models.

### **What I deliver:**

- A lot of variations of generated images with google disk links, which contain all results.
- A complete report detailing the steps taken, the models used, the selected inference parameters, and some intermediate results.
- A notebooks with research and deployed model

### **Constraints:**

- Faced up memory limits in google colab, so significantly reduced combinations of different models and control nets types. Created a few notebooks from different accounts, so got the result saved in 3 different google disks.

## **Modeling**

My experiments can be fined here -

[https://colab.research.google.com/drive/179mOSjzjPh3JhFTA\\_ffJ3QjLeEWBQGUB?usp=sharing](https://colab.research.google.com/drive/179mOSjzjPh3JhFTA_ffJ3QjLeEWBQGUB?usp=sharing)

## **Models**

Firstly I needed to choose the model, so I was surfing the internet looking at some articles with stable diffusion models review. I chose two models:

1. Juggernaut - [https://huggingface.co/digiplay/Juggernaut\\_final](https://huggingface.co/digiplay/Juggernaut_final)
2. Anything v5 - <https://huggingface.co/stablediffusionapi/anything-v5>

I definitely wanted to try more models, but the limits of google colab did not agree with me on this issue.

P.S. Anything v5 is known to be the best anime model, as you wanted!

## Control nets

Then, I was looking at different control nets and decided that in terms of current It-Jim logos these will work best :

### canny\_edge:

- **Description:** Uses Canny edge detection to create an edge map of the input image.
- **Use Case:** Useful for maintaining the structure of the input image while allowing for creative modifications and enhancements. Ideal for applications where you want to preserve the geometric outline of the original image.

This one will try to retain all elements of the logo and add creative modifications simultaneously.

### hed:

- **Description:** Uses Holistically-Nested Edge Detection (HED) to create an edge map.
- **Use Case:** Similar to canny\_edge but can provide more detailed and intricate edge maps for complex structures. Useful for applications that require detailed and nuanced edge detection, such as architectural rendering and technical illustrations.

May be a good idea, if we want to experiment with some additional figures.

### hough:

- **Description:** Uses Hough Line Transform to detect lines in the image.
- **Use Case:** Suitable for images with strong linear structures or for emphasizing lines and geometric shapes. Ideal for applications in technical drawing, architectural visualization, and any scenario where linear features are prominent

I think it is the best one, because it is mainly used in technical drawing, so it can work well with geometric figures and some lines, which are on the current It-Jim logo.

## Pipelines

Then I created a testing pipeline, which gave me the ability to test each model and control net type combinations.

Firstly, I explored different pipeline hyperparameters:

#### **width and height:**

- These parameters manage the resolution of the generation. I reduced them from original ones, to (512, 256), considering that speed was crucial in my case. Anyway, resolution can be increased, if we choose some of the generated logos, but , ideally, it should be a part of the pipeline. It is one of that things, which can improve my approach.

#### **guidance\_scale:**

- Prompt guidance dictates how closely the generation follows a given prompt. The default value is 7, with a recommended value range of 4 to 9.

#### **image (control\_image):**

- This is used to guide or control the generation based on another reference image.

#### **controlnet\_conditioning\_scale:**

- This influences the strength of the guidance image, with a recommended value range of 0.0 to 1.5.

#### **num\_inference\_steps:**

- The higher the value, the better the quality we get, with a recommended value range of 20 to 50.

Then I made an auto save of generated logos to google drive in order not to lose some good ones.

## **Evaluation**

I created a few prompts, which I used for model testing. Here are some I used.

Interesting ones:

```
prompts = [
    1.   "Modern tech logo, futuristic design, sleek lines, incorporating blue and gray colors, high-tech feel, minimalist, clean edges, digital aesthetics, best quality, highly detailed",
    2.   "Nature-inspired logo, incorporating blue and gray colors, abstract leaves and geometric shapes, organic and modern blend, eco-friendly feel, high-resolution, detailed, clean and professional",
```

3. "Futuristic abstract logo, vibrant colors, incorporating blue and gray, dynamic shapes, geometric patterns, modern design, high-tech look, glowing edges, 3D elements, highly detailed, professional quality",
4. "Beautiful anime girl with sparkling glitter, vibrant colors, detailed eyes, flowing hair, magical aura, fantasy setting, high-resolution, highly detailed, enchanting, best quality.",
5. "Mystical forest scene, glowing plants, enchanted atmosphere, deep greens and blues, otherworldly creatures, fantasy art, high-resolution, highly detailed",
6. "Steampunk cityscape, intricate machinery, bronze and copper tones, Victorian architecture, bustling streets, high-resolution, highly detailed",
7. "Cyberpunk character, neon lights, futuristic attire, urban setting, dark atmosphere, high-resolution, highly detailed",
8. "Retro-futuristic landscape, vibrant colors, imaginative technology, 1980s sci-fi aesthetic, high-resolution, highly detailed",
9. "Abstract art piece, vibrant colors, dynamic shapes, fluid forms, modern and experimental, high-resolution, highly detailed",
10. "Underwater scene, colorful coral reefs, diverse marine life, sun rays penetrating the water, serene and vibrant, high-resolution, highly detailed"

]

More serious ones:

prompts = [

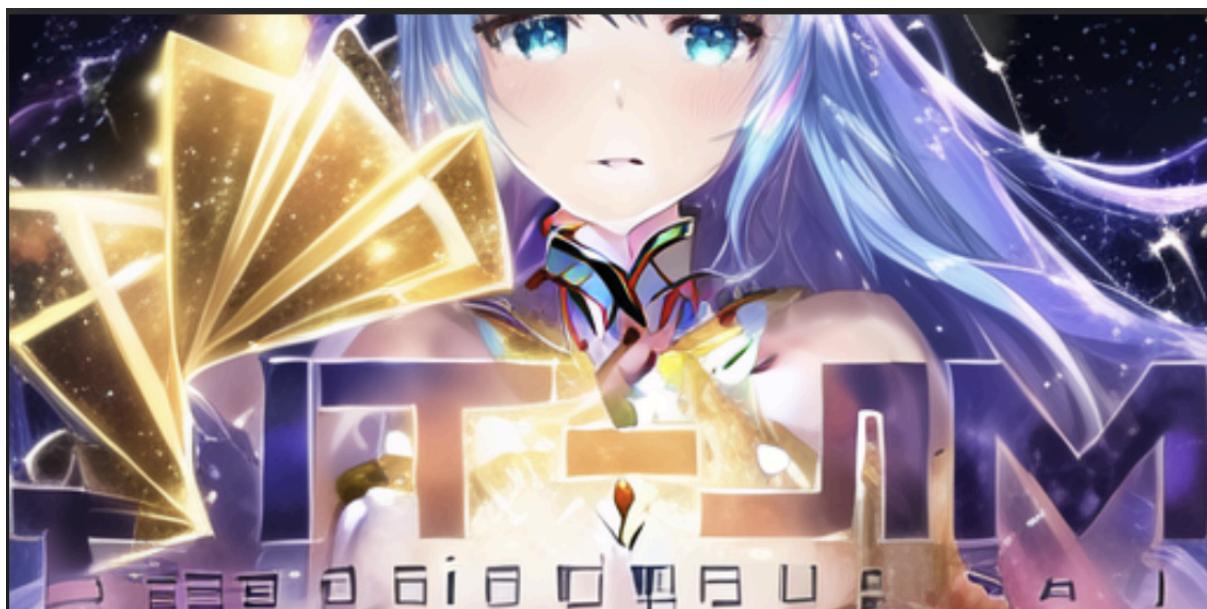
1. "Professional tech logo, sleek design, incorporating blue and gray colors, clean lines, modern typography, minimalist, best quality, highly detailed",
2. "Cutting-edge technology logo, futuristic design, blue and gray color scheme, abstract elements, digital aesthetics, sharp lines, high-resolution, highly detailed",
3. "Innovative company logo, blue and gray palette, modern and sophisticated design, tech-oriented, clean and professional, high-resolution, highly detailed",
4. "Modern IT logo, geometric shapes, blue and gray colors, sleek and professional, minimalist, high-tech feel, high-resolution, highly detailed",
5. "Abstract tech logo, innovative design, blue and gray hues, modern and dynamic, clean lines, futuristic, high-resolution, highly detailed",
6. "Corporate technology logo, elegant design, blue and gray color scheme, professional and modern, clean lines, minimalist, high-resolution, highly detailed",
7. "Dynamic tech logo, blue and gray palette, abstract design, modern and sleek, professional appearance, high-resolution, highly detailed",
8. "Sophisticated IT logo, clean and modern design, blue and gray colors, geometric elements, professional look, high-resolution, highly detailed",
9. "High-tech company logo, blue and gray hues, futuristic and minimalist design, clean lines, digital aesthetics, high-resolution, highly detailed",
10. "Innovative tech firm logo, abstract and modern design, blue and gray color scheme, professional and sleek, high-resolution, highly detailed"

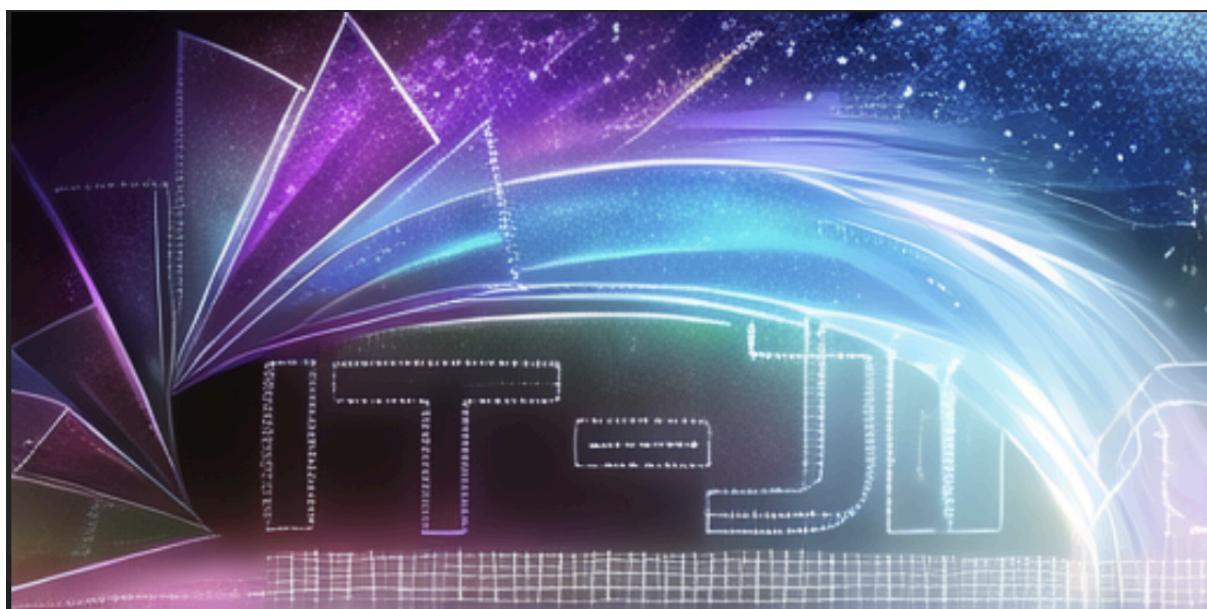
]

Considering that while researching I was working in 2 different collabs, I also have two different disks with experiments -

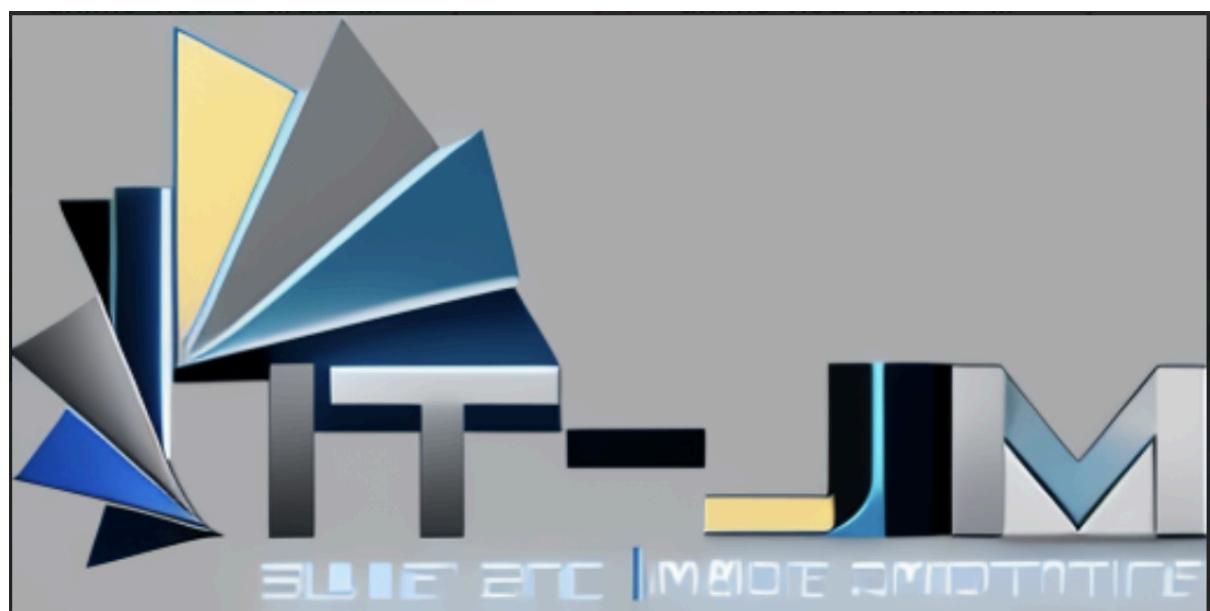
1. <https://drive.google.com/drive/folders/1QBB6tYo40xrJRTZwgiM0T3ZEfkIg12V?usp=sharing>
2. <https://drive.google.com/drive/folders/1iDmefYWcmc2tB5WsEILHG9tzhSAvOys6?usp=sharing>

Here are some of them:









## Deployment

I thought that “hough” control net would work best and in order to try to generate some logo, which really may be used as logotip of It-Jim I decided to not to use Anything v5, so I implemented the pipeline with Juggernaut model.

To launch the proposed logo generator implementation you need to copy this notebook [https://colab.research.google.com/drive/16PChXk1uyum7XAJPmlmU004VCPGCkB\\_k?usp=sharing](https://colab.research.google.com/drive/16PChXk1uyum7XAJPmlmU004VCPGCkB_k?usp=sharing)

connect it to your google drive, where you have a current It-Jim logo and the folder where generated ones will be saved.

To connect to google drive

```
from google.colab import drive  
drive.mount('/content/drive')
```

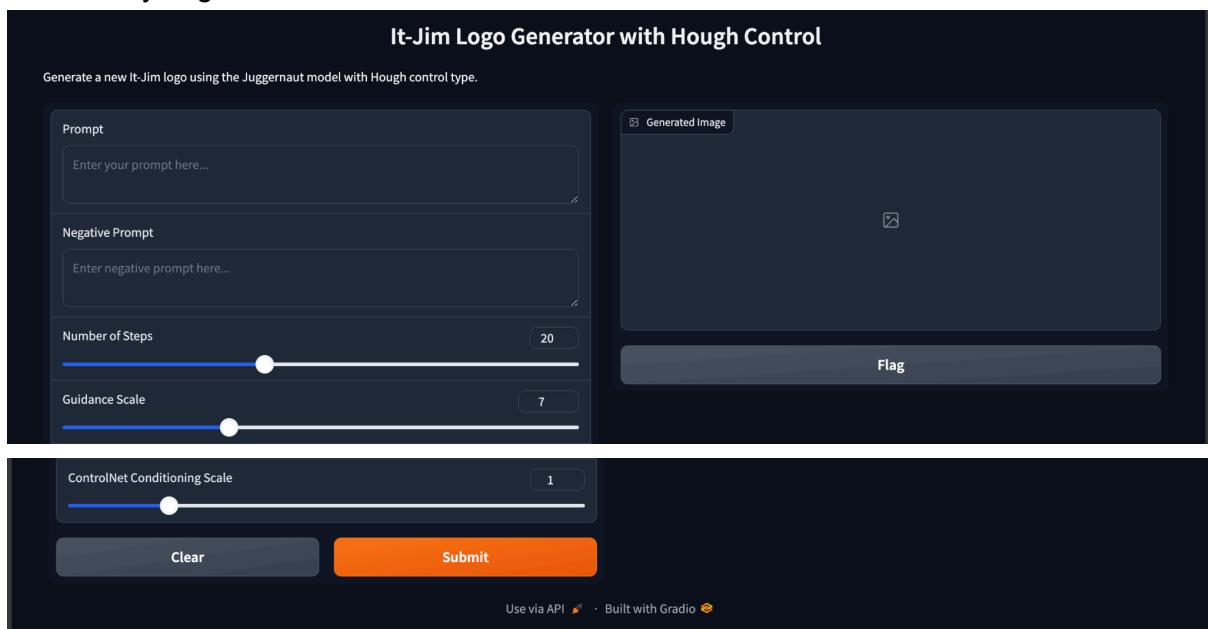
then specify logo path and save folder path:

```
IMG_PATH = "/content/drive/MyDrive/It-Jim/It-Jim_logo.png"  
SAVE_FOLDER_PATH = "/content/drive/MyDrive/It-Jim/generated_logs/"
```

Then run all cells.

I created a web ui using gradio. Did not have time due to the session period to dive into proposed web interfaces, so I used what I knew.

In the end you get this window

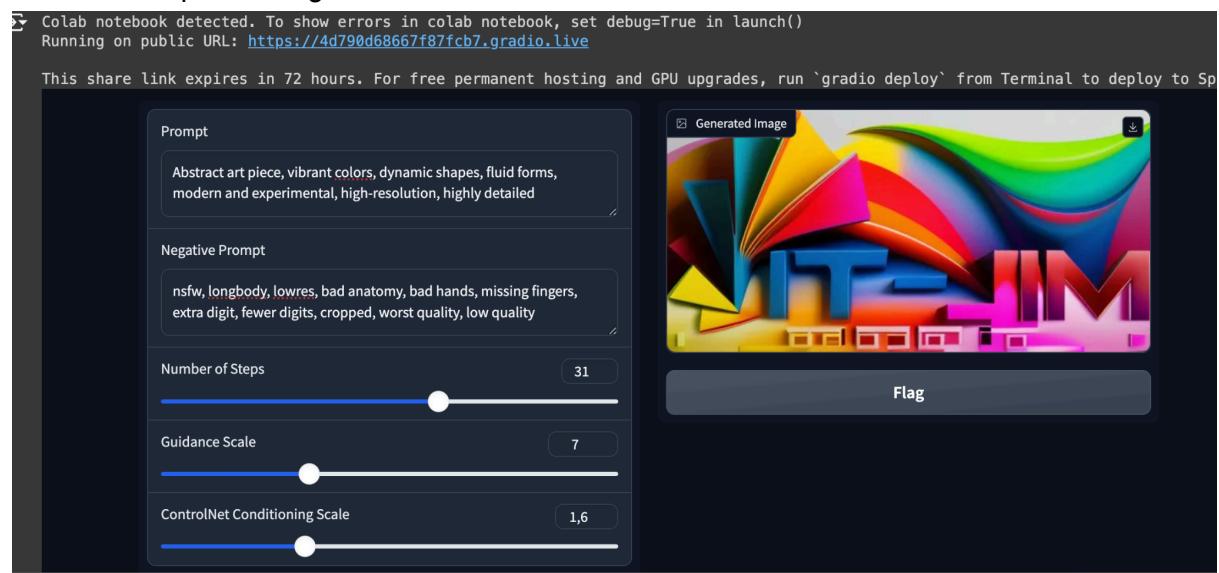


where you can specify prompt, negative prompt and control parameters of generation.

negative prompt - you can use this one

```
nsfw, longbody, lowres, bad anatomy, bad hands, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality
```

Here is example of usage:

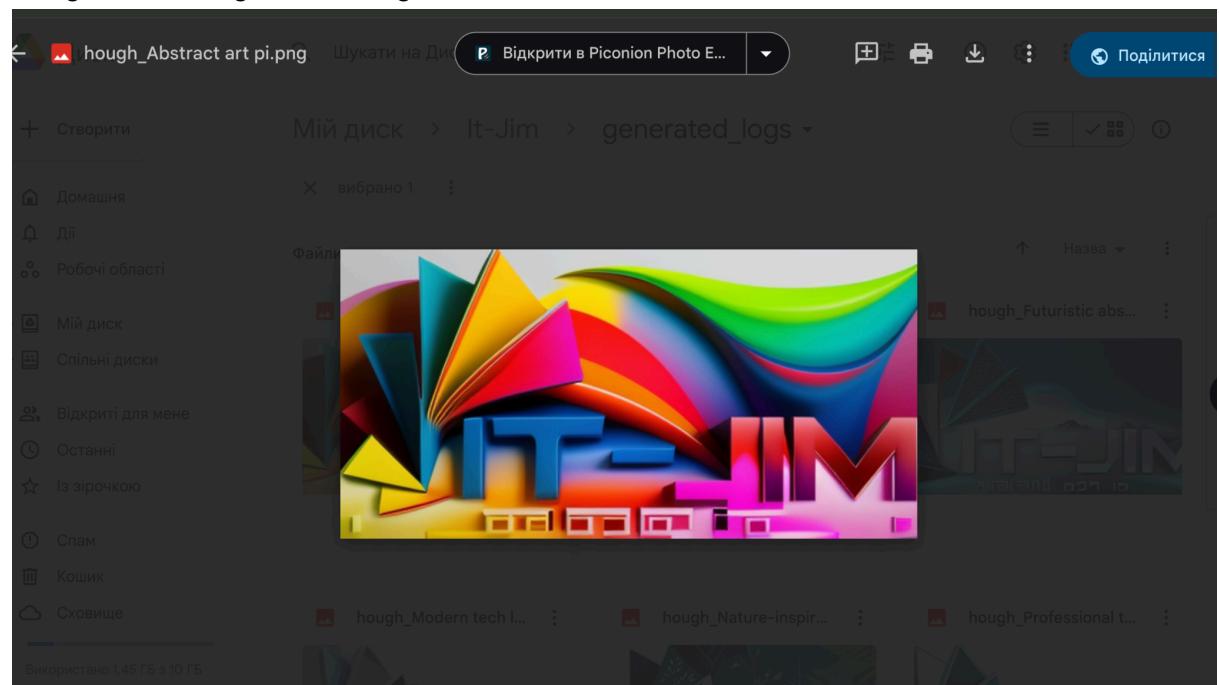


If you want to save the current logo, press Flag and it will be saved to the specified google disk folder.

In my case -

[https://drive.google.com/drive/folders/164Jp4PrtMRnjslrlFy-PyVSH\\_WmO8nA6?usp=sharing](https://drive.google.com/drive/folders/164Jp4PrtMRnjslrlFy-PyVSH_WmO8nA6?usp=sharing)

and got new one generated logo



Testing changing params:

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to S

**Prompt**

Abstract art piece, vibrant colors, dynamic shapes, fluid forms, modern and experimental, high-resolution, highly detailed

**Negative Prompt**

nsfw, longbody, lowres, bad anatomy, bad hands, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality

**Number of Steps** 32

**Guidance Scale** 3

**ControlNet Conditioning Scale** 2,5

**Clear** **Submit**

**Generated Image**

**Flag**

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to S

**Prompt**

Abstract art piece, vibrant colors, dynamic shapes, fluid forms, modern and experimental, high-resolution, highly detailed

**Negative Prompt**

nsfw, longbody, lowres, bad anatomy, bad hands, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality

**Number of Steps** 32

**Guidance Scale** 3

**ControlNet Conditioning Scale** 0,1

**Clear** **Submit**

**Generated Image**

**Flag**

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to S

**Prompt**

Abstract art piece, vibrant colors, dynamic shapes, fluid forms, modern and experimental, high-resolution, highly detailed

**Negative Prompt**

nsfw, longbody, lowres, bad anatomy, bad hands, missing fingers, extra digit, fewer digits, cropped, worst quality, low quality

**Number of Steps** 32

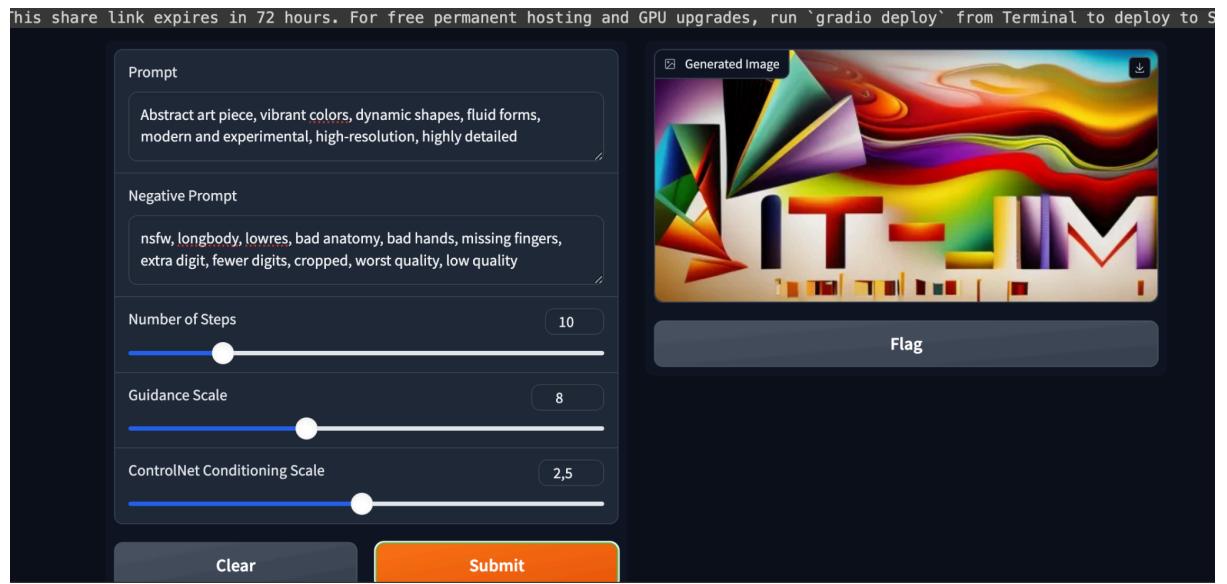
**Guidance Scale** 1

**ControlNet Conditioning Scale** 2,5

**Clear** **Submit**

**Generated Image**

**Flag**



## Conclusion

As a signal processing task this one also is new to me, and it was very exciting to explore abilities of modern AI technologies. Of course, my solution is certainly not the best and there are a lot of things to improve, but I found out a lot of new things and I am pretty satisfied with my result for today!