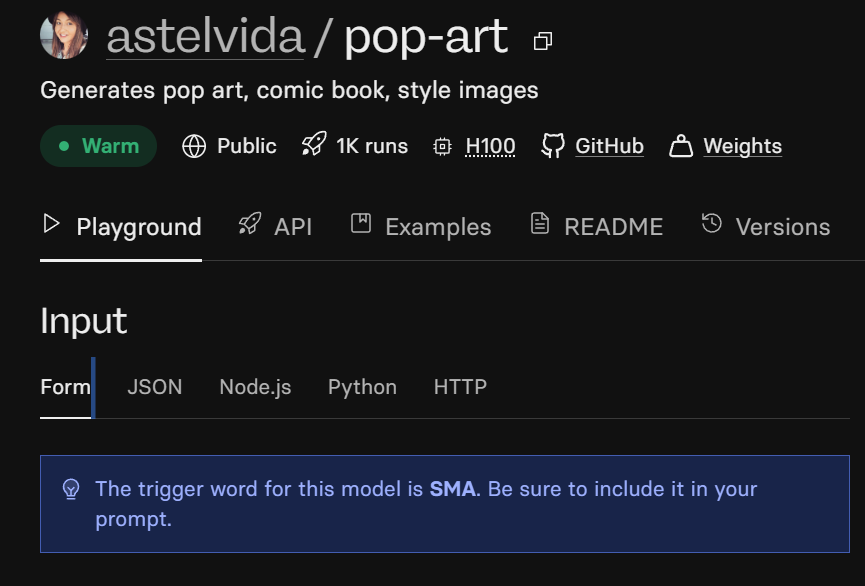
My approach to the comic book cover generator problem

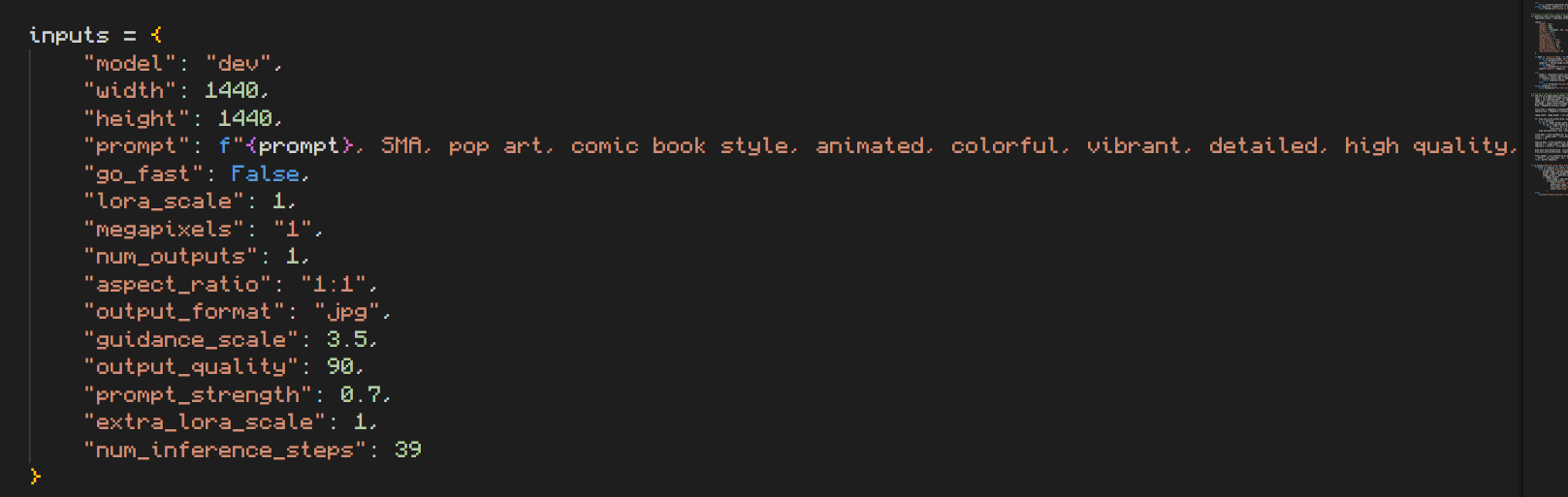
1. I started off with crafting the prompt for comic book cover generation directly with gpt 4o vision.   
   Input : Image, title, tagline style   
   Using this with the prompt , gpt 4o vision accurately gives us a good descriptive prompt to use for stable diffusion  
     
   "You are given an image. Use this image as reference. The title of the comic is '{title}', and the tagline is '{tagline}'. "

f"The style is '{style}'. Generate a Stable Diffusion positive prompt to create a comic book art picture with these details."

1. Then this prompt is input into a fine tuned model hosted on Replicate trained on comic book art



I tested the best settings for this model with a lot of trial and error including the perfect prompt



However I kept 2 modes

1. Creative mode : DOES NOT USE IMG2IMG for generation, solely works on prompts

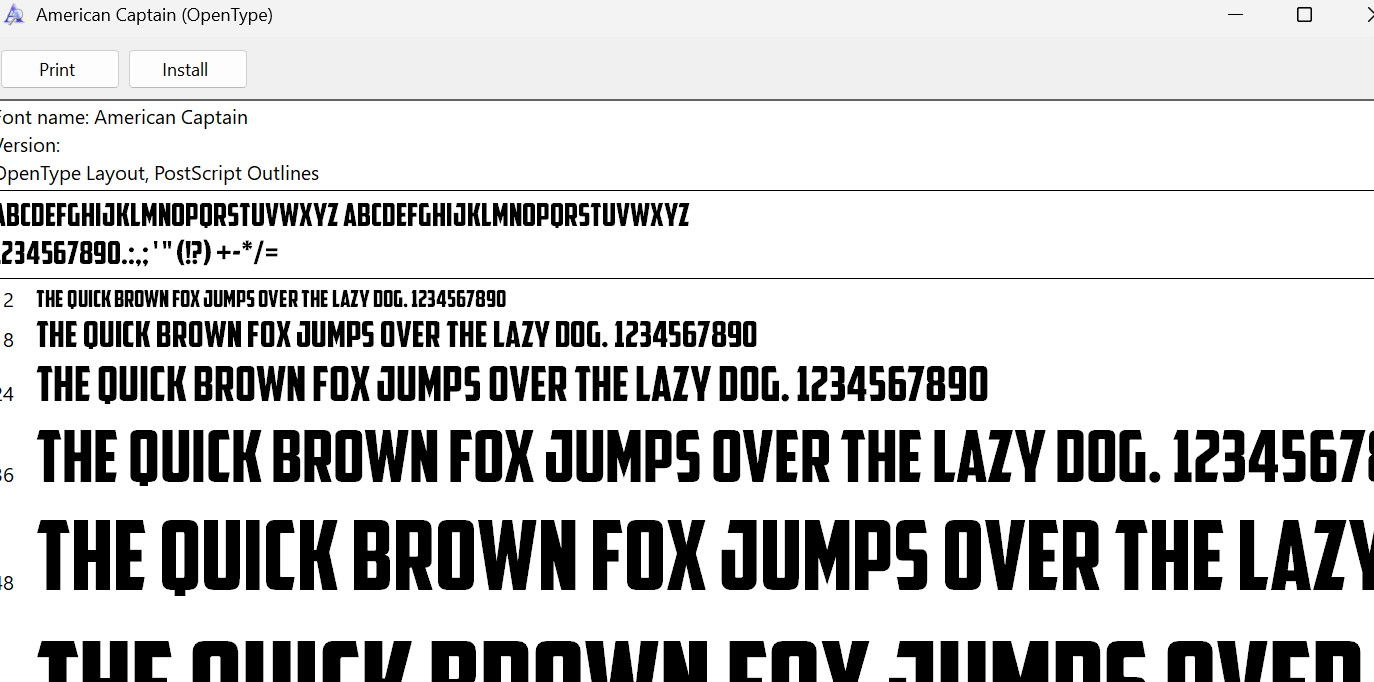
I did this because sometimes with img2img input the output is deformed and this model did not have the option of negative prompt.

1. Stick to image : If the user gives preference of sticking to input image it uses that as input to the model
2. For the problem of text overlaying like cover and tagline on top of the comic cover,

**KEEPING IN MIND, DIFFUSION MODELS CANNOT GENERATE TEXT WELL**

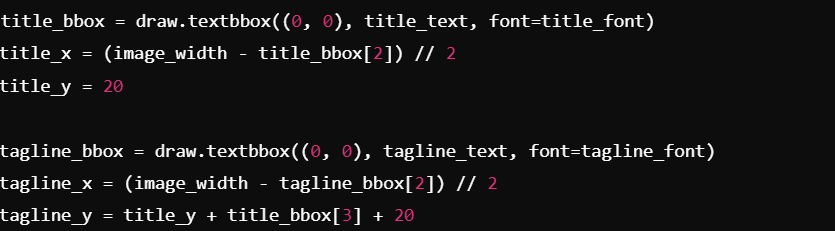
I used a pre existing font for comic books

(these can be replaced with multiple fonts and we can train a model to choose font depending on genre but for now I have chosen one only )



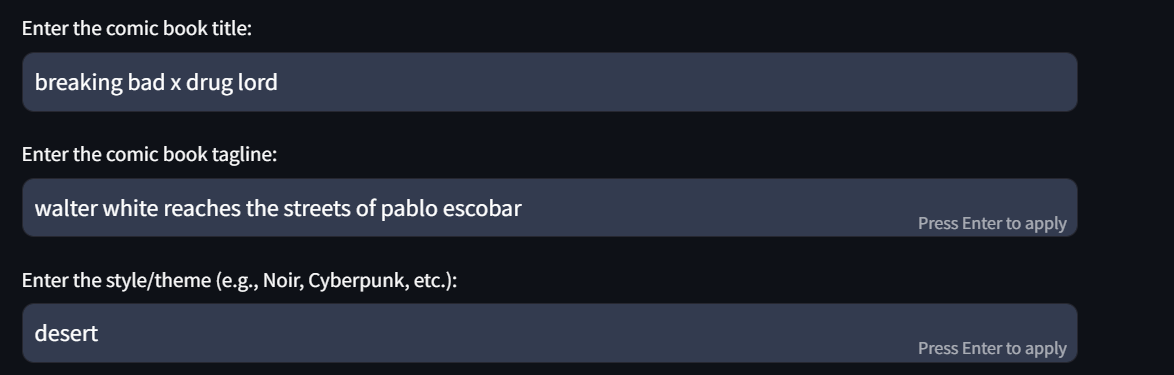
**Install this font before running the project**

1. Font is loaded
2. Retrieve the width and height of the image for positioning text in the next steps.
3. Add a "border" effect around text by drawing multiple offset versions of the text in a border color before rendering the main text.
4. Calculate text placement based on image size



This is how I am able to place the text at fixed positions after cover is generated

**RESULTS**





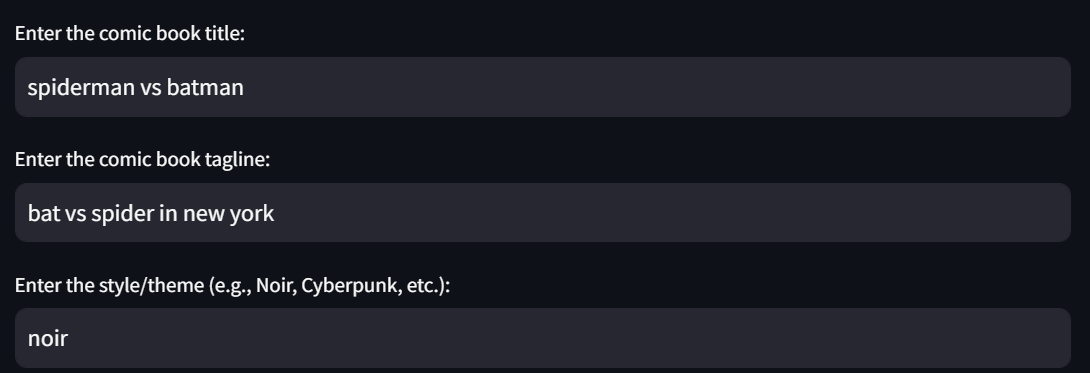
**Output :**

****

**The font and its placement can be improved this is just for POC**

1. **Input**

****

****

**Output**

****