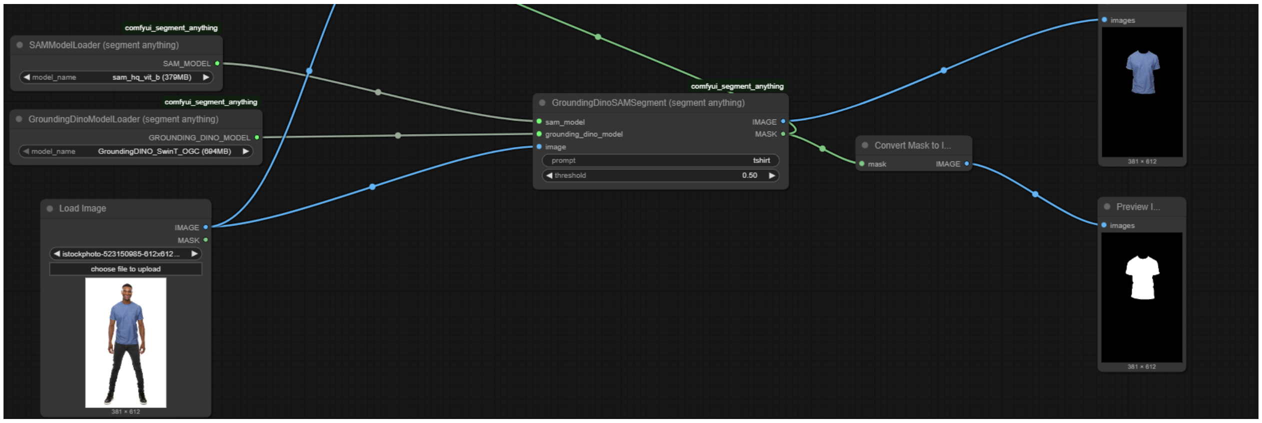
LICENTA DENIS ANDRIS – POC

I was able to create a POC using already trained models.

1) Load image and use Segment Anything and Grounding Dino combined to get the mask of the clothing we want to change



model_name sam_hq_vit_b (379MB)

model_name GroundingDINO_SwinT_OGC (694MB)

istockphoto-523150985-612x612...

prompt tshirt

threshold 0.50

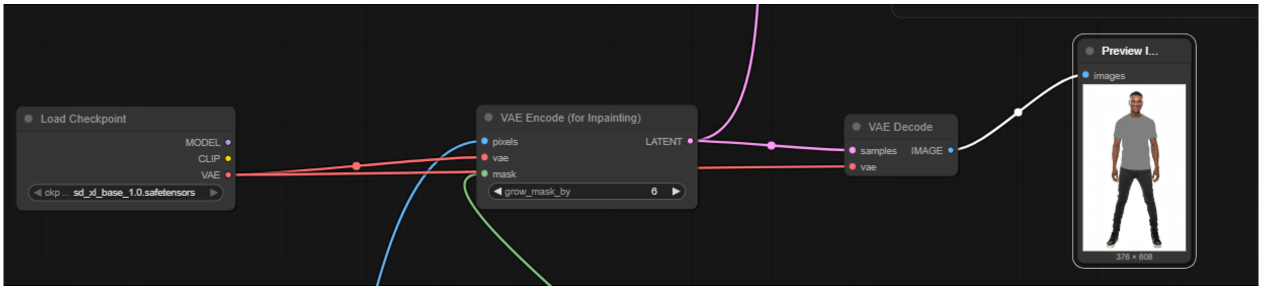
Grounding DINO acts as a "smart" object detector that uses natural language to find and localize objects in an image without needing predefined categories or specific training data for clothing.

- **Input:** You provide the input photograph and a specific text prompt, such as "the shirt" or "a red t-shirt".
- **Output:** The result of Grounding DINO is one or more bounding boxes around the detected object(s) that match your prompt. It provides a rough, but accurate, location of the clothing item.

SAM is a "promptable" segmentation model. It is not an object detector itself but excels at creating high-quality, pixel-level segmentation masks when given spatial prompts, such as points or, the bounding box from Grounding DINO.

- **Input:** SAM takes the original image and the bounding box coordinates generated by Grounding DINO as its input prompts
- **Output:** The final output is a segmentation mask: a binary image where the pixels corresponding to the clothing item are white, and everything else is black. This mask precisely isolates the clothing that has to be changed.

2) Encode of the image and decoding to create latents



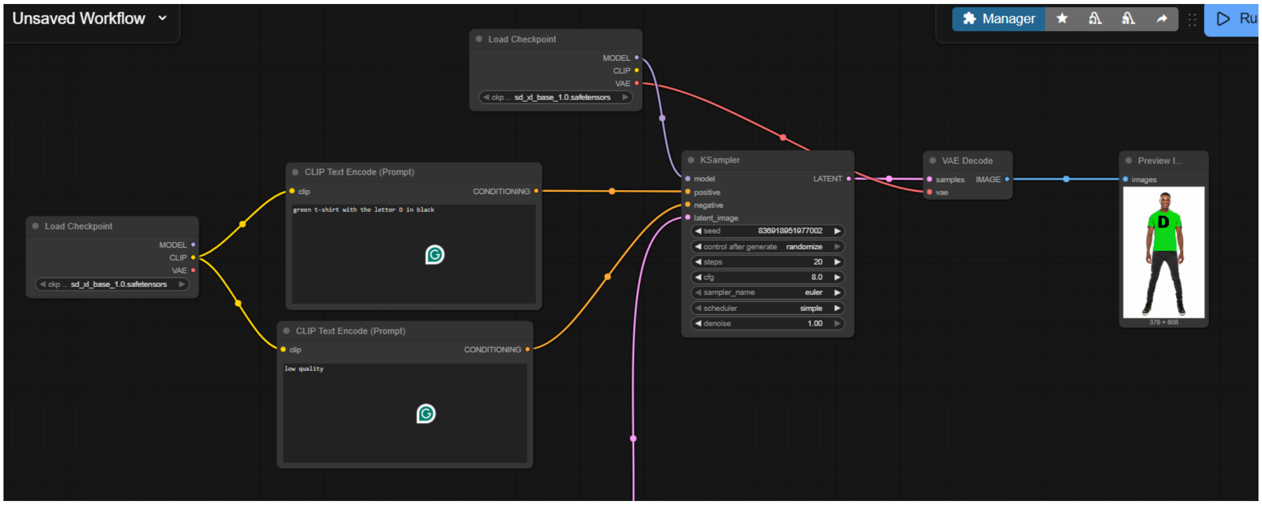
ckp ... sd_xl_base_1.0.safetensors

grow_mask_by 6

- The encoder is like a super-smart translator that takes a large, detailed image (which is too big for the AI to work with quickly) and shrinks it down into a tiny, compressed summary called a latent.
- The decoder is the reverse translator. It takes that tiny, coded latent summary and blows it back up into a full-sized, viewable photograph.

I use the mask to define the area of the image that I want to be potentially changed or modified during the diffusion process. This is strictly used for speed and for next steps in stable diffusion, because stable diffusion uses latents.

3) Creating the final image



ckp ... sd_xl_base_1.0.safetensors

ckp ... sd_xl_base_1.0.safetensors

green t-shirt with the letter D in black

low quality

seed 836918951977002
control after generate randomize
steps 20
cfg 8.0
sampler_name euler
scheduler simple
denoise 1.00

The Starting Point (Input Latent): It takes the existing latent representation of the image from Step 2, where the shirt area is marked for change.

The Goal (Positive Prompt): You tell it exactly what you want to see (e.g., "A vibrant red t-shirt, cotton texture, realistic lighting").

What to Avoid (Negative Prompt**):** You tell it what not to draw (e.g., "blurry, weird colors, low quality, bad anatomy").

The Preservation Zone (Masking): It ensures that the areas outside the masked shirt area are preserved exactly as they were in the original photo.

The KSampler doesn't start with a clean image; it starts with a base of random visual noise. This noise is carefully mixed with the input latent image, specifically concentrating the noise within the area of the shirt mask.

I don’t understand exactly what is happening from the encode-decode step. I read something, but I did not have enough time to understand completely. The next step would definitely be to understand this workflow from start to end, to understand the math and everything behind it. After that I would continue with Grounding Dino but implement my own model for clothes detection. There is a nice dataset I found, Deepfashion2, and I believe it would be easy to train a model on it and after that, implement the encode-decode step from scratch and the KSAMPLER.

Please let me know if I thought about the right next steps or if I am missing something. I will read more and understand exactly how they work. Thank you!!