Lab Assignment Report: Image-to-Prompt-to-Image Workflow

1. Failure cases

The five failure cases are shown as below:

- **Case 1:**
  - **Original Image:** **Failure Image:**

  - **Reasons:** The characteristic black-and-white stripes are missing. The horse's dark color remains largely unchanged, and its body structure, including the mane and legs, still looks like a horse.
- **Case 2:**
  - **Original Image:** **Failure Image:**

  - **Reasons:** The zebra pattern has been incorrectly applied to both the horse and the humans. Instead of selectively transforming only the horse, the model has transferred zebra-like stripes onto the people as well, making them look unnatural.
- **Case 3:**
  - **Original Image:** **Failure Image:**

- **Reasons:** The zebra stripes are applied inconsistently, particularly on the smaller animal. While the adult zebra has a somewhat realistic striped pattern, the foal retains a large portion of its original appearance, with only partial and distorted stripes.

- **Case 4:**
  - **Original Image:** **Failure Image:**

  - **Reasons:** The zebra's head is heavily distorted or missing, making the transformation look unnatural. The body has zebra-like stripes, but they appear artificially mapped and do not follow the natural muscle contours properly.

- **Case 5:**
  - **Original Image:** **Failure Image:**

  - **Reasons:** The stripes are mostly black, with unnoticeable white. The horse's overall body shape remains unchanged, particularly in the head and legs, making it still resemble a horse rather than a zebra.

2. Brief Description of the Process

- I refer to the workflow in https://www.runcomfy.com/comfyui-nodes/ComfyUI/Canny for ControlNet and canny settings. For preprocessing, I first resize the horse image to 512×512 pixels. Using SAM (Segment Anything Model), I extract both the mask and the segmented horse from the image.
- Next, I apply a custom Canny edge detection node in combination with ControlNet to detect and outline the edges of the extracted horse. With these edges as a reference, I input the text prompt "a photo of a zebra" to perform inpainting, generating a zebra that maintains the original shape and contours of the horse.
- Following this, I utilize KSampler and a VAE decoder to process and refine the generated images, resizing them to 256×256 pixels. Finally, I use ImageCompositeMasked to restore the original background while replacing the original horse with the newly generated zebra.

3. Missing Custom Nodes

- No nodes are missing. All nodes used in this assignment come from lab sheets and internal nodes of ComfyUI.

4. Checkpoint/Model(s) Used in the Workflow

List of Checkpoint/Model(s):

- **dreamshaper_8: Base model:** SD 1.5, used for text-to-image generation.
  - **https://civitai.com/models/4384/dreamshaper**
- **absolutereality_v181:** Base model: SD 1.5, used for hyper-realistic image generation with lifelike details.
  - **https://civitai.com/models/144952/mohawk**
- **sam_hq_vit_h.pth:** Base model: Segment Anything Model (SAM). SAM is specifically designed for image segmentation tasks, offering an ideal balance between model size and performance. It excels at accurately segmenting objects in complex scenes while maintaining computational efficiency. This makes it highly suitable for applications that demand both precision and speed, such as automated image editing and improving content accessibility.
  - **https://dl.fbaipublicfiles.com/segment_anything/sam_vit_l_0b3195.pth**
- **GroundingDINO_SwinB:** Base model: GroundingDINO. GroundingDINO is an open-vocabulary detection framework built on the Swin Transformer Base architecture, with a file size of 938MB. It excels at precise object localization by deeply integrating textual and visual features, making it highly effective for open-set detection tasks involving complex semantic descriptions, such as natural language-guided object recognition. This model delivers high accuracy in applications like e-commerce image annotation and medical imaging analysis. When combined with SAM, it supports English text prompts like "skirts" or "person with green dress" for targeted segmentation.
  - **https://github.com/IDEA-Research/GroundingDINO**

- **control_mlsd-fp16.safetensors:** Base model: ControlNet. The model is a ControlNet checkpoint designed to enhance Stable Diffusion's image generation by integrating MLSD (Mobile Line Segment Detection). This model specializes in detecting straight lines within images, making it particularly useful for architectural designs, structured compositions, and scenes with prominent linear features.
    - **https://huggingface.co/webui/ControlNet-modules-safetensors**

5. Regenerated images by using self-proposed workflow

The regenerated five cases are shown as below:

- **Case 1:**
    - **Original Image:** **Re-generated Image:**

- **Case 2:**
    - **Original Image:** **Re-generated Image:**

- **Case 3:**
    - **Original Image:** **Re-generated Image:**

- **Case 4:**
  - **Original Image:** **Re-generated Image:**

- **Case 5:**
  - **Original Image:** **Re-generated Image:**