Documentation for

Scene Manipulation

via Text-Controlled Object Relighting and Relocation

Maha Qaiser
22i-2348

Thought Process

1. Test and understand SAM
2. Test and understand DETR
3. Test and understand DETR + SAM combined
4. Experiment with text instruction parsing
5. Move onto solution implementation + carry out experiments

Solution

Step 1: Load an Image

- A file dialog opens so the user can select an image from their computer.
- The image is loaded using both PIL (for Transformers) and cv2 (for OpenCV and SAM).

Step 2: Parse Text Instruction

- User types an instruction like: "Move the dog to the right and add sunset lighting."
- The instruction is processed using spaCy NLP to extract:
    - The target object (noun) – e.g., “dog”
    - The action (verb) – e.g., “move”
    - The direction – e.g., “right”
    - The lighting – e.g., “sunset”
- These are stored for further use.

Step 3: Object Detection with DETR

- Uses DETR (DEtection TRansformer) to find objects in the image.
- DETR returns a bounding box for the target object and object label.

Step 4: Get Object Mask using SAM

- Uses Segment Anything Model (SAM) to generate masks.
- The point at the center of the detected box from DETR is given as input to SAM.
- SAM returns 3 masks — the one with the best combination of confidence and area is chosen.

Step 5: Cut out the Object

- Using the selected mask:
    - All object pixels are copied into a blank image (black background).
    - Then it is cropped tightly to include only the object.

Step 6: Inpaint to Remove Original Object

- The selected object is removed from the original image using OpenCV's inpaint function.
- This fills the object's region with a smooth background.

Step 7: Calculate New Location

- Based on the user's instruction (left/right/up/down/center):
    - A new top-left position is calculated where the object will be pasted.
    - Basic boundary checks are applied to keep the object inside image limits.

Step 8: Paste Object at New Location

- The cropped object is pasted onto the inpainted image using the mask:
    - Only the masked pixels from the cropped object are pasted.

Step 9: Visualize All Results

- Five side-by-side subplots are shown:
    1. Original image with bounding box
    2. Mask over the object (SAM)
    3. Cropped object
    4. Inpainted image (object removed)
    5. Final image with object moved

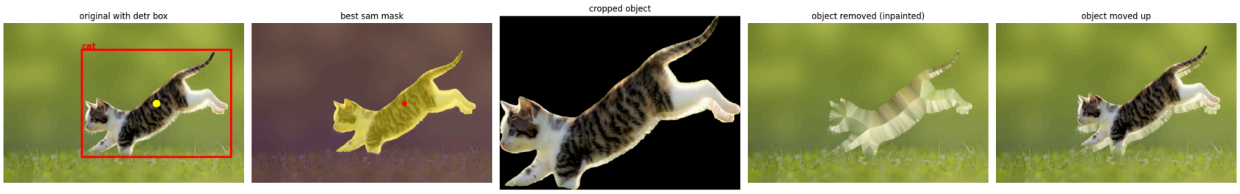
Features

✔️ Text Instruction Parsing
✔️ Object Detection
✔️ Object Segmentation
✔️ Object Relocation
❌ Relighting

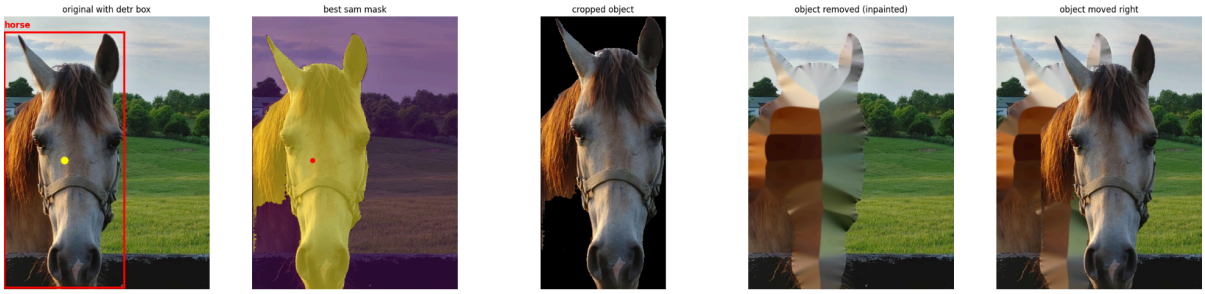
Test Outputs

Instruction: Move the dog to the left and add sunset light

Instruction: Move the cat up



Instruction: Move the horse to the right side



Challenges & (some) Solutions

- Mask selection
    - Used a weighted score (score × area) to select the most confident and visually complete SAM mask from multiple outputs.
- Diffusion with AI
    - Stable Diffusion often generated unrealistic, overly artistic results instead of subtle, clean edits. It lacked fine control, especially for object-level changes, so I used OpenCV's classical inpainting method.
- Relighting
    - Tried both Stable Diffusion and Neural Gaffer for lighting edits, but neither worked effectively. Prompts failed to guide lighting direction, and outputs were inconsistent. Neural Gaffer was also difficult to set up and repeatedly threw errors that I could not resolve.

Improvements

To improve this solution:

- Explore alternative diffusion methods for more stable outputs.
- Use alternate relighting techniques, such as image filters or lightweight learned relighting methods.
- Enhance NLP parsing by integrating a more advanced interpreter (e.g., GPT) for better understanding of multi-part instructions.