Avataar Assignment Report

Kunind Sahu
kusahu@ucsd.edu

December 23, 2024

1 Introduction

The goal is to use pre-trained model and combine them to form an application such that it can segment an object (defined by a user-given class prompt) in a given scene. The second task is to change the position of the object in the same scene and make a composite image. The code for this task can be found here: https://github.com/kunind27/Scene-Composition

2 Methodology

2.1 Text-guided Object Detection and Segmentation

We wish to segment an object of a given class. It is but obvious that Segment Anything (SAM) [1] should be used, since it is a foundation model for semantic segmentation. Unfortunately, SAM cannot take in textual prompts. So, we use a composite strategy where we use Grounding DINO [2], to allow text-guided object detection, and then we use the resultant bounding boxes, and feed it into SAM for mask-conditioned semantic segmentation.

2.2 Object Repositioning in 2D

We take the resultant bounding box from Grounding DINO and convert it into a mask. We then use ControlNet [3] guided Image Inpainting through Stable Diffusion [4] to fill in the details in the masked out region (i.e where the object originally was). Then we compute a shifted mask based on the shift provided by the user. Then, on the basis of the new mask, we use ControlNet guided Image Inpainting through Stable Diffusion, again, but this time, conditioned on the object image. The hope is that the second inpainting step can fill in the masked region with the details of the object, thus effectively repositioning it.

3 Results and Analysis

We ran the above methodology for all the provided images. We show results of repositioninig by an input $(x, y) = (20, 0)$.

3.1 Bagpack

The outputs of various intermediate stages of the pipeline are shown in fig. 1. The repositioning fails primarily because Grounding DINO fails to regress the correct bounding boxes for the bagpack and instead detects the table as a bagpack instead.
Even with the table as the object, the repositioning is still not smooth. The reason for this is the output of the first inpainting stage as seen in fig. 1(e). The first inpainting stage does not effectively fill-up the mask with the context of the areas surrounding the masks.

(a) Original Image (b) Repositioned Image (c) SAM Mask on Object

(d) G-DINO Bounding Box (e) First Inpainting Stage

Figure 1: Outputs of Different Stages of the Pipeline for Class: `'Bagpack'`

3.2 Wall Hanging

The outputs of various intermediate stages of the pipeline are shown in fig. 2. Grounding DINO is able to regress the correct bounding boxes for the wall hanging. On the basis of the bounding boxes, SAM is able to give a great segmentation result for the wall hanging
The repositioning fails because of the unsmooth output of the first inpainting stage as seen in fig. 2(e). The first inpainting stage does not effectively fill-up the mask with the context of the areas surrounding the masks in this image either.

(a) Original Image (b) Repositioned Image (c) SAM Mask on Object

(d) G-DINO Bounding Box (e) First Inpainting Stage

Figure 2: Outputs of Different Stages of the Pipeline for Class: `'wall hanging'`

3.3 Stool

The outputs of various intermediate stages of the pipeline are shown in fig. 3. Grounding DINO is able to regress the correct bounding boxes for the stool. Based on the bounding boxes, SAM is able to give a great segmentation result for the stool as well.
The repositioning fails because of the unsmooth output of the first inpainting stage as seen in fig. 3(e). The first inpainting stage does not effectively fill-up the mask with the context of the areas surrounding the masks in this image either.

(a) Original Image (b) Repositioned Image (c) SAM Mask on Object

(d) G-DINO Bounding Box (e) First Inpainting Stage

Figure 3: Outputs of Different Stages of the Pipeline for Class: `'stool'`

3.4 Comments

From these examples, it can be inferred that, while grounding-DINO augmented SAM is a satisfactory solution for text-guided segmentation. Image Inpainting is still the root of problems in object repositioning. More specifically, it is the first stage of inpainting that is to blame. The results even after the second inpainting stage are satisfactory, especially considering the output it was conditioned on (the output of the first inpainting stage and the object image).

I very closely followed the API description and tutorials provided in the Diffusers [5] documentation for ControlNet-guided inpainting and even toyed around with the Guidance Scale Hyperparameters (both CFG and ControlNet) for rectifying the first inpainting stage, but to no avail. A deeper dive into the API documentation, API implementation, and hyperparameter selection might be warranted.

I also tried setting the Guidance Scale for ControlNet to zero, effectively removing ControlNet guidance and converting it into a simple Inpainting Task. However, the resultant inpainting was heavily biased towards the shape of the original objects, which seems absurd, since the mask I am providing is a bounding box around the objects themselves and not the SAM segmentation map, which outlines the objects' shapes as well. These observations make a deep dive into the API necessary.

4 Methods of Improvement

4.1 Rectifying First Inpainting Stage

Experiment with a different StableDiffusion Model or with an API other than Diffusers for the Inpainting task.

4.2 Text-guided Segmentation

Although Grounding-DINO and SAM work well together, there is still much room for improvement, especially, with the advent of Foundational Models for Vision-related tasks. Here are some possible techniques:

- Use recent works such as EVF-SAM [6], that have enabled text prompted segmentation within the SAM framework
- Use Multimodal Foundation Models for segmentation related tasks such as YOLOWorld [7], Florence-2 [8] etc.

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[2] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[3] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," 2023.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

[5] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf, "Diffusers: State-of-the-art diffusion models." https://github.com/huggingface/diffusers, 2022.

[6] Y. Zhang, T. Cheng, R. Hu, L. Liu, H. Liu, L. Ran, X. Chen, W. Liu, and X. Wang, "Evf-sam: Early vision-language fusion for text-prompted segment anything model," 2024.

[7] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.

[8] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," 2023.