Segment-and-Sticker: Object-Aware Transparent Cutout Generation Using SAM and Open-Vocabulary Detection

Vedanta Gawande
University of Maryland
8125 Paint Branch Dr,
College Park, MD
vgawande@terpmail.umd.edu

Jai Bobal
University of Maryland
8125 Paint Branch Dr,
College Park, MD
jaibobal@terpmail.umd.edu

Mridul Purkayastha
University of Maryland
8125 Paint Branch Dr,
College Park, MD
mridul@terpmail.umd.edu

Abstract

In recent years, there have been a considerable number of advancements in the vision-language scene. Segmentation architectures have enabled us to have a more intuitive interaction with visual content, opening horizons to many new possibilities. We discuss one such adaptation through our project, where we generate a pipeline that leverages Meta's Segment Anything Model (SAM) and OWL-ViT (Object detection with Vision-Language Transformers) to generate PNG stickers from user images, where the user also provides a natural language prompt. Our system allows users to input an image and a textual description of the desired object within the image. Many objects of the description are located via OWL-ViT, and the one with the maximum confidence is passed into SAM for segmentation. The segmented mask is then extrapolated into a PNG without any other pixels. Our framework is similar to the current zero-shot object detection and zero-shot segmentation delivered by Apple's sticker-generation feature on iPhones. However, we go a mile further by providing the capability to derive stickers of secondary subjects through a natural language input. We demonstrate that our method generalizes well across diverse images and object categories, and we highlight its potential for applications in digital communication and localized photo editing.

1. Introduction

Visual content plays a central role in how we communicate online. From social media posts and messaging apps to augmented-reality filters, users increasingly seek intuitive tools that let them isolate and manipulate objects within photographs.

1.1. Past Research

On the other side, there has also been vast research areas that aid such image manipulation. First, Meta's Segment Anything Model (SAM) provides a powerful segmentation backbone that can carve out intricate masks around subjects in an image. Second, transformers such as OWL-ViT enable open-vocabulary object detection by amalgamating text input with binding boxes in images.

1.2. Working Context

In our work, we build upon both technologies to introduce a lightweight framework where a user simply provides an image, along with a textual description such as "dog," "car," or "cake". This prompt, along with the image is passed into OWL-ViT, which locates the best matching region. That region is then passed to SAM, which outputs a segmentation mask. Finally, we assemble an RGBA image that retains only the selected pixels and saves them as a standalone PNG sticker.

1.3. Our Contributions

Our main contributions are:

- Hybrid zero-shot pipeline. We showcase seamless integration of open-vocabulary detection using OWL-ViT with segmentation with SAM for flexible sticker generation.
- Natural-language control. Unlike existing smartphone features, our system can target secondary or background objects simply by changing the prompt.
- Practical implementation. We deliver a concise, colab-ready codebase that works across diverse object categories and image conditions, highlighting the method's accessibility for students and hobbyists.

1.4. Overview of the Paper

The rest of the paper is organized as follows. Section 2 motivates our design with user scenarios and also highlights our motivation to pursue this project. Section 3 reviews related work in segmentation and vision-language detection. Section 4 details the technology behind SAM and OWL-ViT and our integration strategy. Section 5 describes our methodology and implementation. Section 6 presents experiments on varied images and prompt types.

In Section 7, we discuss challenges, limitations, and future directions, and Section 8 concludes.

2. Motivation

With the rise in online communication, there is a need for more ways to be expressive. One such way is visual communication, which has made it easier than ever to share moments, ideas, and creativity through stickers and cutouts.

However, existing solutions, such as the one offered in the Photos app by Apple, where a user holds an object to generate the desired sticker, is tightly bound to only being able to identify a primary object. There is no straightforward way to extract any other object from the same image.

In practical scenarios, users often want more fine-grained control. Perhaps a user wishes to select a smaller object in the background as their sticker, and not the person in the foreground.

Since other sticker generators do not support this level of flexibility, our team found motivation to build a system where this could be achieved. We went a step ahead to make this more accessible by allowing natural language input to obtain the desired object's sticker. This capability unlocks richer creative workflows and increases the accessibility of sticker-style editing.

3. Related Works

Our Segment-and-Sticker framework utilized advanced technologies undergoing active research.

Kirillov et al. (2023) introduced Meta's Segment Anything Model (SAM), which demonstrated production of high-quality masks for any region, without per-class training. SAM accepts points, boxes, or masks as prompts, achieving zero-shot performance across a wide range of objects.

Moreover, vision-language transformers like OWL-ViT, enable zero-shot detection of regions in an image from text prompts. These models leverage CLIP-style embeddings.

In one of our class projects, Project 6, we got first hand experience with SAM. We prompted SAM with user-selected points to showcase success and failure cases.

That exercise highlighted two key things:

- Prompt Sensitivity: SAM's mask quality can vary dramatically depending on point placement.
- Integration Opportunity: Neither a segmentation-only nor a generation-only pipeline directly supports text-driven, object-specific cutouts.

Our work addresses this by inculcating OWL-ViT with SAM, providing us with a robust pipeline, where we can generate stickers through text and input images in a single pass.

4. Methodology

Our sticker-generation pipeline is implemented end-to-end in Python on Google Colab. Our primary implementation is organized into two main functions: bounding_box_detection and generate_segmask.

This section discusses how our code works and is put together.

4.1. Environment setup and model instantiation.

We begin by installing our dependencies (segment-anything, transformers, opencv-python, matplotlib) using pip. Once the libraries are available, our script imports PyTorch, NumPy, and PIL's Image and ImageOps. Next, we get in the OwlViTProcessor and OwlViTForObjectDetection from HuggingFace's Transformers library, and sam_model_registry along with SamPredictor from Meta's Segment Anything repository.

For the device selection, we check if PyTorch's CUDA is available by running torch.cuda.is_available() and create a device handle (device = torch.device("cuda")) so that subsequent model loads and inferences run on GPU.

With our device determined, we initialize the vocabulary detector by calling OwlViTProcessor.from_pretrained("google/owlvit-base-patch32") to load the text/image tokenizer and OwlViTForObjectDetection.from_pretrained(...) to load the model weights.

We immediately transfer the detector to GPU via to(device). In parallel, we register the SAM checkpoint (sam_vit_h_4b8939.pth) with sam_model_registry["vit_h"](checkpoint=…), move it to GPU, and wrap it in a SamPredictor object, which will accept images and generate segmentation masks.

4.2. bounding_box_detection(inputimg, prompt, threshold).

This function locates the user-specified object in the image:

- **Image loading and preprocessing**:

We open the file named by inputimg, convert to RGBA with, and correct for any rotated EXIF metadata. We then convert this PIL image to a NumPy array.

- **Text–image encoding and inference**:

We pass in the user prompt and the image into a HuggingFace batch by calling processor(text=[prompt], images=[img_pil], return_tensors="pt").

- **Post-processing and thresholding**:

We construct a tensor with a single element called target_sizes, which matches the image's height and width to guide the processor's resizing logic. We pass these outputs and target_sizes into processor.post_process_object_detection(...) with our confidence threshold to remove low confidence matches. The result is a dictionary containing "boxes" and "scores".

- **Result handling and selection**:

We print how many boxes survived thresholding. If none remain, we say that no matches were found to notify the caller that no object matched the prompt. Otherwise, we find the index of the highest score, cast the corresponding box coordinates to integers, print the chosen box and its score for transparency, and return the edge coordinates.

4.3. generate_segmask(img, box, imgpath="sticker.png")

- **Segmentation prompt setup**:

We instruct SAM to preprocess the image using predictor.set_image(img).

- **Mask prediction**:

We perform a single-box segmentation request via predictor.predict(box=np.expand_dims(box, axis=0), multimask_output=False). SAM returns along with a confidence score we ignore here.

- **Alpha-channel construction**:

The mask is converted to an 8-bit array alpha = (mask.astype(np.uint8) * 255).

- **RGBA assembly and saving**:

We concatenate the original RGB image and the alpha mask along the channel axis (np.dstack([img, alpha])). Wrapping this in Image.fromarray(...) produces a PIL image with transparency. We then save the result to a disk. These two functions allow for a modular application which can be reused for many images. This creates a ready to use pipeline combining SAM and OWL-Vit to create a sticker generator.

5. Experimentation

We tested our application through many hurdles. These include ambiguous prompts, choosing different confidence thresholds, complex images, multi-object detection, and false prompts. In each experiment we measured whether OWL-ViT successfully located the intended region and whether SAM produced a tight mask around it.

5.1. Ambiguous Prompts

With a fixed confidence threshold (0.01), we compared generic prompts ("car," "wheelchair sign," "person") against more descriptive ones ("a yellow car," "a wheelchair sign on the road," "a person in a hoodie"). We found that brief, one-word prompts often failed to yield any detections. More descriptive prompts were more reliable in producing bounding boxes. Hence we can see that OWL-ViT remains sensitive to wording, requiring extensive detail.

5.2. Threshold Variation

In our work, we build upon both of these technologies to introduce a lightweight framework where a user simply provides an image, along with a textual description such as "carrot," "car," or "people". This prompt, along with the image is passed into OWL-ViT, which locates the best matching region. That region is then passed to SAM, which outputs a segmentation mask. Finally, we assemble an RGB image that retains only the selected pixels and saves them as a standalone PNG sticker.

5.3. Complex Images

We tested on photographs with blur, grain, low resolution, and cluttered backgrounds. Here, OWL-ViT often missed even prominent subjects when details were obscured, and SAM could not compensate if no box was produced. In some cases, even when detection succeeded, the quality of masks by SAM degraded.

5.4. Multi-Object Detection

We tested images containing plural objects (e.g: two people). We tried detecting multiple subjects simultaneously, but results were inconsistent. Interestingly, multi-instance detection worked more reliably for inanimate, well-separated items (e.g. clusters of vegetables) than for people.

5.5. False Prompts Under Low Threshold

We assert a low threshold and ask OWL-ViT to detect objects not present in the input image. As expected, it began proposing random coordinates. SAM then produced masks around those arbitrary regions, resulting in useless

stickers.

Overall, we got to test our project's extremes and realize that our experiments reveal the need for prompt specificity, setting an appropriate threshold, choosing images that produce the desired results.

6. Discussion

In this section, we talk about some of the challenges we faced, the limitations of our implementation, and possible avenues for future improvement.

6.1. Challenges

Model selection and integration. We had to choose a detector that could reliably ground arbitrary text prompts. OWL-ViT offered easy HuggingFace integration but proved highly sensitive to prompt phrasing and occasionally failed on common nouns. We briefly experimented Grounding DINO but settled on OWL-ViT for its simplicity in Colab.

Threshold tuning. Our confidence threshold was a trade-off between the detector's ability to detect objects and its ability to detect correct objects. At high thresholds ($\geq$0.3), only the most exact matches passed through. At low thresholds ($\leq$0.005), we got many false positive matches. Finding a single working value that generalized across dozens of test images demanded extensive manual experimentation- and we settled on 0.01.

6.2. Limitations

Multiple-object detection. Our current design only selects the single highest-confidence object per prompt. Scenes with multiple valid instances are not supported yet (only one of them will be chosen, arbitrarily).

Sensitivity to image quality. Blurred, dark, or highly cluttered images frequently produced bad results.

False positives at low thresholds. While lowering the threshold improved recall for faint or secondary objects, it also produced false positive matches through unrelated regions.

6.3. Future Avenues

Multi-object selection. In future, as more research is put into advancing SAM, OWL-ViT, and other related models, we would be able to build a pipeline to return multiple candidate boxes and masks per prompt, allowing us to match and retrieve several objects in one go.

Background change via diffusion. After extracting the PNG, we could feed it into a text-to-image diffusion model (e.g. Stable Diffusion) to generate new backgrounds or stylized contexts in a new scene.

7. Conclusion

In our project, we generated Segment-and-Sticker, a framework that combines OWL-ViT with SAM to produce transparent cutouts from arbitrary images. Our application allows users to specify any object through a simple, natural-language prompt, which gets fed into our pipeline along with an image of their choice. Our pipeline then allows fine-grained selection of both primary and secondary subjects. Through extensive experiments on prompt variations, detection thresholds, complicated images, we demonstrated the trade-offs between precision and recall.

Looking ahead, our app offers many avenues for richer photo-editing applications. Our concise Colab implementation makes our app accessible, and we believe this work highlights the creative potential unlocked by bridging vision-language grounding with promptable segmentation.

8. Project Work Distribution

Our work was streamlined by each of us working like in an assembly line. Initially, we wrote the code for our application together. Vedanta helped by researching resources we used, Mridul typing it out, and Jai experimenting and calculating the edge cases where our code failed. When it came to writing the paper and finalizing our application, Jai helped in writing the content, Vedanta helped formatting it into a document and proofreading, and Mridul helped beautify our Colab notebook and presenting it for the video section of this project. Our work was split equally and we all contributed to the product.

References

[1] Minderer, Matthias, et al. "Simple open-vocabulary object detection." European conference on computer vision. Cham: Springer Nature Switzerland, 2022.

[2] Kirillov, Alexander, et al. "Segment anything." Proceedings of the IEEE/CVF international conference on computer vision. 2023.