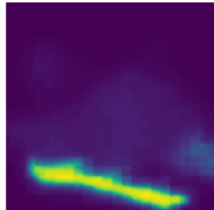


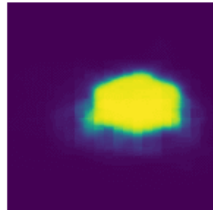
ZERO-SHOT IMAGE SEGMENTATION WITH CLIPSEG



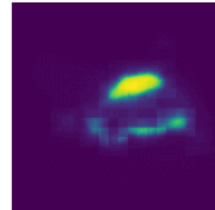
cutlery



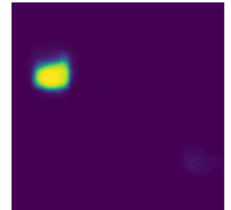
pancakes



blueberries



orange juice



Introduction

Image segmentation is a well-known task within the field of computer vision. It allows a computer to not only know what is in an image (classification), where objects are in the image (detection), but also what the outlines of those objects are. Knowing the outlines of objects is essential in fields such as robotics and autonomous driving.

One limitation of most image segmentation models is that they only work with a fixed list of categories. For example, you cannot simply use a segmentation model trained on oranges to segment apples. To teach the segmentation model an additional category, you have to label data of the new category and train a new model, which can be costly and time-consuming. But what if there was a model that can already segment almost any kind of object, without any further training? That's exactly what CLIPSeg, a zero-shot segmentation model, achieves.

CLIP: the magic model behind CLIPSeg

CLIP, which stands for Contrastive Language–Image Pre-training, is a model developed by OpenAI in 2021. You can give CLIP an image or a piece of text, and CLIP will output an abstract *representation* of your input. This abstract representation, also called an *embedding*, is really just a vector (a list of numbers). You can think of this vector as a point in high-dimensional space. CLIP is trained so that the representations of similar pictures and texts are similar as well. This means that if we input an image and a text description that fits that image, the representations of the image and the text will be similar (i.e., the high-dimensional points will be close together).

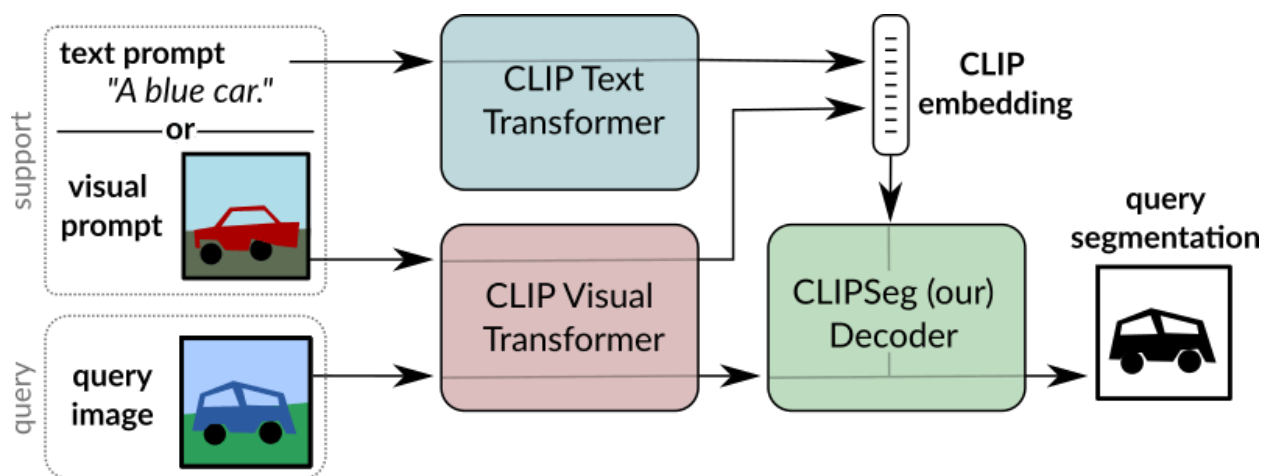
At first, this might not seem very useful, but it is actually very powerful. As an example, let's take a quick look at how CLIP can be used to classify images without ever having been trained on that task. To classify an image, we input the image and the different categories we want to choose from to CLIP (e.g. we input an image and the words "apple", "orange", ...). CLIP then gives us back an embedding of the image and of each category. Now, we

simply have to check which category embedding is closest to the embedding of the image, et voilà! Feels like magic, doesn't it?

The reason why CLIP works so well is that the model was trained on a huge dataset of images with text captions. The dataset contained a whopping 400 million image-text pairs taken from the internet. These images contain a wide variety of objects and concepts, and CLIP is great at creating a representation for each of them.

CLIPSeg: image segmentation with CLIP

CLIPSeg is a model that uses CLIP representations to create image segmentation masks. It was published by Timo Lüddecke and Alexander Ecker. They achieved zero-shot image segmentation by training a Transformer-based decoder on top of the CLIP model, which is kept frozen. The decoder takes in the CLIP representation of an image, and the CLIP representation of the thing you want to segment. Using these two inputs, the CLIPSeg decoder creates a binary segmentation mask. To be more precise, the decoder doesn't only use the final CLIP representation of the image we want to segment, but it also uses the outputs of some of the layers of CLIP.



The decoder is trained on the PhraseCut dataset, which contains over 340,000 phrases with corresponding image segmentation masks. The authors also experimented with various augmentations to expand the size of the dataset. The goal here is not only to be able to

segment the categories that are present in the dataset, but also to segment unseen categories. Experiments indeed show that the decoder can generalize to unseen categories.

One interesting feature of CLIPSeg is that both the query (the image we want to segment) and the prompt (the thing we want to segment in the image) are input as CLIP embeddings. The CLIP embedding for the prompt can either come from a piece of text (the category name), or from another image. This means you can segment oranges in a photo by giving CLIPSeg an example image of an orange.

Approach

We start by installing the transformers.

Then we download a pre trained CLIPSeg model, by simply instantiating it.

Next we load an image from the Phrasecut dataset to try out segmentation.

Then we provide prompts by defining text categories that we want to segment

Now that we have our inputs in the form of image and text prompts, we can process them and input them to our model

Finally we visualize the output using matplotlib.pyplot

Conclusion

CLIPSeg is a zero-shot segmentation model that works with both text and image prompts. The model adds a decoder to CLIP and can segment almost anything. However, the output segmentation masks are still very low-res for now, so you'll probably still want to fine-tune a different segmentation model if accuracy is important.