# Region-based Image Similarity Search

# Contents

# 1 Region-based Image Retrieval

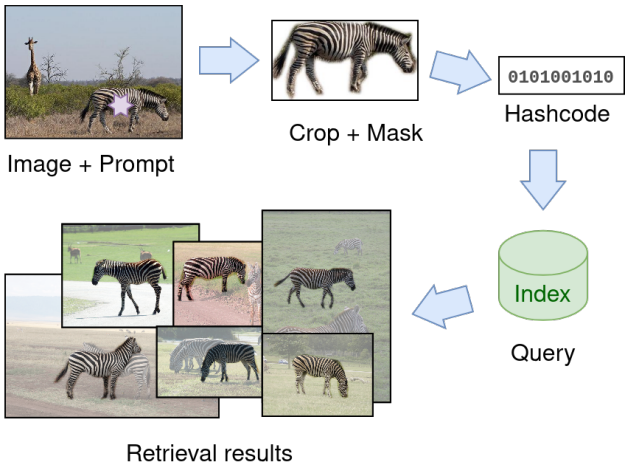


Figure 1: Example query image with a point prompt for segmentation-based similarity search.

Our approach allows users to better express their search intent by enabling a more refined search for objects or image regions. This search intent can be conveyed via prompts, such as point, box, and text prompts. Fig. 1 shows how a query is processed in our approach, called *Search Anything*. First, a user selects a query image. Then, the user selects an image region of this query image via a prompt. The selected image region is automatically segmented. Next, a binary feature vector is extracted based on the masked region. This feature vector is used as a query for an image region index, and the user gets back as a result the images that contain the corresponding region.

# 2 Model

Figure 2 shows our approach, called *Search Anything*, for generating compact binary codes for image regions in a query image via prompts. First, a user chooses a query image that

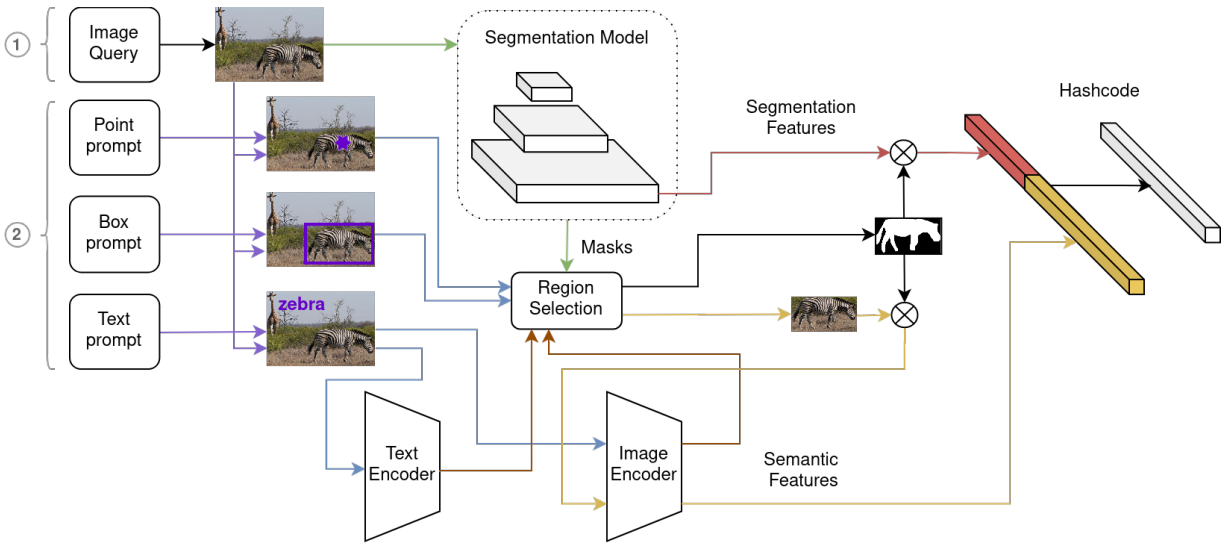


Figure 2: **Similarity search with region prompts.** First, a user selects a query image (1). Then, the user refines the search with a region prompt (2). Segmentation features and semantic features are masked and concatenated. The combined feature vector is compressed by deep hashing.

contains an object or region of interest. This query can then be refined by the user via various prompts, such as point prompt, box prompt, or text prompt (violet). The query image is then segmented by a foundation model for image segmentation. The segmentation masks of the query image are used in the subsequent region selection step (green). It selects the corresponding query object or region according to the user's prompt (blue). In case of a text prompt, an image-to-text foundation model is used to match the text encoding (brown) to one of the query image's segmentation masks. The image encoder of the image-to-text model is then used again for obtaining generalized image features (yellow) for the selected region. Depending on the size of the region, the input to the model is masked with the selected segmentation mask ($\otimes$ in Fig. 2 denotes pixel-wise multiplication). The same mask is used when creating a segmentation feature vector (red), but here, instead of masking the input, the feature map extracted from the feature pyramid [1] of the segmentation model is

masked. Then, the semantic feature vector is concatenated with the extracted segmentation feature vector of the segmentation model. Finally, the concatenated feature vector is fed into a First, a user chooses a query image that contains an object or region of interest. This query can then be refined by the user via various prompts, such as point prompt, box prompt, or text prompt (violet). The query image is then segmented by a foundation model for image segmentation. The segmentation masks of the query image are used in the subsequent region selection step (green). It selects the corresponding query object or region according to the user's prompt (blue). In case of a text prompt, an image-to-text foundation model is used to match the text encoding (brown) to one of the query image's segmentation masks. The image encoder of the image-to-text model is then used again for obtaining generalized image features (yellow) for the selected region. Depending on the size of the region, the input to the model is masked with the selected segmentation mask (⊗ in Fig. 2 denotes pixel-wise multiplication). The same mask is used when creating a segmentation feature vector (red), but here, instead of masking the input, the feature map extracted from the feature pyramid [1] of the segmentation model is masked. Then, the semantic feature vector is concatenated with the extracted segmentation feature vector of the segmentation model. Finally, the concatenated feature vector is fed into a deep hashing neural network that returns a compact binary representation of the combined mask and semantic image features for the image region. Finally, the obtained region-of-interest hashcode is used to efficiently search the database of image regions.

a foundation model for image segmentation

## 2.1 Region Prompts

To process the prompts and extract the segmentation masks for selected regions, we use the FastSAM [2] approach for dividing the segment anything task into two stages. In the first

stage, the masks for all regions in the query image are generated using a CNN-based object detector (i.e., All Instance Segmentation). In the second stage, the corresponding mask is assigned to the prompt (i.e., Prompt-guided Selection). The generation of segmentation masks for all instances in the image is performed by the segmentation model's backbone, which is a YOLOv8 object detector [3]. Both stages are summarized in Fig. 2 as *Region Selection*; they enable the utilization of point prompts, box prompts, and text prompts. While in SAM prompts are part of the transformer-based architecture as inputs, in FastSAM prompts are processed after segmentation has been performed. But image-based prompting works the same way: as with SAM, foreground and background points can be set. If a foreground point lies in several masks, background points are used to exclude irrelevant masks, and multiple foreground points are used to merge segmentation masks into a single mask. With a box prompt, a user can draw a box around a region-of-interest. By matching the Intersection over Union (IoU), the corresponding mask is assigned. Finally, text prompts offer text-based queries to specify a region-of-interest within the query image. Text and image embeddings are extracted from CLIP, and the masked features with the highest similarity score to the text embedding are selected. While text prompt processing is the most expensive step at inference time, the image embeddings can be used directly for the following semantic feature extraction step. In the context of region-based retrieval, these different prompts are used to specify regions as queries, and we therefore refer to them as *region prompts*. Based on the selection of a segmentation mask for the region prompt, we extract segmentation features for the corresponding prompt region, from the highest resolution level of the feature pyramid of the segmentation model and semantic features from the image encoder.

### 2.2 Deep Hashing

We use a hashing layer for compressing masked region features. It generates a compact representation of length $L$ for a given image region $x \in \mathbb{R}^3$ as follows:

$$hash_L(x) = tanh(W(f_M(x) \oplus f_I(x)) + b), \tag{1}$$

where $\oplus$ denotes concatenation. The segmentation features $f_M$ are obtained as follows. From the highest resolution layer of the feature pyramid of the segmentation model, each channel is multiplied with the predicted segmentation mask and then average-pooled. This results in a 320-dimensional feature vector (corresponding to the number of channels of the highest resolution layer). The semantic feature vector $f_I$ is extracted from the image encoder of the CLIP model by feeding it with the masked image crop. To obtain robust image features, we use OpenCLIP ViT-H-14 [4] that showed a high zero-shot performance and is robust to natural distribution shifts [5]. The image feature vector is 1024-dimensional.

## 3 Examples

### 3.1 Region Prompts

Fig. 3 demonstrates that our approach can be used to perform a fine-granular region similarity search. The queries are performed on the PASCAL VOC dataset. For each image, the 25 largest regions were used for indexing regions with 256-bit codes. Object classes such as *wheel* or *jeans* are not part of the PASCAL VOC class lexicon, but can be searched for via region prompts using our approach.

Only box prompts for image regions are considered in this example, but other types of region prompts work in the same way.

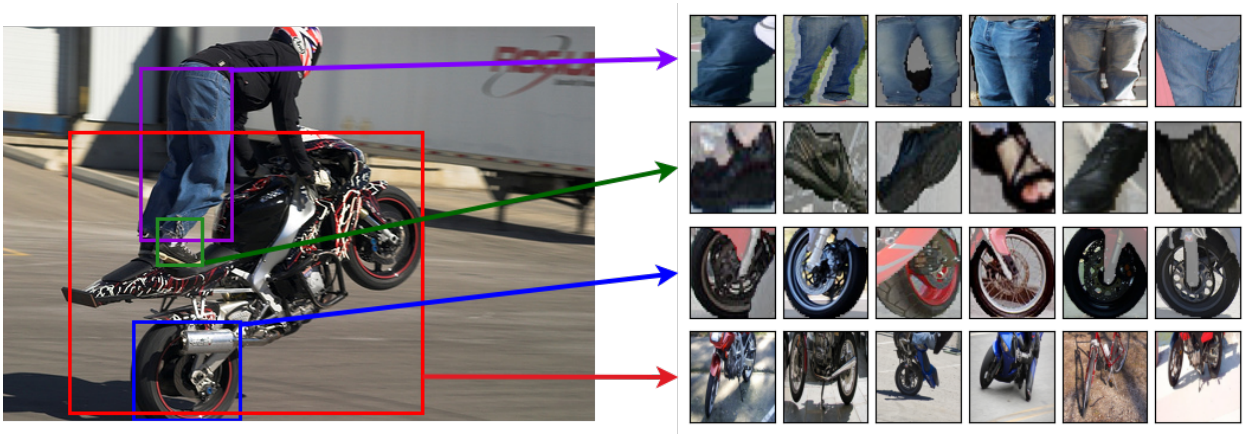

Figure 3: Example query image with region prompts and top 6 results performed on PASCAL VOC [6]

## 3.2 Features

We give some more examples where we qualitatively evaluate the effect of using masking and mask features compared to using CLIP features alone.

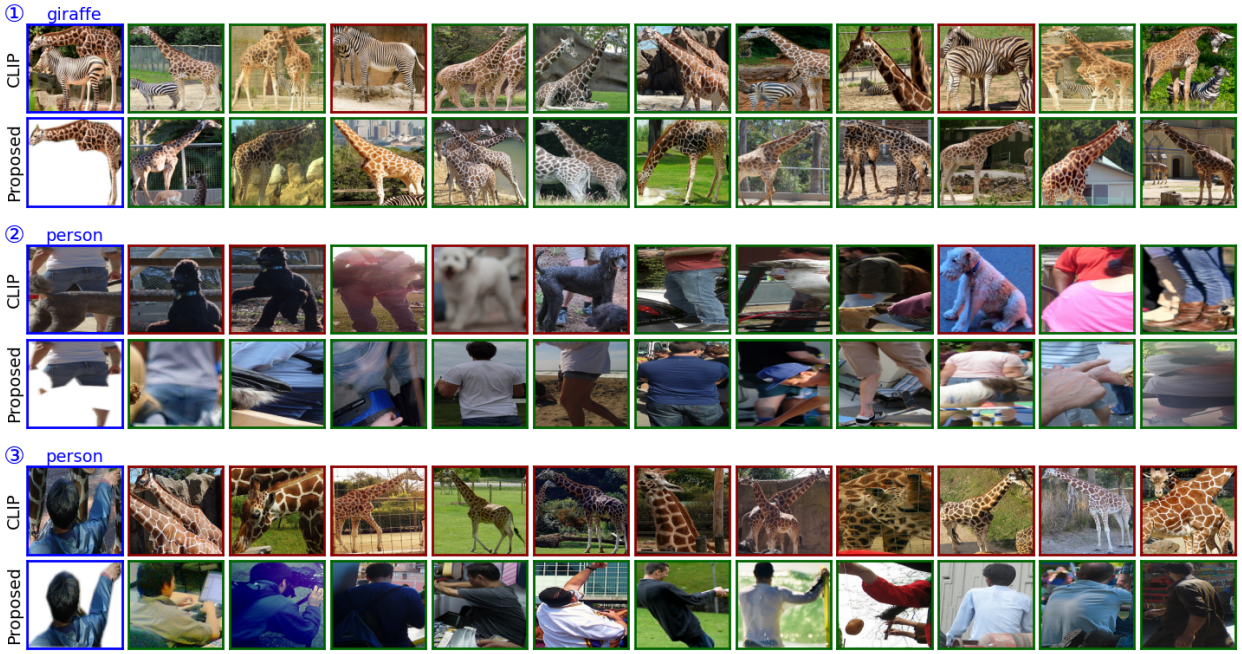


Figure 4: Ambiguous box prompt 3 / 3

# References

[1] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2117–2125.

[2] Xu Zhao et al. “Fast Segment Anything”. In: *arXiv preprint arXiv:2306.12156* (2023).

[3] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *YOLO by Ultralytics*. Version 8.0.0. Jan. 2023. URL: `https://github.com/ultralytics/ultralytics`.

[4] Gabriel Ilharco et al. *OpenCLIP*. Version 0.1. July 2021. DOI: `10.5281/zenodo.5143773`. URL: `https://doi.org/10.5281/zenodo.5143773`.

[5] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[6] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *International Journal of Computer Vision* 88 (2010), pp. 303–338.