Open-Vocabulary Segmentation Report

Task Overview

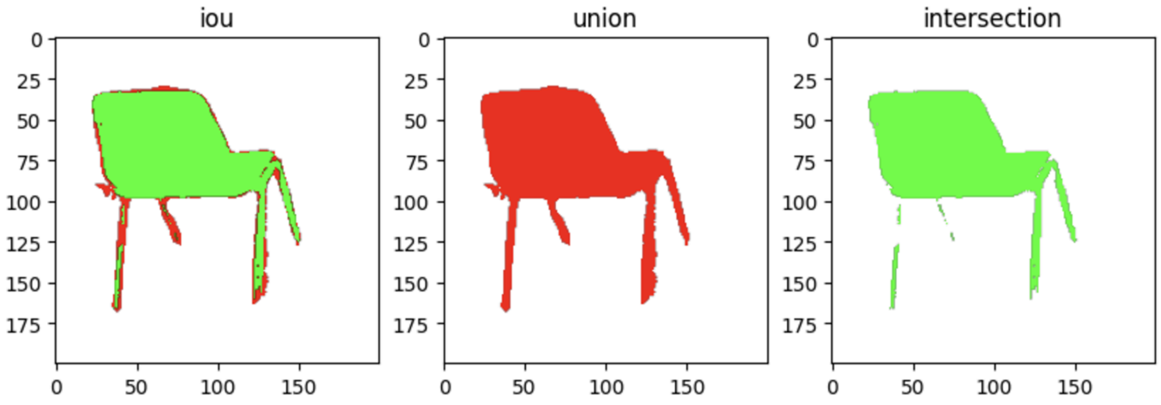
This task involves using pre-trained foundation models, **Segment Anything Model (SAM)** and **CLIP**, to perform open-vocabulary segmentation. The objective was to retrieve masks corresponding to a specific object (e.g., "chair") in images and evaluate the performance using mean IoU against ground truth masks. Experiments included tuning mask generation parameters and testing different point sampling strategies.

Methods

1. **Mask Generation**:
    - **Model**: SAM was used for mask generation.
    - Two approaches were used: to sample mask by anchor points, and to iterate through all masks obtained by MaskGenerator
2. **Point Sampling Strategies**:
    - **Rectangular Grid**: Points are sampled as range with parameter **step**, where step is a step of range. They were located diagonally just to possibly cover more rectangular objects, as our chair.
    - **Uniform Random Sampling**: Points are sampled randomly across the image, with different sample densities.
    - Then, sam_predictor gives 3 masks, I took medium, although other also can be tested
3. **Instance generation:**
    - Masks were generated for all points in the grid and multiplied by image, obtaining the actual instance, segmented by the mask.
4. **Trimming:**
    - Instances obtained from segmented images, gave embeddings far from text prompt embedding, so I trimmed them by bounding values, obtaining bounding boxes
    -

Same object, not trimmed	0.25
Same object, trimmed chair	0.309
Same object, trimmed table	0.285
Different object, not trimmed	0.254
Different object, trimmed	0.247

5. **Threshold:** On test samples, similarity for text and image embeddings were more than 0.3, and for other classes from text embedding less, so I chose to take threshold 0.3. However, this works only with chair class. For other classes thresholds should be chosen specially.
6. **Evaluation**:
   - Mean IoU was computed for all images (`mean`) and specifically for images containing chairs (`mean on images w chair`).



Results

Method	Parameters	Mean IoU	Mean IoU (Only with objects present)
Mask Generator, Threshold 0.28	`threshold=0.28`	0.652	0.388
Mask Generator, Threshold 0.3	`threshold=0.3`	0.693	0.351
Rectangular Grid, Step 30	`step_size=30, threshold=0.3`	0.724	0.423
Rectangular Grid, Step 50	`step_size=50, threshold=0.3`	0.706	0.381
Uniform Grid, 600 Samples, the=0.3	`N=600, threshold=0.3`	**0.728**	**0.433**
Uniform Grid, 400 Samples	`N=400, threshold=0.3`	0.622	0.318

Uniform Grid, 600 samples, 0.27 threshold	`N=600, threshold=0.27`	0.407	0.319
Backpack	`N=500, threshold=0.27`	0.755	0.312

Observations

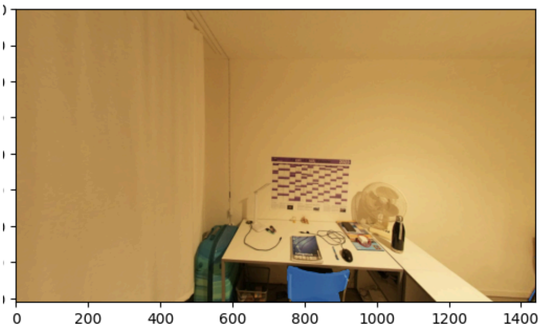
1. **Mask Generator Tuning**:
   - Lower thresholds like `pred_iou_thresh=0.28` improve overall IoU but reduce performance for chairs.
   - Increasing the threshold to `0.3` slightly improves the overall IoU but further decreases performance for chairs.
2. **Grid Sampling**:
   - A rectangular grid with a step size of 30 performed well across both metrics, balancing computational efficiency and coverage.
   - Increasing the step size to 50 reduced performance, likely due to insufficient point density.
3. **Random Sampling**:
   - Uniform random sampling with 600 samples achieved the best results for both metrics, demonstrating that increased sampling density captures more detailed information.
   - Reducing the sample count to 400 caused a significant drop in performance, especially for chairs.
4. **IOU>0.5:**
   With all approaches, I obtained IOU > than 0.5.
5. **Performance on Chair Images**:
   - IoU scores for images with chairs is lower, because in failure cases part of chairs are hidden, or there are two chairs, where the second chair is seen only by its part. So in general, the model managed to recognize the chair, but failed to recognize it by parts.
   - However, changing prompt to “chair part”, or “chair back” does not help to increase the performance, and the model still does not recognise them. This happens because such details are not present in CLIP.
6. **Performance on Backpack class:**
   - I got better results on backpack instances. I think it happens because it is easy to distinguish it from other wooden obstacles. However, It needed to find threshold to achieve improved results.

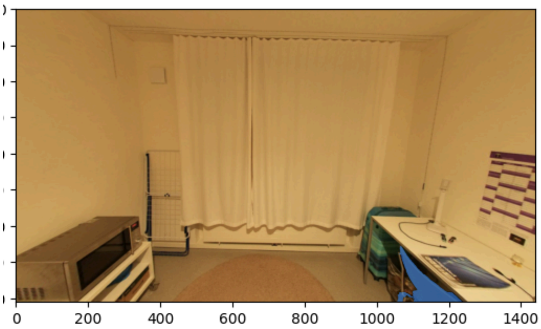
Insights and Proposed Improvements

1. **Grid adjustment:**
    - For grid methods, results depends on whether the grid captured the object. I we could gain objects proposals, we could decrease the number of dots or increase the results.
2. **Clip recognition**
    - Clip does not recognize the object by parts, so it downgrades the metric
3. **Model Parameters**:
    - Results depend high on hyperparameters: threshold, number of grid points, parameters of mask generator.
4. **Further Exploration**:
    - Investigate how mask generator predicts on other init parameters
5. **Other classes**
    - Same object, trimmed chair 0.30908203125
    - Same object, trimmed table 0.285400390625
    - Clip embedding for table is further from text as for chair, so we should find other parameters, however we can modify only threshold, because even if there are enough points, embeddings will be further than 0.3, chosen for chair.
    - Besides, for objects that are bigger, it is profitable to take [2] mask from sam_predictor. However, their are is bigger, so P_uniform is bigger, so they need less points in Uniform grid, to be captured, as probability of capture = Area(Object)/Area(Image)

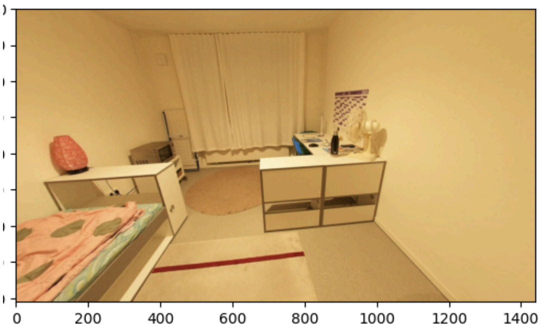
Conclusion

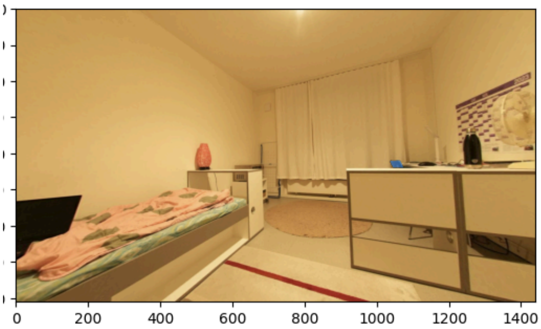
This task demonstrates the powerful zero-shot capabilities of SAM and CLIP for open-vocabulary segmentation. The best results were achieved using **uniform random sampling with 600 points**, achieving a mean IoU of **0.728** and a mean IoU of **0.433** for chairs. Some extra remarks are presented in the Notebook.


0
200
400
600
800
1000
1200
1400


0
200
400
600
800
1000
1200
1400

0
200
400
600
800
1000
1200
1400


0
200
400
600
800
1000
1200
1400


0
200
400
600
800
1000
1200
1400