Scaling Open-Vocabulary Image Segmentation with Image-Level Labels

Golnaz Ghiasi, Xiuye Gu, Yin Cui, Tsung-Yi Lin

Google Research

Open Vocabulary Image Segmentation

- We naturally use **arbitrary text** to refer to objects and stuff in an image.
- The semantic segmentation models are much more **restricted** as they only support a fixed small number of classes (e.g. 100).
- We propose OpenSeg that can organize pixels into meaningful regions indicated by arbitrary text queries.

Training on captions to scale up vocabulary size

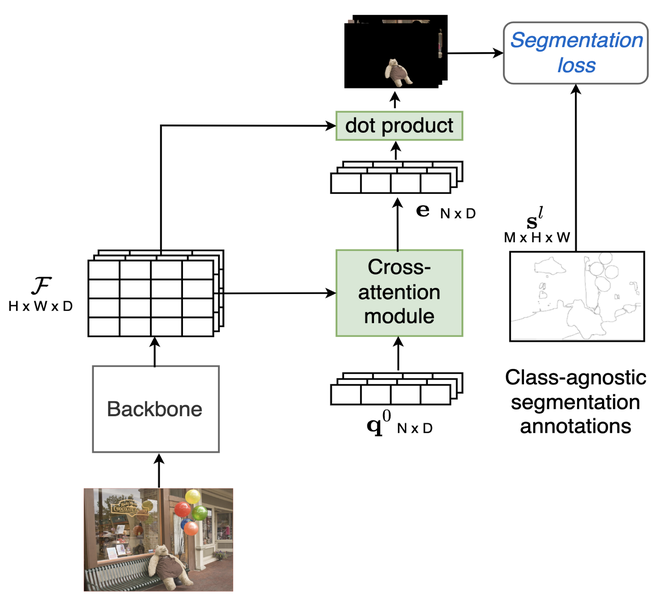
- Captions contain arbitrary text and helps zero-shot transfer.
- Captions are cheap to collect.

To effectively train with caption supervision:

- Supervise on proposal level (**mid-level grouping**) instead of **pixel level**.
- Our model predicts segmentation proposals for **both things and stuff**.
- Use these proposals to represent image.

Mid-level grouping

- We use a cross-attention module to predict proposals (similar to maskformer[1]).
- The input to this module is N learnable queries and image features.
- The output is N embedding vector which are used to compute N proposals.



[1] Per-Pixel Classification is Not All You Need for Semantic Segmentation, NeurIPS 2021

OpenSeg Architecture

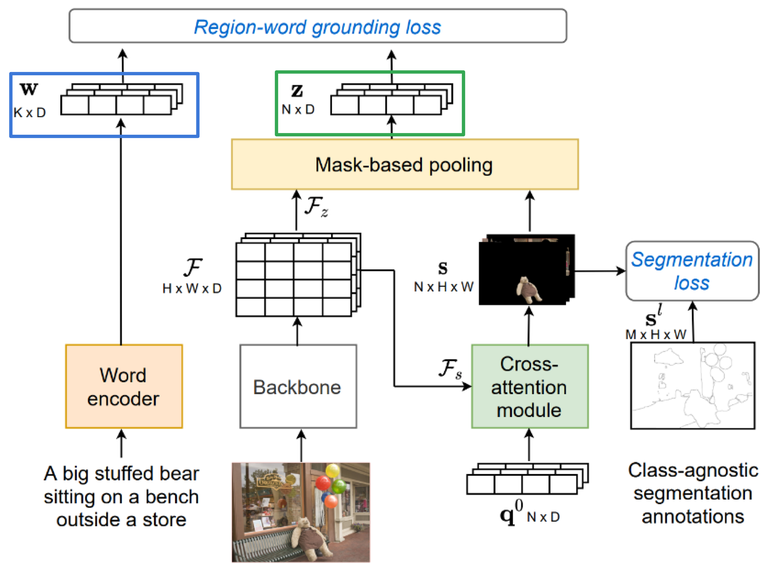


Image representation

- We extracts N proposals for each image and for each proposal computes a feature vector → **Represent an image with N feature vectors (N is number of segments).**

Text representation

- We extract list of nouns and adjectives from caption and compute text embedding for each word → **Represent a text with K feature vectors (K is the maximum number of words).**

Region-word grounding loss

- We use contrastive loss similar to CLIP/ALIGN: Optimize the weights so that similarity score of same pairs of image and caption in a batch become closer and similarity score of different pairs become further.

Similarity score of an image and a caption

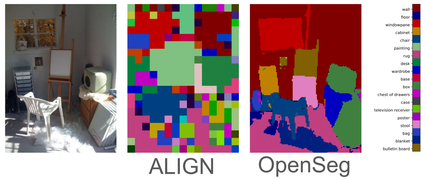
- CLIP/ALIGN computes the similarity of the whole image and the whole caption.
- But we divide the image into regions and caption into the words and compute the similarity score using the alignment score of regions and words.

Experiment setup

- Training data:
  - COCO class agnostic masks + COCO caption.
  - COCO class agnostic masks + COCO caption and Localize Narrative (We annotate Localized Narrative with segmentation proposal pseudo labels).
- Evaluate on hold out semantic segmentation datasets (e.g. ADE-150, ADE-845, Pascal-59, PASCAL-459).

OpenSeg significantly outperforms pre-trained ALIGN

- Pre-trained ALIGN can only roughly localize text queries.
- Adding proposals to pre-trained ALIGN improves mIoU results. But OpenSeg still performs significantly better:



	COCO Train			mIoU				
	label	mask	cap.	A-847	PC-459	A-150	PC-59	COCO
ALIGN	✗	✗	✗	4.8	3.6	9.7	18.5	15.6
ALIGN w/proposal	✗	✓	✗	5.8	4.8	12.9	22.4	17.9
OpenSeg	✗	✓	✓	6.3	9.0	21.1	42.1	36.1

Scaling training data with captions improves performance

- By scaling training data from 118k images (COCO) to 652k images (COCO+Localized Narrative), the performance of OpenSeg improves on average by 2.5 mIoU across 4 benchmarks.

	ADE-847	PASCAL Context-459	ADE-150	PASCAL Context-59
OpenSeg (COCO)	6.3	9.0	21.1	42.1
OpenSeg (COCO + Loc. Narr)	(+0.5) **6.8**	(+2.2) **11.2**	(+3.7) **24.8**	(+3.8) **45.9**

Comparison with existing methods

- OpenSeg (trained on COCO caption) has a better generalization in comparison to LSeg and LSeg+ which are trained with COCO labels.
- Compared with GroupVit, OpenSeg learns visual grouping with class-agnostic segmentation, and has a superior performance

	backbone	external dataset	target dataset	A-847	PC-459	A-150	PC-59	PAS-20
LSeg [29]	ResNet101	✗	✓(seen classes)	-	-	-	-	47.4
LSeg+	ResNet101	COCO	✗	2.5	5.2	13.0	36.0	59.0
OpenSeg(ours)	ResNet101	COCO	✗	4.0	6.5	15.3	36.9	60.0
OpenSeg(ours)	ResNet101	COCO+Loc. Narr.	✗	4.4	7.9	17.5	40.1	63.8
GroupVit [51]	VIT-S	CC12M+YFCC	✗	-	-	-	22.4	52.3
OpenSeg(ours)	eff-b7	COCO+Loc. Narr.	✗	8.1	11.5	26.4	44.8	70.2
+prompt eng.	eff-b7	COCO+Loc. Narr.	✗	**8.8**	**12.2**	**28.6**	**48.2**	**72.2**

https://github.com/tensorflow/tpu/tree/master/models/official/detection/projects/openseg