Revised Section 4&5 of "Open-Vocabulary Object Detection and Part Segmentation"

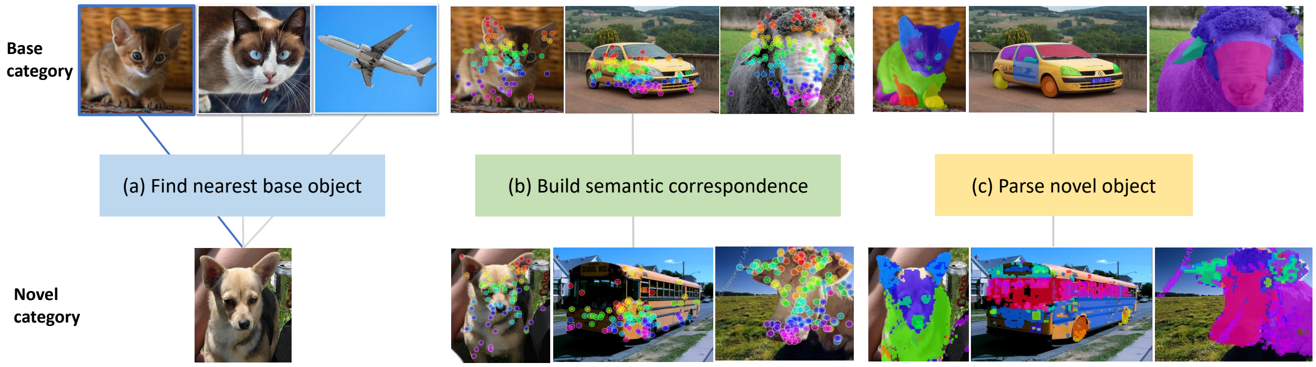
Base category
(a) Find nearest base object
(b) Build semantic correspondence
(c) Parse novel object
Novel category

Figure 4. **The pipeline of parsing novel objects into parts.** (a) Finding the nearest base object for each novel object. (b) Building the dense semantic correspondence between a novel object and its corresponding base object. For better visualization, we only show some points sampled from the feature map grid. (c) The novel object is parsed into its parts as defined in the part segmentation of its corresponding base object.

4. Our Method

We first propose a method to provide part annotations for novel objects, which is based on dense semantic correspondence between the base object and the novel object extracted from DINO [3]. Then we train the detector on part-level, object-level and image-level data, where the detector a vision-language version of Mask R-CNN [11] and its classifier is the text embedding of category name from CLIP [18]. Finally, we obtain an open-vocabulary part segmentation model to support inference on text prompt.

4.1. Parsing Novel Objects into Parts

Most novel objects share the same part taxonomy with one of the base objects, for example, the novel `dog` has the same parts as the base `cat`. Since the part segmentation of the base object is known, we could parse the novel object according to its dense semantic correspondence to the base object. The whole pipeline is shown in Figure 4.

Finding the nearest base object for each novel object. We use DINO [3] to extract the [`class token`] of each base object, denoted as $t^{cls}(\cdot)$, and save these features as the database. Then, for each novel object i, we extract its [`class token`] and find its nearest base object i_{near} in the database by the cosine similarity.

$$i_{near} = \arg\max_{j} \ \mathrm{sim}(t^{cls}(I_i),\ t^{cls}(I_j))$$

Building dense semantic correspondence between the base object and its nearest novel object. We further use the DINO feature map as dense visual descriptors [1], denoted as $F_{x,y}(\cdot)$, where x, y are grid indexes in the feature map. After computing the spatial feature similarity between the novel object $F_{x,y}(I_i)$ and its nearest base object $F_{p,q}(I_{i_{near}})$, for each token (x, y) in the novel object, its corresponding token in the base object are chosen as the token with the highest cosine similarity.

$$x_{corr}, y_{corr} = \arg\max_{p,q} \ \mathrm{sim}(F_{x,y}(I_i),\ F_{p,q}(I_{i_{near}}))$$

Parsing novel parts by semantic correspondence. After dense correspondence between the base object and novel object is obtained, we could parse the novel object into its part segmentation $M_i(x, y)$ as defined in its corresponding base object part segmentation $M_{i_{near}}(p, q)$.

$$M_i(x, y) = M_{i_{near}}(x_{corr}, y_{corr})$$

For example, an image of novel `dog` first finds its nearest image, an image of base `cat`, then builds the dense semantic correspondence with this `cat`. The segmentation map of `cat: head` is known, it can be transferred to the segmentation map of `dog: head`. Similarly, the segmentation map transfer is performed on other parts `torso, leg, tail`. In this way, the novel object `dog` is parsed into its parts `dog: head, torso, leg, tail`.

4.2. Detector Architecture

Image encoder. The image encoder is based on convolutional neural networks such as ResNet [12] or Transformer-based models like Swin [17], followed by Feature Pyramid Network [15] to generate multi-scale feature maps to be used in the detection decoder.

Detection decoder. The architecture of detection decoder is composed of a region proposal network (RPN) [20] and a R-CNN recognition head. RPN provides box proposals for both objects and parts. R-CNN recognition head refines the box location and the classification score. Notably, the classifier weight in the recognition head is replaced by text embedding of the class name of the object and the part.

Text embedding as the classifier. The classification score of the recognition head is implemented as a dot-product operation between the region features and the text embeddings, where the region features are cropped from feature maps of the image encoder, and the text embeddings are extracted from the text encoder in CLIP [18].

Mask decoder. We choose the architecture of mask decoder from Mask R-CNN [11] and replace the original multi-classification head with a class-agnostic head to support segmentation on novel categories. We note that more advanced architecture such as Mask2Former [5] has the potential to further improve the performance but is not the focus of this work.

4.3. Training on Parts, Objects, and Images

The training data includes part-level, object-level, and image-level data. The image data is further parsed into the part annotation. Our detector is joint-trained on these data to establish multi-granularity alignment.

Part segmentation data. Part segmentation data [4, 10, 19] contains part mask segmentation and its category. Part is always defined as an object-part pair since the same semantic part can be very different when it is associated with different objects. The category name of the part is formalized as follows:

$$C_{part} = [\text{“dog: head”}, \text{“dog: nose”}, ..., \text{“cat: tail”}]$$

Object detection data. Object detection data contains object boxes and its category. Most object detection datasets [9, 16] also provide object mask segmentation annotations.

$$C_{object} = [\text{“person”}, \text{“bicycle”}, ..., \text{“toothbrush”}]$$

The training loss for part and object data includes all location loss, classification loss, and mask loss.

Image classification data. Image classification data provides a large vocabulary of object categories in the form of images. Although object-level or part-level bounding annotations are absent, these images could be effectively used by the following ways: (1) The classification loss can be performed on max-size proposal [22] for each image, and therefore expands the object-level vocabulary. This is similar to previous open-vocabulary object detection works [13, 14, 21, 22]. (2) As introduced in section 4.1, the image can be parsed into parts and used as part-level annotations to expand the vocabulary of part categories. The training loss about image data only includes classification loss.

4.4. Inference on Text Prompt

In inference, the model takes as input the image and outputs the part segmentation for the object. Since all vocabulary of both objects and parts are a large number, and the user may not be interested in obtaining all possible object and part segmentation, our detector supports inference on text prompt by user input.

The left section of Figure 1 is a case using a dog as an example. When the user-input is [`dog`], [`dog: head, torso, leg, tail`] and [`dog: head, ear, eye, nose, torso, leg, paw, tail`], the detector outputs the segmentation results in different granularities accordingly. The right section of Figure 1 is a range of objects in the open world. It can be seen that our model is able to detect both open-vocabulary objects and their parts. When our detector is used in real applications, one can flexibly choose to use the pre-defined part taxonomy in datasets such as Pascal Part, PACO, or custom text prompt.

5. Experiment

5.1. Datasets

For part segmentation data, we use three sources of datasets, Pascal Part [4], PartImageNet [10] and PACO [19].

Pascal Part. The original Pascal Part provides part annotations of 20 Pascal VOC classes, a total of 193 part categories. Its taxonomy contains many positional descriptors, which is not suitable for this paper, and we modify its part taxonomy into 93 part categories.

PartImageNet. PartImageNet groups 158 classes from ImageNet into 11 super-categories and provides their part annotations, a total of 40 part categories.

PACO. PACO supplements more electronic equipment, appliances, accessories, and furniture than Pascal Part and PartImageNet. PACO contains 75 object categories, 456 object-part categories and 55 attributes. The image sources of PACO are LVIS and Ego4D [8]. In this work, we use PACO-LVIS set as default. We focus on object parts and leave attributes for future research.

For image-level data, we use ImageNet1k (**IN**) [6]. We also create ImageNet-super11 (**IN-S11**) and ImageNet-super20 (**IN-S20**) that overlap with PartImageNet and Pascal category vocabulary separately. As introduced in Section 4.3, these images can be parsed into parts and used as part-level data, including **Parsed IN**, **Parsed IN-S11** and **Parsed IN-S20**. They share the same set of images with image-level data but have different levels of categories. For example, the categories of IN-S20 are [`quadruped`, `biped`, etc.], and the categories of Parsed IN-S20 are [`quadruped: head`, `quadruped: body`, etc.].

Data	# Img	All	quadruped			
		(40)	head	body	foot	tail
Pascal Part	4k	4.5	17.4	0.1	0.0	2.8
+ IN-S11	20k	5.3	21.1	2.6	0.7	0.8
+ Parsed IN-S11	36k	7.2	37.0	9.5	3.1	4.9
vs. Pascal Part		+2.7	+19.6	+9.4	+3.1	+2.1
PartImageNet	16k	29.7	57.3	25.8	22.9	22.9

(a) **Cross-dataset generalization when only one part dataset, Pascal Part, is available.** Pascal Part is trained on the Pascal Part training set. Starting from Pascal Par, IN-S11 and Parsed IN-S11 are added into the training sequentially: + IN-S11 is joint-training on Pascal Part and IN-S11 data; + Parsed IN-S11 is joint-training on Pascal Part, IN-S11 and Parsed IN-S11 data.

Data	# Img	All	quadruped			
		(40)	head	body	foot	tail
Pascal Part	4k	4.5	17.4	0.1	0.0	2.8
+ PACO	49k	5.4	18.7	3.5	0.3	3.7
+ LVIS	149k	7.5	28.7	7.2	0.2	2.6
+ Parsed IN-S11	165k	9.8	50.7	11.5	3.8	6.5
vs. Pascal Part		+5.3	+33.3	+11.4	+3.8	+3.7
PartImageNet	16k	29.7	57.3	25.8	22.9	22.9

(b) **Cross-dataset generalization when more than one part datasets are available.** Starting from Pascal Part, LVIS, PACO and Parsed IN-S11 are added into the training sequentially: + PACO is joint-training on Pascal Part and PACO data; + LVIS is joint-training on Pascal Part, PACO and LVIS data; + Parsed IN-S11 is joint-training on Pascal Part, PACO, LVIS and Parsed IN-S11 data.

Table 3. **Cross-dataset generalization on PartImageNet part segmentation.** The evaluation metric is mAP_{mask}@[.5, .95] on the validation set of PartImageNet. All models are ResNet50 Mask R-CNN and use the text embedding of the category name as the classifier. PartImageNet is the fully-supervised method as the oracle performance.

For object-level detection data, we use VOC [7], COCO [16] and LVIS [9]. Their details are in Appendix.

5.2. Cross-dataset segmentation on PartImageNet

In Table 3, we study cross-dataset generalization by using PartImageNet validation set as the evaluation dataset, where the metrics of all (40) parts and the detailed metrics of parts of `quadruped` are reported.

Table 3a shows when Pascal Part is the only available human-annotated part dataset, using IN-S11 and parsed IN-S11 data could help to improve PartImageNet performance.

Baseline from Pascal Part. The baseline method directly uses the Pascal Part-trained model to evaluate PartImageNet. As shown in Table 3a first row, the performance is poor, for example, `body` and `foot` of the `quadruped` are nearly to zero. Pascal Part has no semantic label of `quadruped`, and the model needs to generalize from parts of `dog`, `cat`, etc. in Pascal Part to parts of `quadruped` in PartImageNet. The possible generalization ability comes from the text embedding generated from CLIP [18]. However, generalization in part-level recognition is beyond its capability since CLIP is pre-trained on only image-level data.

IN-S11. Considering that Pascal Part has no semantic label such as `quadruped`, `piped`, *etc.*, we collect IN-S11 images from ImageNet and add them to the training as image-level classification data. As shown in Table 3a second row, the performance is improved to some extent. This shows that image-level alignment is beneficial to the part recognition task. However, since no additional part-level supervision signal is introduced when using IN-S11 as image classification data, the improvement is still limited.

Parsed IN-S11. We use our parsing pipeline to deal with IN-S11 images and generate their part annotations. As shown in the third row in Table 3a, introducing these parsed parts into the training brings a significant improvement, 2.1∼19.6 mAP improvement on the parts of `quadruped` and 2.7 mAP gain on all 40 parts over the baseline method. This suggests that our proposed methods are able to provide an effective part-level supervision signal to the detection model and boosts its performance on cross-dataset generalization.

More part datasets are available. Table 3b shows when more than one human-annotated part datasets are available, including Pascal Part, PACO, and LVIS. Although LVIS is an object-level dataset, we find its categories contain many object parts, such as shoes, which can also be seen as parts. From the first three rows of Table 3b, we can see that when the part-level annotations grow in training, the part segmentation obtains better performance, from 4.5 mAP to 7.5 mAP. When Parsed IN-S11 are added to the training, the performance is further boosted by a large margin. For example, the head of `quadruped` has achieved 50.7 mAP, close to fully-supervised 57.3 mAP. This shows that when more data sources are available in the future, a strong model for part segmentation in the open world is promising.

5.3. Cross-category segmentation on Pascal Part

We evaluate the cross-category generalization within the Pascal Part dataset. All 93 parts are split into 77 base parts and 16 novel parts, detailed in Appendix. Table 4 reports the metrics of all (93), base (77), and novel (16) parts.

Baseline from Pascal Part base. Table 4 first row is the baseline, which is trained on base parts and evaluated on novel parts. Since the detector uses CLIP text embedding as the classifier, the novel parts obtain non-zero segmentation performance.

VOC object. Compared with the part annotation, the object annotation is much easier to collect. We add VOC object data to verify whether this could help to improve the per-

Data	# Img	All (93)		Base (57)		Novel (36)	
		AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
Pascal Part base	4k	11.9	26.9	18.6	41.0	1.2	3.0
+ VOC object	6k	13.4	30.0	20.0	44.2	2.8	7.5
+ IN-S20	56k	13.6	30.2	20.5	44.8	2.5	6.9
+ Parsed IN-S20	105k	13.7	30.4	20.4	44.6	3.0	7.9
vs. Pascal Part base		**+1.8**	**+3.5**	**+1.8**	**+3.6**	**+1.8**	**+4.9**
Pascal Part	4k	19.4	42.7	18.8	41.5	22.1	48.9

Table 4. **Cross-category generalization on Pascal Part part segmentation**. The evaluation metric is mAP_{mask}@[.5, .95] and AP_{mask}@.5 on the validation set of Pascal Part. All models are ResNet50 Mask R-CNN and use the text embedding of the category name as the classifier. Pascal Part base is the base (57) parts split from all (93) parts. Starting from Pascal Par base, VOC object, IN-S20 and Parsed IN-S20 are added into the training sequentially: + VOC object is joint-training on Pascal Part base and VOC object detection data; + IN-S20 is joint-training on Pascal Part base, VOC object detection and IN-S20 data; + Parsed IN-S20 is joint-training on Pascal Part base, VOC object detection, IN-S20 and Parsed IN-S20 data. Pascal Part is the fully-supervised method as the oracle performance.

Method	PartImageNet		Pascal Part		PACO	
	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
Joint	29.1	52.0	22.6	47.8	9.3	18.9
+ IN	30.8	54.4	23.6	49.2	9.0	18.7
+ Parsed IN	31.6	55.7	24.0	49.8	9.6	20.2
vs. baseline	**+2.5**	**+3.7**	**+1.4**	**+2.0**	**+0.3**	**+1.3**
Oracle	29.7	54.1	19.4	42.3	10.6	21.7

(a) All models are ResNet50 [12] Mask R-CNN [11].

Method	PartImageNet		Pascal Part		PACO	
	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
Joint	40.0	64.8	31.2	60.5	15.4	30.3
+ IN	41.2	66.8	31.7	61.1	15.9	30.8
+ Parsed IN	42.0	68.2	31.9	61.6	15.6	30.6
vs. baseline	**+2.0**	**+3.4**	**+0.7**	**+0.9**	**+0.2**	**+0.3**
Oracle	41.7	68.7	27.4	56.1	15.2	29.4

(b) All models are Swin-B [17] Cascade Mask R-CNN [2].

Table 5. **Part segmentation across datasets.** All models are evaluated by setting the classifier as text embedding of category name in the evaluation dataset. Joint denotes the joint-training on LVIS, PartImageNet, Pascal Part and PACO datasets. Dataset-specific uses the training data of each dataset, separately.

formance. As shown in the second row of Table 4, adding VOC object data helps to improve the performance on both base parts and novel parts in Pascal Part. This demonstrates that object-level alignment could lead to better part-level performance.

IN-S20. Image-level classification data is also an easy-to-get annotation. We collect images with Pascal categories from ImageNet, IN-S20, and add them to the training. As shown in Table 4 third row, additional image-level data does not bring much gain than object detection data. This is because image-level data has a similar effect as object-level data on part-level recognition. Most of its gain is diminished by object data.

Parsed IN-S20. We use our proposed parsing method to generate part annotations for novel objects, and they provide supervision on part classification. As shown in Table 4 fourth row, our method improves the performance on both base and novel categories. This shows that our parsing pipeline is an effective solution to both base and novel object part segmentation.

5.4. Part segmentation across datasets

Towards detecting and parsing *any* object in the open world, we train a detector on the joint of available part segmentation datasets, including LVIS, PACO, Pascal Part and PartImageNet. The performance is shown in Table 5.

This joint training model shows good generalization ability on various evaluation datasets, for example, Pascal Part obtains 22.6 mAP, better performance than its dataset-specific training. However, the potential problem lies in that joint training does not benefit all datasets, where PartImageNet and PACO decrease the performance a little.

To make up for the performance loss, we add IN and Parsed IN into the training. It can be seen all datasets obtain the performance gain accordingly. When we scale up the model capability from ResNet50 [12] to Swin-B [17], the detector achieves better performance than dataset-specific training on all Pascal Part, PartImageNet and PACO datasets.

5.5. Ablation Study

Text prompt template. Since the part is associated with the object category, we study how to design the text prompt template of (`object`, `part`) pair to the text encoder. We select two common expressions: a [`object`] [`part`] and [`part`] of a [`object`]. For example, [`dog`] and [`head`], these two expressions are [`a dog head`] and [`head of a dog`]. As shown in Table 6, a [`object`] [`part`] behaves a little better than [`part`] of a [`object`] in Pascal Part while not in PartImageNet. Which expression is a generally better usage of text prompt to the part needs to be verified on more datasets and we leave it for future research. In addition, more advanced prompt engineering for part segmentation is also an open problem.

Aligning method for novel parts. We compare different aligning methods to use IN-S11 data to help part segmentation in PartImageNet. We select two popular designs in

Pascal Part	All	dog			
	(93)	head	torso	paw	tail
a [object] [part]	19.1	50.7	18.7	20.7	10.4
[part] of a [object]	18.4	48.8	17.6	21.3	9.2
PartImageNet	All	quadruped			
	(40)	head	body	foot	tail
a [object] [part]	29.7	57.3	25.8	22.9	22.9
[part] of a [object]	29.9	55.9	25.1	22.9	24.3

Table 6. **Text prompt template to object part.** We compare different templates of text prompt to object part in the fully-supervision setting of Pascal Part and PartImageNet. The evaluation metric is mAP_{mask}@[.5, .95] on the validation set.

Method	All	quadruped			
	(40)	head	body	foot	tail
Baseline	5.3	21.1	2.6	0.7	0.8
Max-score [21]	5.8	29.0	7.6	1.6	2.3
Max-size [22]	4.4	17.3	0.1	0.0	2.8
Parsed (ours)	7.2	37.0	9.5	3.1	4.9

Table 7. **Comparisons of different aligning methods for novel parts.** The experiments are carried out on cross-dataset generalization from Pascal Part to PartImageNet. The evaluation metric is mAP_{mask}@[.5, .95] on the validation set of PartImageNet. Fine-tuning from the baseline model, max-score, max-size and our method apply different designs to utilize image-level data to further improve part segmentation performance, where the former two are trained on part labels expanded from the image label.

open-vocabulary object detection, max-score and max-size. Max-score is selecting the proposal that has the highest score of the target category as the matched proposal, used in [21]. Max-size is selecting the proposal that has the maximum area among all proposals as the matched proposal to the target category, proposed in [22]. For each ImageNet image, its object category is known, and its part taxonomy can be inferred, these parts will be used as the target category in max-score and max-size methods.

- *Max-score.* As shown in Table 7 second row, max-score helps to improve the performance a little over baseline. Fine-tuning from the baseline model, its selected highest score proposals contain efficient training samples, and these samples bring performance gain.

- *Max-size.* As shown in Table 7 third row, the max-size method degenerates the performance in most metrics. According to the max-size rule, all parts are assigned to the same proposal, it is hard to align the part-level region to its part name text embedding. This shows that part-level alignment is more difficult than object-level and an efficient fine-grained supervision signal is necessary.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021. 1

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018. 4

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1

[4] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 2

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 3

[8] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2

[9] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 2, 3

[10] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. *arXiv preprint arXiv:2112.00933*, 2021. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4

[13] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10965–10975, 2022. 2

[14] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. *arXiv preprint arXiv:2211.14843*, 2022. 2

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2, 3

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 4

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3

[19] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. PACO: Parts and attributes of common objects. In *arXiv preprint arXiv:2301.01795*, 2023. 2

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1

[21] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. *arXiv preprint arXiv:2112.09106*, 2021. 2, 5

[22] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022. 2, 5