VizWiz-FewShot: Locating Objects in Images Taken by People With Visual Impairments - Supplementary Materials

Yu-Yun Tseng*, Alexander Bell*, and Danna Gurari
* denotes equal contribution

University of Colorado Boulder

This document supports Sections 3 and 4 of the main paper. In particular, it includes the following:

- List of categories in VizWiz-FewShot (supplements *Section 3.1*)
- Annotation interfaces (supplements *Section 3.1*)
- Quality control mechanisms for crowdsourcing instance segmentations (supplements *Section 3.1*)
- Comparison of unique categories with backward compatible categories (supplements *Section 3.2*)
- Comparison of instance sizes in VizWiz-FewShot-IS-25^i with COCO-20^i (supplements *Section 3.2*)
- Fine-grained analysis of holes in instance segmentations (supplements *Section 3.2*)
- Analysis of the prevalence of instances and categories (supplements *Section 3.2*)
- Examples of our benchmarked algorithm's few-shot object detections on VizWiz-FewShot-OD-25^i (supplements *Section 4.1*)

1 List of Categories in VizWiz-FewShot

The 100 categories in VizWiz-FewShot is divided into 4 folds in VizWiz-FewShot-OD-25^i and VizWiz-FewShot-IS-25^i. Our dataset includes object categories that overlaps with those in COCO-20^i [5], PASCAL-5^i [1], FSOD [2], and FSS-1000 [3] as well as categories that are unique to our dataset. In Table 1, we list all the categories in the 4 folds and indicate which categories are backward compatible with other few-shot datasets.

2 Annotation Interfaces

We utilized two interfaces for collecting annotations from Amazon Mechanical Turk crowdworkers.

The first interface is for image classification, and a screenshot of it is shown in Figure 1. It shows instructions on the left side indicating to select all categories present in the image or *None of the above*. The image is displayed in the center

Table 1: List of the 100 categories in VizWiz-FewShot and the 4-fold class splits in VizWiz-FewShot-OD-25^i and VizWiz-FewShot-IS-25^i. The category that are backward compatible with other few-shot datasets are indicated as below. • refers to the overlapping categories with COCO-20^i [5], ▲ refers to the overlapping categories with PASCAL-5^i [1], ■ refers to the overlapping categories with FSOD [2], and ⧫ refers to the overlapping categories with FSS-1000 [3].

$i = 0$	$i = 1$	$i = 2$	$i = 3$
couch •	curtain ■	sign	laptop ⧫ • ■
watch ■	spoon • ■ ⧫	truck • ■	cell phone
drawer ■	food menu	vacuum ⧫	bed •
landline phone	bowl • ■	clock •	vase • ■ ⧫
strawberry ■ ⧫	receipt	ipad ⧫	dial
painting	newspaper	perfume ■ ⧫	broccoli • ■ ⧫
pillow ■ ⧫	album	towel ■	scale ■
crockpot	cereal box	stool ■ ⧫	piano
pen ■ ⧫	lamp ■	microwave ⧫ • ■	oven • ■
speaker ⧫	dog • ▲	magazine	house
ring	sandwich • ⧫	gift card	bottle • ▲ ■
sock ■ ⧫	ceiling fan ⧫	stapler ■ ⧫	hat
car • ▲	toaster • ■ ⧫	dog collar	bird • ▲
fork • ■ ⧫	cake • ■	sandal ■ ⧫	knife • ■ ⧫
coin ■ ⧫	book • ■	shoe	wallet ■ ⧫
envelope ■ ⧫	apple • ■ ⧫	banana • ■ ⧫	suitcase • ■ ⧫
bracelet ⧫	packet ■	carrot • ■ ⧫	backpack • ■ ⧫
television ■ ⧫	rug	microphone ⧫	sweatshirt ⧫
cash	printer ■ ⧫	plate ■ ⧫	stove ⧫
monitor ▲ ⧫	electric fan ⧫	sticker	computer mouse • ■ ⧫
sink • ■	remote •	person • ▲	wine ■
guitar ■ ⧫	orange • ■ ⧫	chair • ▲	laundry machine
purse	ramen	refrigerator • ■ ⧫	keyboard • ⧫
calculator ■ ⧫	cup • ⧫	bar	key ⧫
cat • ▲ ■	flashdrive	pizza • ■ ⧫	tube

of the interface and approximately 25 categories to select from are displayed on the right side.

The second interface is for instance segmentation, and a screenshot of it is shown in Figure 2. A step-by-step list of instructions is shown on the left of the interface and is listed in Figure 3. It is supplemented with *More Instructions*, which we partially show in Figure 4. The additional guidance includes links to videos explaining how to perform various annotation operations such as drawing a polygon, undoing an action, modifying an existing instance, and erasing an existing instance or polygon partially or entirely. The extra instructions also indicate how to handle edge-case scenarios, alongside examples of what to do and

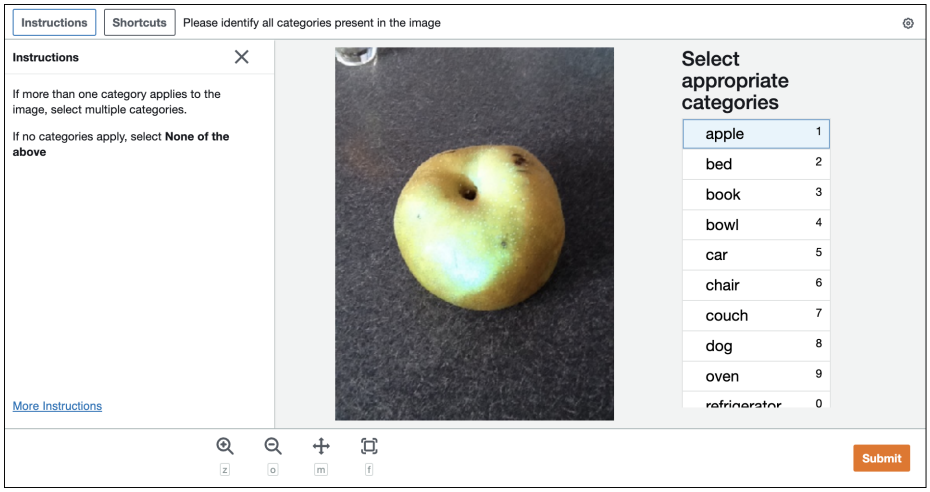


Fig. 1: Our image classification interface. We provide instructions, an image, and a multi-select list that shows ~ 25 categories on the right.

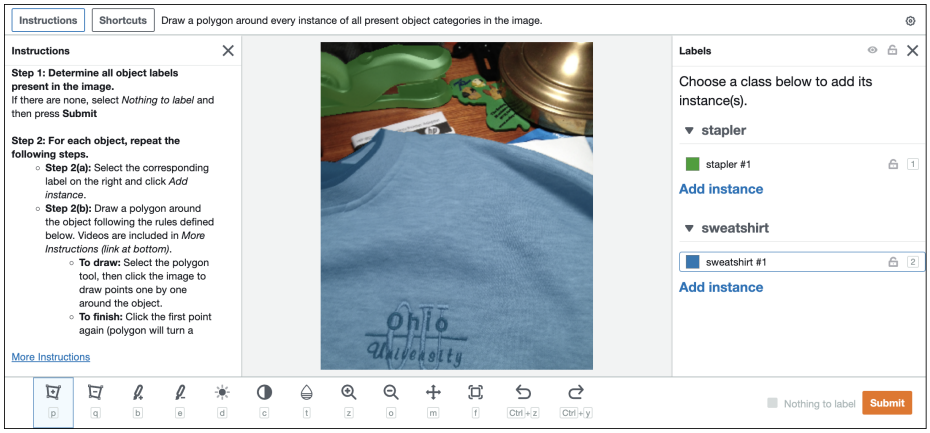


Fig. 2: Our instance segmentation interface. We provide instructions, an image, and a few categories for which annotators should locate all instances.

what not to do. Covered edge-case scenarios are how to handle complex boundaries, high image coverage, holes, and occlusions. We also specify that objects printed on boxes (such as pizza on a frozen pizza box) should be annotated.

3 Quality Control Mechanisms for Crowdsourcing

In the main paper, we noted that crowdworkers had to pass a qualification test in order to work on our instance segmentation tasks. The nine tasks that we

Step 1: Determine all object labels present in the image. If there are none, select Nothing to label and then press Submit
Step 2: For each object, repeat the following steps.
Step 2(a): Select the corresponding label on the right and click Add instance.
Step 2(b): Draw a polygon around the object following the rules defined below. Videos are included in More Instructions (link at bottom).
- **To draw:** Select the polygon tool, then click the image to draw points one by one around the object.
- **To finish:** Click the first point again (polygon will turn a color when it is fully connected), and then press the return key. If you need more segments for the same object, you can draw them from here and continue to press return after each one.
- **To undo:** Click the undo button or use the keyboard shortcut Ctrl+Z.
- **To erase:** Select the polygon eraser tool, then draw the area you wish to erase and press enter.
- **To fix a polygon:** Follow the draw/erase steps above for any existing polygon to make changes to it. This can be done after the polygon is finalized by pressing the return key.

Step 2(c): YOU MUST press the return key to finalize the polygon.
Step 2(d): Verify that the polygon is stored for the instance.
Step 3: Press Submit.

Fig. 3: Instructions provided for our instance segmentation task.

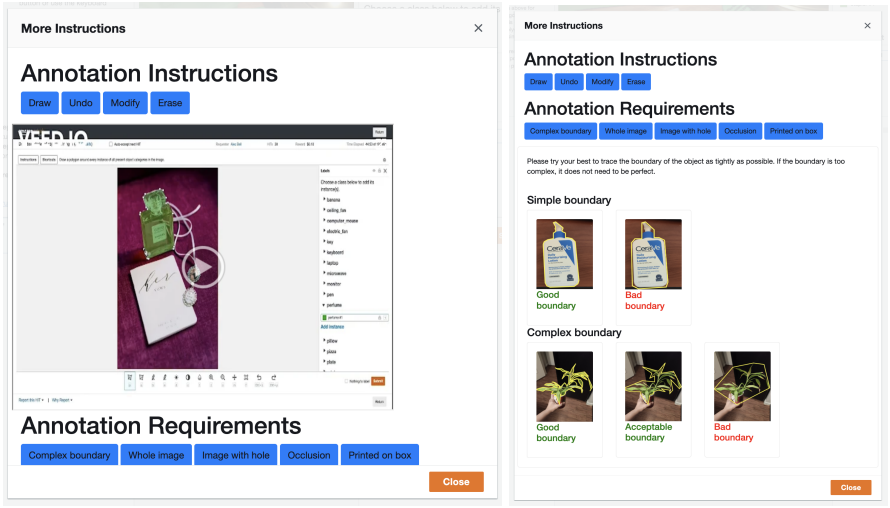
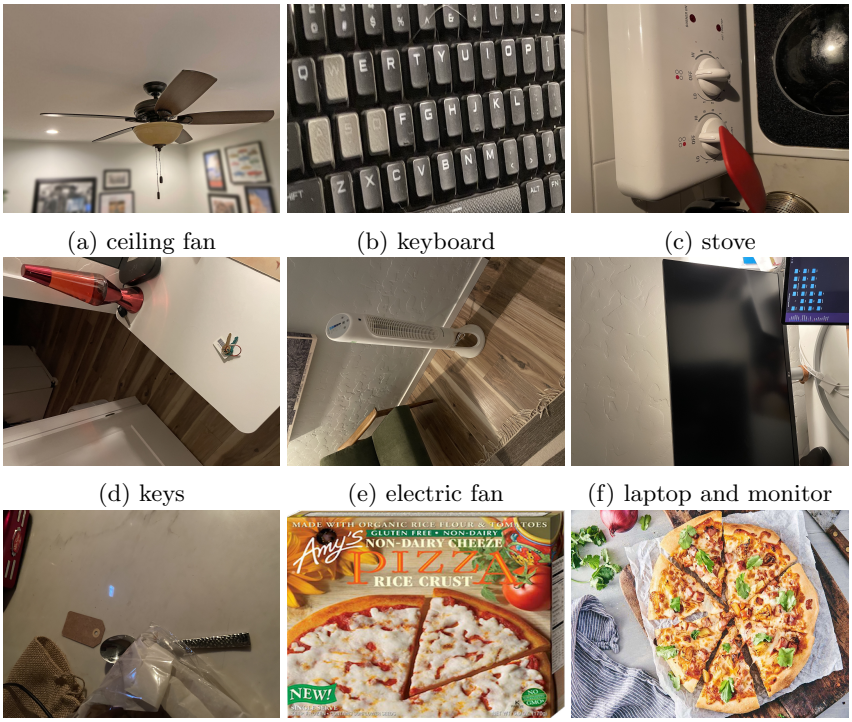


Fig. 4: Additional instructions for our instance segmentation task showing short video tutorials for how to use the annotation tools (on the left) and indicating how to handle edge-case scenarios (on the right).



(a) ceiling fan (b) keyboard (c) stove

(d) keys (e) electric fan (f) laptop and monitor

(g) spoon (h) pizza (i) pizza

Fig. 5: The nine images that had to be correctly annotated by crowdworkers in their qualification test for our instance segmentation task.

presented in this qualification test are shown in Figure 5. These photos were taken by the authors or found online. The tasks test workers ability to annotate complex boundaries (images *a,d*), the whole image (image *b*), occlusions (images *c,g*), holes in instances (image *e*), and objects found both its original physical form and in print (images *h,i*).

4 Comparison of Unique Categories in VizWiz-FewShot-IS-25 with COCO-20^i Categories

We conduct fine-grained analysis of our dataset to elucidate whether categories that are unique to our dataset manifest different characteristics than categories that are in common with COCO-20^i [5]. To do so, we analyze the boundary complexity, image coverage, and presence of text separately for the unique categories in our dataset and the 37 categories overlapping with those of COCO-20^i. Across all instance segmentations in each subset, we compute the mean and standard deviation of the isoperimetric inequality and the image coverage as well as the percentage of instance segmentations containing text.

	Shared	Unique
Boundary Complexity	0.51 ± 0.21	0.54 ± 0.21
Image Coverage	0.24 ± 0.27	0.23 ± 0.27
Presence of Text	17.69%	23.69%

Table 2: Comparison of instance segmentations in our dataset between those that show unique categories versus backward compatible categories in COCO-20^{i}.

Dataset	Instance Sizes		
	small	medium	large
Ours	2.11%	10.07%	87.82%
COCO-20^{i}	41.78%	34.92%	23.31%

Table 3: Proportion of instances belonging to each sizing category as defined by MS-COCO. We notice that the vast majority of instances in our dataset are large, while instance sizes are more evenly distributed in COCO.

Results are shown in Table 2. The key difference between the two subsets is that the presence of text on an instance is more prevalent for categories unique to our dataset. We also observe for categories unique to our dataset that, on average, the objects' boundaries are slightly less complex (i.e., larger scores) and occupy slightly more of the images.

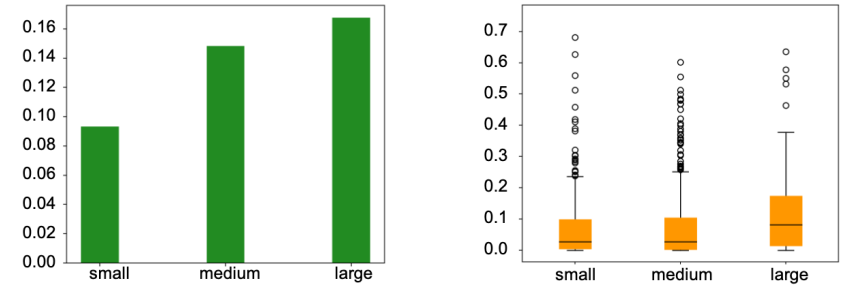
5 Comparison of Instance Sizes in VizWiz-FewShot-IS-25^{i} with COCO-20^{i}

We analyze what proportion of instance segmentations in our dataset fall into small, medium, and large sizes based on the thresholds proposed in MS COCO [4]: 32^2 and 96^2. Results are shown in Table 3. While these thresholds are able to roughly divide COCO-20^{i} into three even subsets, our dataset appears to have an extreme distribution where most of the instances fall into the large category.

6 Fine-grained Analysis of Holes in Instance Segmentations

Limitation of prior work in lacking hole annotations: As noted in the main paper, COCO-20^i, instance segmentations lack holes whereas our dataset includes hole annotations. We note that the absence of holes leads to a limitation in prior work's approach for locating instance segmentations. Specifically, content from occlusions is excluded from instance segmentations when the occlusions overlap their outer boundaries but included when the occlusions are fully enclosed in the

Fig. 6: Examples from COCO-20^i dataset shows how occluding objects are sometimes removed from instance segmentations (e.g., flower in first example) but sometimes not (e.g., oranges in the bowl in the first example).



(a) Hole prevalence for objects of different sizes.

(b) Distribution of hole coverage for objects of different sizes.

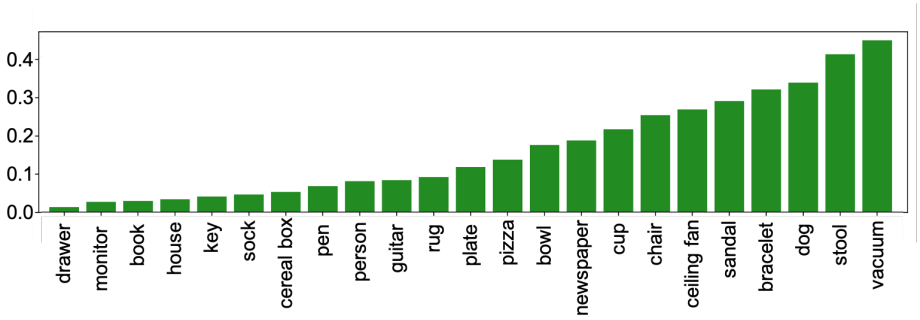
Fig. 7: Analysis on hole prevalence and hole coverage distribution in VizWiz-FewShot-IS^i by category. Only the instances with holes are included in the hole coverage distribution.

instance segmentations. In other words, there is an inconsistency in whether the instance segmentations include pixels that do not belong to the objects. This is exemplified in Figure 6, where the food contained in the bowl is included in the instance segmentation while the flower occluding the bowl is excluded from the instance segmentation. Our dataset, in contrast, leads to a consistent definition of instance segmentations by annotating holes and so always excluding any pixels that do not belong to the target category.

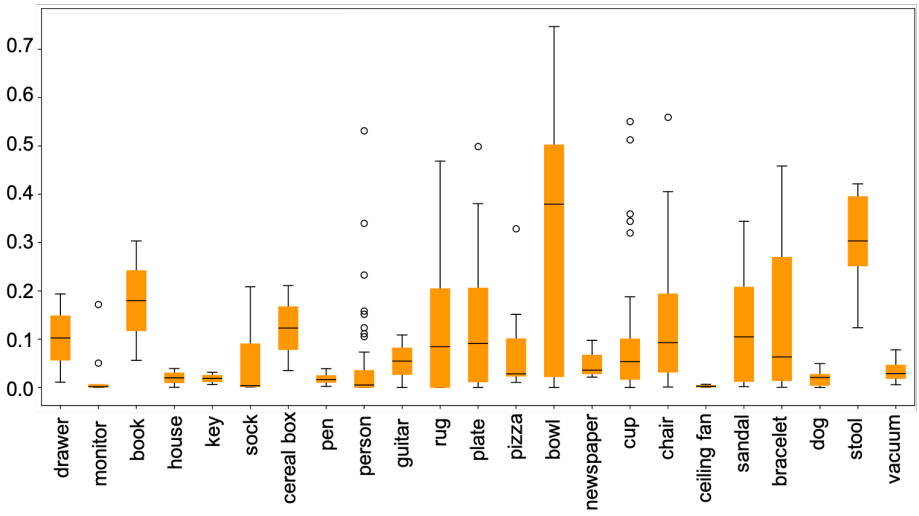
Hole prevalence: We conducted fine-grained analysis on the proportion of instance segmentations with holes based on object size. To do so, we divided all instance segmentations into small, medium, and large buckets using as thresholds 200^2 and 550^2. The holes with areas smaller than 10 pixels are excluded from consideration since we found that such annotations typically reflected holes that were accidentally/mistakenly created by the annotators. Some of these errors are due to the conversion of annotations from vertex-base to pixel-base. Results are shown in Figure 7a. The trend shows that the proportion of instances with holes

grows as object size grows. We also report the proportion of instance segmentations with holes based on object category for a random sample of 23 categories. The results are shown in Figure 8a, sorted by the proportion of instances with holes in ascending order. Altogether, these results highlight that the prevalence of holes differs on per-size and per-category bases.

Typical hole coverage: We also analyzed the percentage of pixels all holes occupy in each instance segmentation from all pixels contained in each instance segmen-



(a) Hole prevalence by category.



(b) The distribution of hole coverage by category.

Fig. 8: Analysis of hole prevalence and hole coverage distribution in VizWiz-FewShot-IS^{i} by category. Only the instances with holes are included in the hole coverage distribution. The categories are sorted by hole prevalence in ascending order.

tation (when including the hole pixels). The results of hole coverage divided by size and category are presented in Figure 7b and Figure 8b respectively. We observe that the proportion of hole coverage tends to be larger for larger objects.

Comparing the hole prevalence and hole coverage results in Figure 8a and Figure 8b, we find that the proportion of instance segmentations with holes and hole coverage do not have a strong correlation in per-category bases. For example, a high percentage of stools and vacuums stools have holes (41.4% and 45.0%, respectively), but based on the natural structure of these categories, they show distinct distributions in hole coverage. On the other hand, the categories that tend to be occluded by other objects, such as bowls, might not as frequently include holes. However, this does not influence the high hole coverage of bowls, where foods are often found to cover a large area in our dataset. Fig. 9 shows examples of stools, vacuums, and bowls.

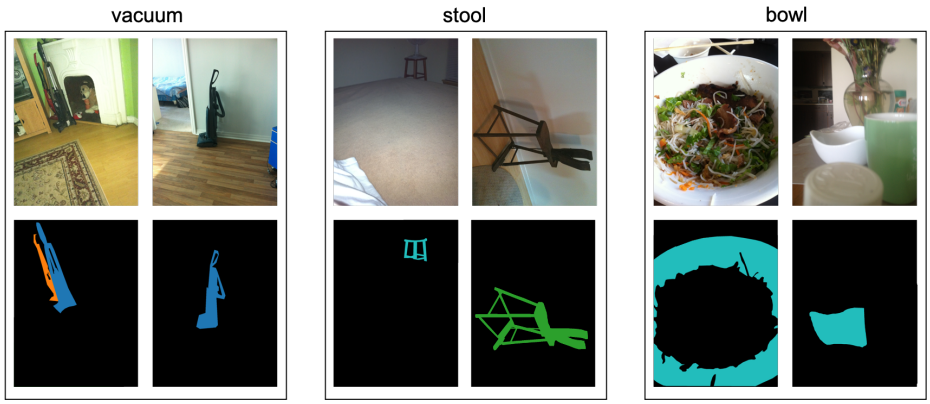


Fig. 9: Examples of object categories which frequently contain holes.

7 Analysis of the prevalence of instances and categories

We find that our dataset has, on average, 2.17 annotated objects per image in contrast to 7.33 per image in COCO. We additionally find that each category in our dataset, on average, represents 1.00% of all instances in comparison to COCO where each category represents, on average, 1.25% of instances. We visualize a subset of our dataset's categorical distribution in Figure 10.

We analyze the co-occurences of categories across the images in our dataset. We observe that our dataset has, on average, 2.29 co-occurences per image versus 8.87 in COCO. We find the ten most co-occurring categories, in order, to be keyboard/monitor, person/rug, person/shoe, person/cup, person/bottle, person/sock, person/chair, person/couch, dog/collar, and bed/pillow. We suspect

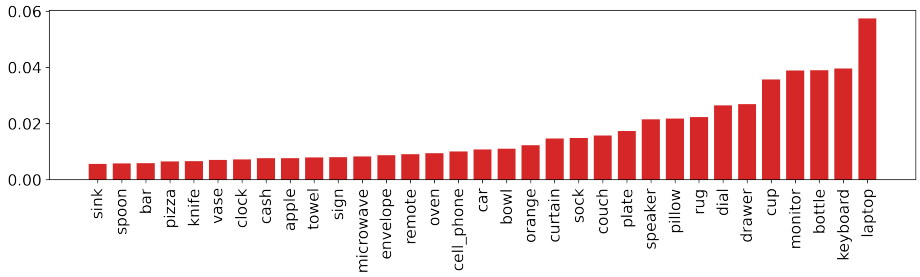


Fig. 10: Proportion of instances corresponding to each category for each third category in our dataset, sorted by frequency of holes.

that the person category appears frequently in co-occurrences because many of the common co-occurring categories are worn or held by people.

8 Examples of few-shot object detection results on VizWiz-FewShot-OD-25^{i}

We visualize the detection results from the few-shot object detection state-of-the-art model, DeFRCN [6]. We randomly selected these images. For each example, we show one object category and the predictions in that category from 1-, 3-, 5-, and 10-shot models, respectively. Results are shown in Figure 11. In some cases, none of the 1-, 3-, 5-, and 10-shot models are able to detect the ground truth object category. Examples of these cases are shown in Figure 12. We observe that these extremely difficult cases include large background objects (i.e. rugs and couches) and objects with low visibility of identifying features (i.e. people and monitors).

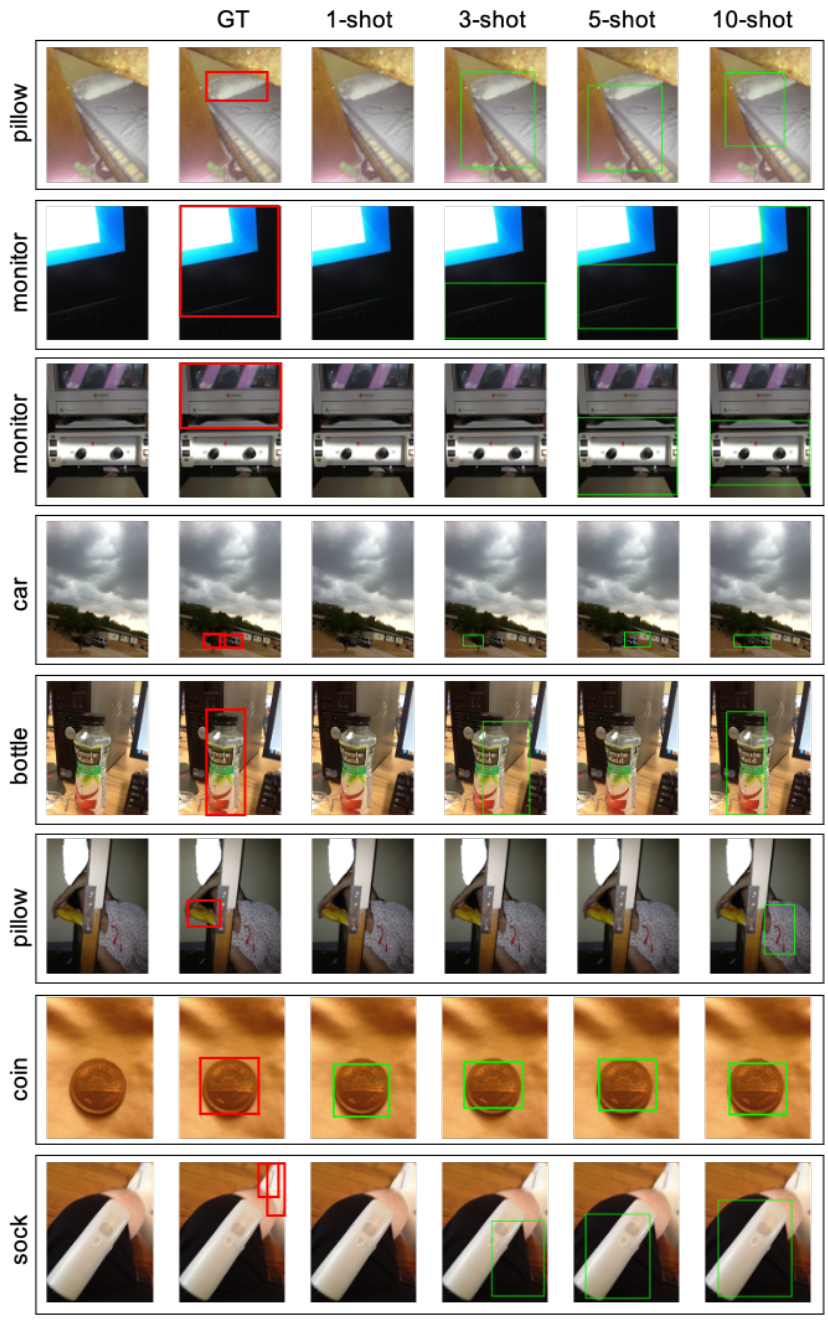


Fig. 11: 1-, 3-, 5-, and 10-shot object detection results from state-of-the-art few-shot object detection model, DeFRCN [6], on our VizWiz-FewShot-OD-25^i. While improvements are seen in some examples with more shots, the 10-shot object detection model is not able to perform well on our dataset.

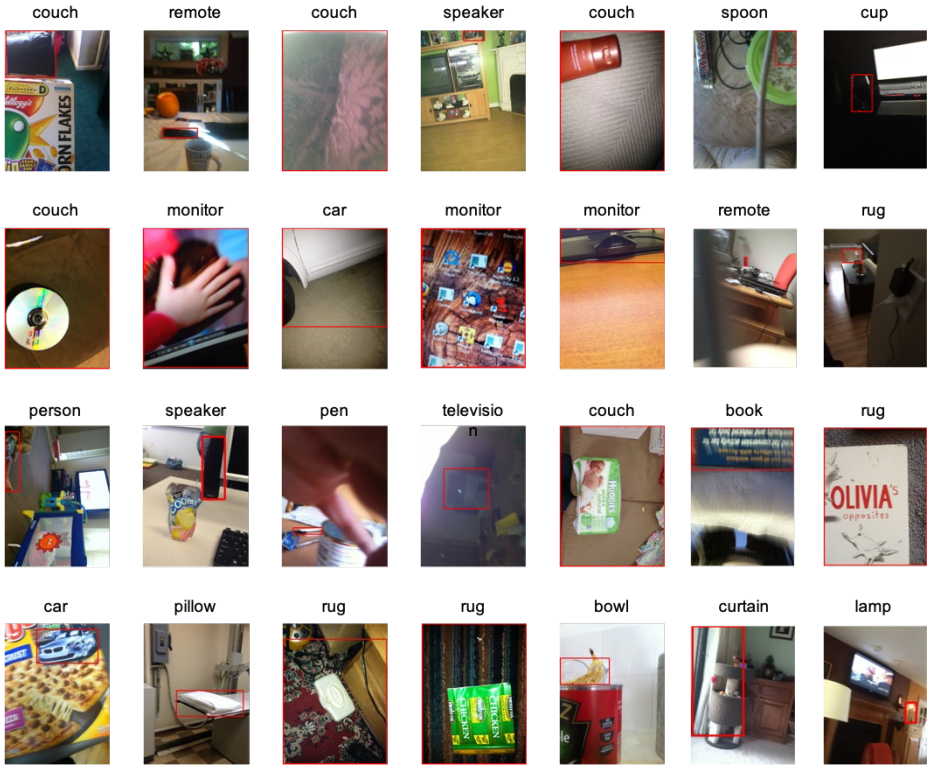


Fig. 12: Examples of difficult cases where the state-of-the-art few-shot object detection model, DeFRCN [6], is not able to detect the correct object categories. Difficult cases include a large amount of background and partially visible objects.

References

1. Amirreza Shaban, Shray Bansal, Z.L.I.E., Boots, B.: One-shot learning for semantic segmentation. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 167.1–167.13 (September 2017)
2. Fan, Q., Zhuo, W., Tang, C.K., Tai, Y.W.: Few-shot object detection with attention-rpn and multi-relation detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
3. Li, X., Wei, T., Chen, Y.P., Tai, Y.W., Tang, C.K.: Fss-1000: A 1000-class dataset for few-shot segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755 (2014)
5. Nguyen, K.D.M., Todorovic, S.: Feature weighting and boosting for few-shot segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 622–631 (2019)
6. Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., Zhang, C.: Defrcn: Decoupled faster r-cnn for few-shot object detection. ArXiv (2021)