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

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

University of Colorado Boulder

Abstract. We introduce a few-shot localization dataset originating from photographers who authentically were trying to learn about the visual content in the images they took. It includes nearly 10,000 segmentations of 100 categories in over 4,500 images that were taken by people with visual impairments. Compared to existing few-shot object detection and instance segmentation datasets, our dataset is the first to locate holes in objects (e.g., found in 12.3% of our segmentations), it shows objects that occupy a much larger range of sizes relative to the images, and text is over five times more common in our objects (e.g., found in 22.4% of our segmentations). Analysis of three modern few-shot localization algorithms demonstrates that they generalize poorly to our new dataset. The algorithms commonly struggle to locate objects with holes, very small and very large objects, and objects lacking text. To encourage a larger community to work on these unsolved challenges, we publicly share our annotated few-shot dataset at `https://vizwiz.org`.

Keywords: Few-shot learning, object detection, instance segmentation

1 Introduction

Our paper is motivated by the belief that people who are blind or with low vision (BLV) would benefit from the ability to locate objects in images that they take, whether with a bounding box or fine-grained segmentation. For people with low vision, localization would enhance their use of magnification tools [4, 30] by automatically enlarging the content of interest. For all BLV users, they could have stronger privacy guarantees with services[1] that describe their images if object localization algorithms were used in place of recognition algorithms. That is because services could use localizations to obfuscate all content except the detected regions needed to justify predictions[2] and so remove accidentally captured private information in the background of images, which is a common occurrence for people with vision impairments [16]. Finally, automatic localization would also

[1] Visual assistance services include Microsoft's Seeing AI, Google's Lookout, and TapTapSee. The popularity of such services is exemplified by companies' reports about hundreds of thousands of users and tens of millions of requests [9, 22, 12].

[2] Recorded evidence can be needed by companies for legal reasons.

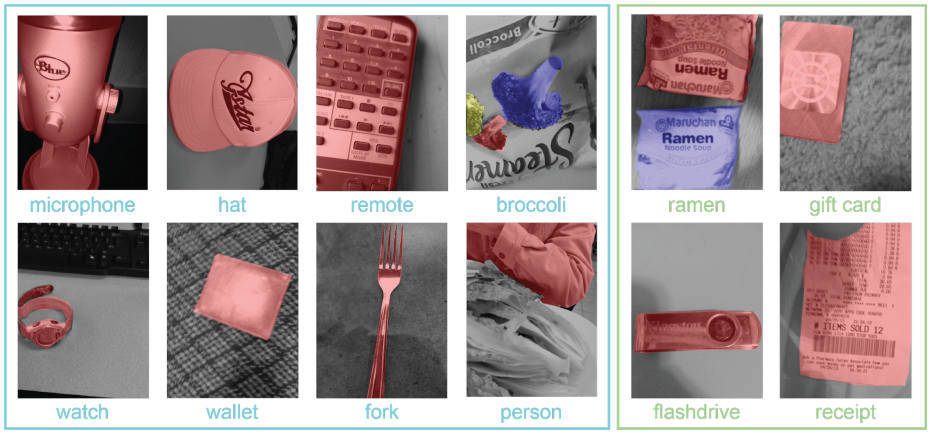

Fig. 1: Examples from our VizWiz-FewShot dataset showing instance segmentation annotations we collected for images taken by people with vision impairments. Annotated categories span those that are in common with prior work (left green box) and unique to our dataset (right blue box). These examples highlight novel aspects of our dataset, including that holes are permitted in our objects, objects vary considerably in how much of the image they occupy, and objects often feature text. (Annotation overlay colors were selected based on the order the instance segmentations appear in our dataset, and so not object categories.)

support users to independently edit their images, which is a feature some BLV photographers have requested.

Observing that BLV photographers take pictures showing a large number of objects (e.g., 16,400 nouns were used to describe less than 40,000 images taken by BLV photographers [18]), we are interested in the problem of few-shot learning. Casting the problem as a few-shot learning problem means that developers can efficiently scale up the number of categories supported in order to locate the long-tail of categories. That is because few-shot learning methods learn to locate a novel object category by observing only K annotated examples, where K is typically 1, 5, or 10 examples.

To support our aim, we introduce a few-shot localization dataset that consists of 100 segmented categories in over 4,500 images taken by people with vision impairments. The images were taken in authentic use cases where the photographers were soliciting human assistance to learn about their visual surroundings [3]. Examples of annotated images in our dataset are shown in Figure 1.

We next analyze how our dataset compares to the four existing few-shot localization datasets [1, 28, 26, 13, 23] to reveal both how our dataset is similar to and different from prior work. We observe several unique aspects about our dataset. First, it is the only dataset that indicates when and where holes are located in objects. Holes are observed in 12.3% of our instance segmentations. Second, our dataset's objects exhibit a much larger range of sizes relative to the image sizes. Finally, our dataset's objects contain text much more frequently.

Specifically, our analysis shows that 22.4% of objects in our dataset contain text versus 4.6% of the objects in the related instance sementation dataset, COCO-20^i [26]. We suspect the latter two unique aspects of our dataset stem from how images were curated. While our images come from a real-world application where photographers were authentically trying to learn about their visual surroundings, existing datasets were contrived by scraping images from photo-sharing websites. Altogether, we believe that our new dataset fills important gaps of existing datasets in the vision community by capturing a greater diversity of challenges that can arise in real-world applications.

We also benchmark top-performing few-shot learning object detection and instance segmentation algorithms on our new dataset. We find that the algorithms perform poorly overall. Our fine-grained analysis reveals that the algorithms commonly fail for objects that contain holes, very small and very large objects, and objects that lack text.

In summary, our contributions include: (1) a new few-shot localization dataset based on images that were taken in a real-world application, (2) the first few-shot localization dataset with metadata showing where holes are located in objects, (3) fine-grained analysis revealing unique aspects of our dataset compared to existing few-shot localization datasets, and (4) analysis of top-performing few-shot localization algorithms that reveals open algorithmic challenges for the vision community. We expect this work will encourage the development of algorithms that can handle a greater diversity of challenges that arise in real-world applications. We expect these advancements will, in turn, benefit a larger audience by facilitating the improvement of algorithms for application domains, such as robotics and wearable lifelogging, that face similar challenges including holes in objects, varying object sizes, and presence of text.

2 Related Work

Few-Shot Learning Datasets for Image Localization. Several dataset challenges have been proposed for few-shot object detection and few-shot instance segmentation: PASCAL-5^i [1], COCO-20^i [28, 26], ImageNet-LOC [8], and FSOD Dataset [13]. A limitation of existing datasets is that images come from contrived settings rather than authentic use cases where people are seeking to learn about their images. Specifically, images were curated by scraping images from the Internet that were tagged with pre-defined categories of interest. To our knowledge, we are introducing the first few-shot dataset challenges based on images that originate from authentic use cases where people took pictures to learn about the content. Our dataset offers new categories that are applicable to real-world applications. In addition, our dataset provides metadata showing holes in objects, which is a unique feature that creates new challenges toward few-shot problems. Finally, it provides additional real-world challenges such as a larger range of object sizes and a higher prevalence of objects containing text.

Few-Shot Algorithms for Image Localization. Few-shot learning was introduced to the community in 2017 for object detection [10] and in 2018 for instance seg-

mentation [26]. Since, a large number of algorithms have been proposed that largely are based on two types of approaches: meta-learning and fine-tuning. To assess how state-of-the-art methods perform on our new dataset, we benchmark the top-performing few-shot object detection and instance segmentation algorithms for which code is publicly-available. Overall, we observe poor performance from these algorithms [29, 26, 27, 5]. From our fine-grained analysis, we find this dataset is challenging for algorithms due to the presence of holes in objects, very small objects, very large objects, and objects that lack text.

Datasets Originating from People With Vision Impairments. In recent years, a growing number of publicly-available datasets have been proposed to facilitate the development of algorithms that can work well on images taken by people with vision impairments [2, 6, 7, 17–19, 25, 15, 16, 32]. For example, existing datasets support the development of algorithms for predicting answers to visual questions [17, 19], recognizing objects in videos [25], and describing images with captions [18]. Complementing prior work, we introduce a dataset for localizing objects in images taken by BLV photographers, either using a bounding box or segmentation. We expect success with developing localization algorithms for images taken by BLV photographers to directly benefit BLV photographers and to, more generally, support a larger number of real-world applications that encounter similar visual characteristics found in our dataset, such as robotics and wearable lifelogging applications.

3 VizWiz-FewShot Dataset

We introduce a dataset that we call "VizWiz-FewShot". It consists of localization annotations for images taken by people with vision impairments who authentically were trying to learn about their visual surroundings.

3.1 Dataset Creation

Data Source. Our dataset extends the VizWiz-Captions dataset [18], which consists of images taken by people with vision impairments paired with five crowdsourced captions per image. The photographers took and shared these images in order to solicit assistance from remote humans in recognizing the contents in the images [3]. We leverage the data in both the train and validation splits, which offers a starting point of 31,181 captioned images.

Category Selection. We chose 100 categories to locate in the images. These categories both support backward compatibility with popular few-shot localization datasets and reflect important categories for people with vision impairments. To select the categories, we first quantified the frequency of all nouns that appeared in at least two of the five captions per image for our images. We then selected 72 non-ambiguous categories that overlap with four existing few-shot localization datasets: MS COCO [24], PASCAL VOC [11], FSS-1000 [23], and FSOD [13].

We also selected 28 non-ambiguous categories that are unique to our target population, by choosing categories that refer to physical objects. All 100 selected categories have at least 10 examples.

Data Filtering. We next filtered the images to only retain those that contained at least one of our 100 categories. First, we removed images which did not mention any of our categories within at least two of their respective captions. Then, the authors subsequently verified that each remaining image contained at least one object that fit the precise definition of at least one of our categories. For example, our automatic collection of images with the category "pen" retrieved some images of pencils without pens and so we removed those images. After filtering, we had total of $4,930$ images.

Annotation Tasks. After iterative prototyping, we settled on a workflow similar to prior work [24], such that we first used an image classification task to flag which categories of interest are present in each image and then an instance segmentation task to locate every instance of each category. For both tasks, we utilized templates provided by Amazon Mechanical Turk (AMT).

For image classification, crowdworkers were shown an image and asked to select all categories that were present, if any. Since showing all 100 categories at the same time could overwhelm crowdworkers and ultimately lead to lower quality results, we instead showed a subset of categories at a time (i.e., $\sim$20).

For instance segmentation, crowdworkers were shown an image with the list of categories known to be present from the image classification task and asked to locate each instance of every category. Like prior work, our annotation tool supported users to create a series of clicks to generate polygons. Going beyond prior work, in addition to being able to draw 'positive' polygons to locate object boundaries, our tool also enabled users to create 'negative' polygons in order to capture when objects contained holes. We offered extensive instructions with our task to cover edge-case scenarios, including how to annotate the presence of holes and how to handle occlusions.

Annotation Collection. We implemented several quality control methods to support our collection of high-quality annotations. First, we only accepted workers who already had completed at least 500 AMT tasks with at least a 99% approval rating. For the more complex instance segmentation task, we also required workers to successfully pass a qualification test consisting of nine challenging annotation edge cases (described in the Supplementary Materials). We then collected redundant results from multiple unique workers for both tasks. For image classification, we collected three results per image and flagged a category as present if at least one worker indicated so. For instance segmentation, we collected two sets of annotations per image-category pair and then computed intersection over union (IoU) scores to determine how to establish a ground truth segmentation per image. When $IoU \geq 0.8$, we randomly chose one of the annotations as the ground truth. Otherwise, the authors reviewed the pair of annotations to choose one as the ground truth (or, in exceptional cases, discarded both annotations).

Finally, we paid above minimum wage to better incentivize the workers.[3] Upon completion, we had a total of 9,861 segmented objects in 4,622 images.

3.2 Dataset Analysis

We now analyze the VizWiz-FewShot dataset and compare it to the other mainstream few-shot localization datasets.

VizWiz-FewShot-IS (Instance Segmentation). We first characterize our few-shot instance segmentation dataset and compare it to the only other few-shot instance segmentation dataset we are aware of: COCO-20^i [24], which has a total of 80 categories. We compute for every instance segmentation the following metrics:

- **Mass center**: location of the center of mass pixel for each object relative to the image coordinates. Consequently, an object's x-coordinate and y-coordinate values can range from 0 to 1.
- **Boundary complexity**: ratio of the area of an instance to the length of its perimeter, also known as isoperimetric inequality. Values range from 0 to 1, with lower values representing more complex boundaries.
- **Image coverage**: percentage of pixels each instance segmentation occupies from the entire image.
- **Prevalence of text**: flag indicating if Microsoft Azure's optical character recognition (OCR) API returned text for an image, after masking out all content except for the instance segmentation.
- **Prevalence of holes**: flag indicating if any holes are present paired with the percentage of pixels each hole occupies from the instance segmentation when any holes are present.

In what follows, we report the statistics summarizing the results for all instance segmentations for each dataset.[4]

Results for *boundary complexity*, object location (i.e., *mass center*), and *image coverage* are shown in Figure 2(a). Amongst these metrics, the only major difference between the two datasets is *image coverage*. For example, objects in our dataset represent on average roughly six times more relative area in images than those in COCO-20^i. We exemplify this finding qualitatively by showing in Figure 2(b) how annotations of two types of content, "sink" and "oven", dramatically differ in image coverage across the two datasets. We attribute the prevalence of larger relative object sizes in our dataset to the fact that photographers in an authentic use case where they are trying to learn about content take up-close pictures of the content. Another key distinction about our dataset is that we observe a considerably larger variability for the image coverage in our dataset. Qualitative results in Figure 3 exemplify this range of relative area occupied by segmentations in our dataset. This finding highlights that a benefit of our dataset is that it encourages the design of algorithms that will be able to handle a larger range of relative object sizes in images.

[3] Average hourly wage was $8.00 and $9.61 for classification and IS respectively.

[4] For efficiency, we evaluated the presence of text for a random sample of images in COCO-20^i that is comparable to the number of images in our dataset: 8,000.

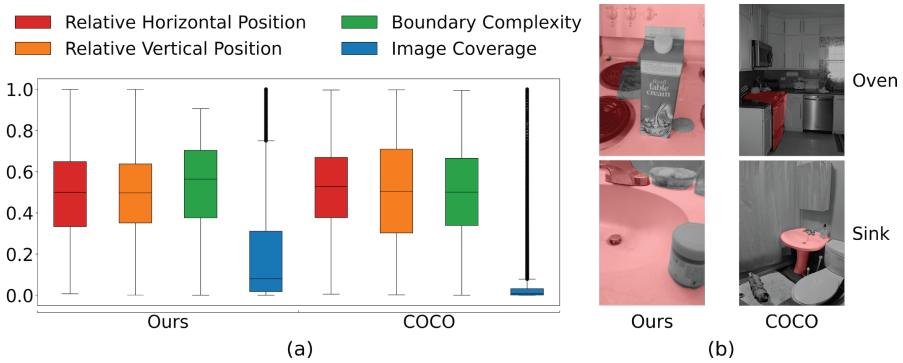

Fig. 2: Comparison of our dataset with the only existing few-shot instance segmentation dataset, COCO-20^i. (a) Summary statistics for all segmented objects in each dataset are shown in the box plot with respect to the location (i.e., relative x-coordinates and y-coordinates for mass center), boundary complexity, and image coverage. The box plot's central mark denotes the median score, box edges the 25th and 75th percentiles scores, whiskers the most extreme data points not considered outliers, and individually plotted points the outliers. (b) Annotations from both datasets exemplify our quantitative finding that an object with larger image coverage is an outlier in COCO-20^i while common in our dataset.

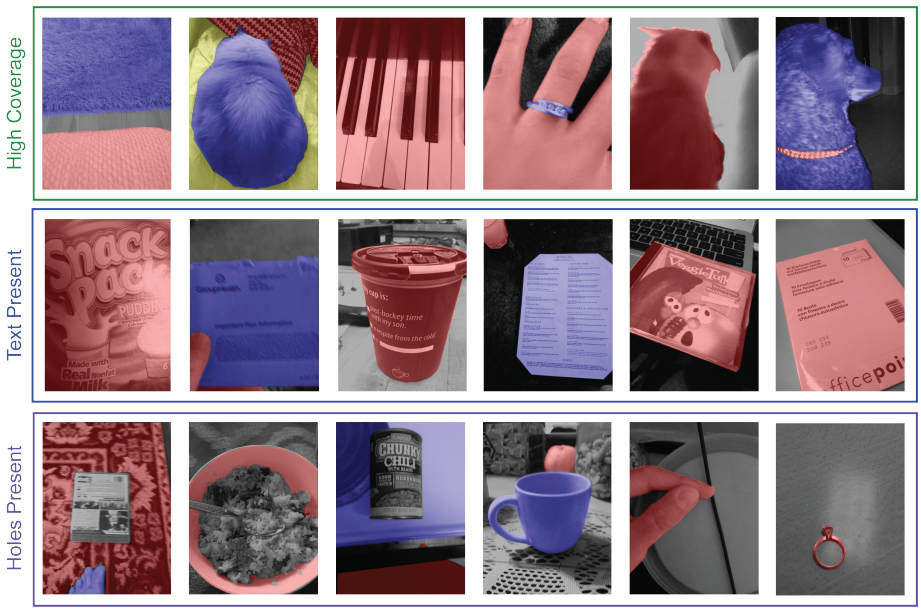

Fig. 3: Examples from our VizWiz-FewShot dataset illustrating its unique aspects, specifically the high variability of object size relative to images, high prevalence of text in objects, and inclusion of holes in segmentations.

When analyzing the *prevalence of text*, we find that 22.4% of instances in our dataset include text compared to only 4.6% in COCO-20^i. We show examples of objects in our dataset that contain text in Figure 3, including of cups, menus, cereal boxes, and albums. We also show in Figure 4 the frequency at which text is found in a sample of our categories. Categories that more commonly contain text include ramen, food menu, packet, and gift card. Categories that rarely contain text include dog, vase, house, and spoon. We hypothesize from our findings that algorithm developers working on COCO-20^i may have a bias to disregard text. We expect our dataset will inspire developers to consider how to take advantage of text recognition methods as potential predictive cues for locating objects with few-shot localization algorithms.

Finally, a unique feature of our dataset that is not supported in COCO-20^i is locating the *holes* in objects. We define a hole as any area in an object that does not belong to the object itself since our goal is to locate all pixels belonging to each category of interest. Thus, a hole may manifest as a property of an object itself (i.e. a ring), an object's orientation (i.e. a side view of an open armrest on a chair), or an occlusion on the object (i.e. a plate partially occluded by food). In total, 12.3% of the instances in our VizWiz-FewShot-IS contain holes. As shown in Figure 5, some of the object categories with the highest proportion

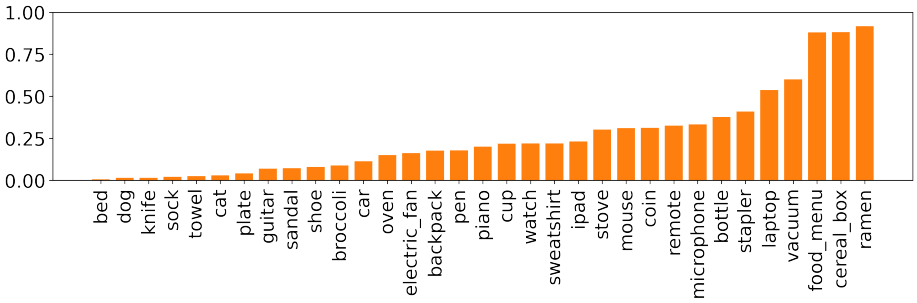

Fig. 4: Proportion of instances with text on a per-category basis for each third category in our dataset, sorted them by frequency of text.

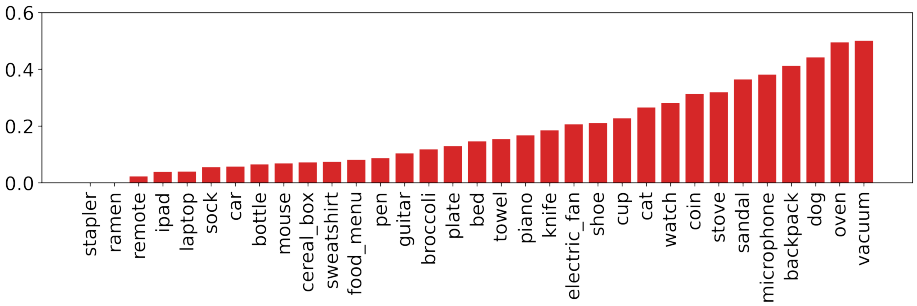

Fig. 5: Proportion of instances with holes on a per-category basis for each third category in our dataset, sorted by frequency of holes.

of instance segmentations that contain holes are chairs, sandals, bracelets, and bowls. For instance, 21.1% of the bowl instances have holes, likely because bowls typically contain food in them. We attribute the high frequency of holes to two causes. First, is that the objects intrinsically contain them; e.g., chairs, stools, and sandals. The second reason is that large-appearing objects get occluded by foreground objects, such as occlusions on rugs, bowls, and plates. Corroborating this hypothesis, we find that the percentage of instances with holes increases with object size, suggesting that larger objects tend to have hole-type occlusions more frequently than smaller objects (results shown in the Supplementary Materials). We also observe that certain categories regularly have a larger percentage of hole pixels in them, such as bowls which typically are occluded by a large amount of food (results are shown in the Supplementary Materials). We anticipate the need to recognize holes will increase our dataset's difficulty for computer vision models since they will need to go beyond merely locating the outermost boundary of objects to also understanding which interior pixels should belong to the objects.

VizWiz-FewShot-OD (Object Detection). We next characterize our dataset in the object detection setting and how it compares to the three mainstream few-shot object detection datasets: COCO-20^i [28], PASCAL-5^i [1], and FSOD [13].[5] To support comparison, we convert each instance segmentation in our dataset into its bounding box representation. For every dataset, we compute for each object detection its relative position and image coverage.[6] Summary statistics for each dataset are shown in Figure 6.

One key distinction of our dataset is the greater variability in the *relative positions* of its objects. This finding contrasts a common photographer's bias

[5] We use both the train and validation splits from each of the mainstream datasets for analysis. We randomly sample 10% of the annotations from COCO-20^i due to its large size, and we use all annotations from PASCAL-5^i and FSOD.

[6] We exclude from consideration the other three metrics used to analyze the instance segmentations because boundary complexity is no longer relevant, text prevalence could be incorrect due to the bounding box extending beyond an object's boundaries, and none of the other datasets located holes in objects.

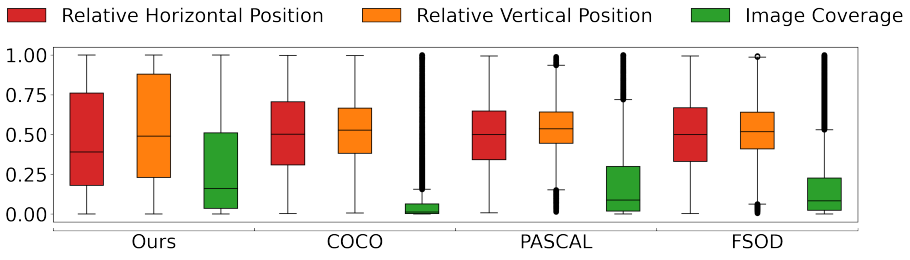

Fig. 6: Box plot showing how objects in our dataset compare to those in the three existing few-shot object detection datasets with respect to relative position and image coverage.

of beautifully capturing the contents of interest near the center of images. We suspect this greater diversity of object positions stems from the inability of BLV photographers to inspect the images to guarantee that they are centering the contents of interest in their images and their inability to verify that clutter gets excluded from the background of their images.

Another distinction in our dataset is its bias towards having objects positioned on the left side of images, as exemplified by the mean and median relative horizontal position being 0.45 and 0.39 respectively. One possible reason for this bias may be a commonality in how the photographers take images. Specifically, when a person is trying to learn about a particular object often the person holds the content of interest in the left hand while taking a picture of it with the right hand. This scenario assumes a tendency in society for people to be right-handed.

Finally, we observe that bounding boxes in our dataset tend to cover more of an image than two of the four existing datasets: COCO-20^{i} and FSOD. Image coverage of objects in our dataset is comparable to PASCAL, which we attribute to PASCAL's focus on iconic images with salient objects [24] and our dataset's inclusion of images of objects taken up-close for visual assistance.

4 Algorithm Benchmarking

We now present our results from benchmarking top-performing computer vision algorithms on our VizWiz-FewShot dataset.

To support use of our annotated data for few-shot localization tasks, we create a 4-fold cross-validation format and split our 100 object categories into four sets, where $i = 0, 1, 2, 3$ for the i^{th} fold. This approach mimics the settings used for the few-shot datasets PASCAL-5^i [1] and COCO-20^i [28, 26]. We refer to the resulting datasets for few-shot instance segmentation and few-shot object detection as VizWiz-FewShot-IS-25^i and VizWiz-FewShot-OD-25^i respectively.

We evaluate the trained models using mAP and mAP_{50}. mAP originates from the MS COCO object detection challenge [24] and is frequently used to evaluate algorithms for FSIS [26, 31, 14, 27] and FSOD [20]. mAP refers to the mean of Average Precision (AP) for all categories and is an average across the IoU threshold of $0.5 : 0.05 : 0.95$ for ground truth and prediction regions. The only difference betweent FSOD and FSID is that that former is evaluated based on bounding boxes while the latter is evaluated based on mask areas. We also present results with respect to mAP_{50}, where only threshold 0.5 is used, since this approach facilitates the comparison with datasets such as Pascal VOC. Our evaluation is based on when $K = 1, 3, 5, 10$ shots are available.

4.1 Few-shot Instance Segmentation Algorithms

We benchmarked the top-performing FSIS algorithm for which code is publicly-available and can be successfully deployed on modern GPUs[7]. Specifically, we

[7] We discuss the limitations of other FSIS algorithms for benchmarking on our dataset in the Supplementary Materials.

evaluated the algorithm YOLACT [5], which was originally proposed outside of a few-shot setting, and then was subsequently shown to yield strong results on COCO-20^i when fine-tuned for FSIS [27]. When using the codebase as is on our new FSIS dataset, the performance on novel classes is consistently negligible (i.e., mAP around 0). We found this occurs because the default hyperparameters leads to training loss explosion. Consequently, we tested with different hyperparameters. Specifically, we (1) explored four learning rates in decreasing order from the original setting (i.e., 1e-3) to a value where saw convergence (i.e., 2e-5), (2) explored weights for the bounding box loss and mask loss in increasing order from 0 to 15 with an increment size of 1, (3) resized all images to match MS COCO's resolutions (i.e., 640×480), and (4) removed object instances of which the areas exceed that of MS COCO (i.e., instances are filtered based on the size range in MS COCO).

Overall performance: Results are shown in Table 1. We report results with respect to each fold as well as the mean across all folds.

Overall, the model performs poorly on VizWiz-FewShot-IS-25^i. Moreover, the performance is much worse on our dataset than observed on the original dataset for which it was proposed [27]; e.g., mAP_{50} score of 2.48 compared to 17.1 for 1-shot and 5.17 compared to 18.9 for 5-shot for VizWiz-FewShot-IS-25^i and COCO-20^i respectively. These findings motivate the benefit of our dataset in providing a challenging problem for the vision community.

Fine-grained analysis: To identify what make the dataset difficult, we next analyze the model's performances with respect to (1) image quality, (2) object size,

Table 1: Overall performance of the few-shot algorithms on our VizWiz-FewShot dataset presented in 4-fold validation style. The FSIS algorithm is benchmarked on VizWiz-FewShot-IS-25^i, and the FSOD algorithms are benchmarked on VizWiz-FewShot-OD-25^i.

		25^0		25^1		25^2		25^3		mean	
	shots	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}	mAP	mAP_{50}
FSIS YOLACT	$k=1$	1.87	2.5	2.91	3.51	1.39	1.79	1.08	2.13	1.81	2.48
	$k=3$	2.31	2.81	4.48	5.24	2.35	2.78	3.59	4.52	3.18	3.84
	$k=5$	3.45	4.30	4.84	5.67	4.34	5.14	4.39	5.56	4.25	5.17
	$k=10$	5.97	7.69	7.71	9.02	6.18	7.18	5.82	7.38	6.42	7.82
FSOD DeFRCN	$k=1$	3.45	5.80	4.67	8.33	3.51	5.10	4.51	8.19	4.03	6.85
	$k=3$	6.80	11.65	7.81	13.85	7.26	11.74	7.88	14.05	7.43	12.82
	$k=5$	8.99	15.19	11.26	19.13	10.60	16.95	11.23	19.11	10.52	17.60
	$k=10$	11.24	21.34	13.36	25.68	11.94	22.07	13.91	24.76	12.61	23.46
FSOD YOLACT	$k=1$	2.05	2.61	2.84	3.66	1.61	1.97	1.91	2.26	2.10	2.63
	$k=3$	2.45	3.05	4.41	5.53	2.58	3.22	3.94	4.89	3.35	4.17
	$k=5$	3.46	4.44	4.87	5.88	4.82	5.68	4.72	5.81	4.47	5.45
	$k=10$	6.27	7.89	7.60	9.29	6.61	7.90	6.06	7.86	6.64	8.24

and (3) presence of text. To do so, we distribute the test examples into subsets with respect to each of the following factors:

- **Image quality**: Leveraging metadata from prior work [18] which indicates how many from five crowdworkers indicated an image is insufficient quality to recognize the content, we classify an image as "high quality" when none indicate insufficient quality and "medium quality" when one or two crowdworkers flagged the image as insufficient quality. We exclude even lower quality images from our analysis since these are rare in our test set.
- **Object size**: The target object size is calculated based on the number of pixels in the instance segmentations. We divide the dataset into small, medium, and large sizes, such that the numbers of images in each set are evenly distributed. This resulted in the following thresholds: 350^2 and 900^2.
- **Presence of text**: We used the metadata collected for Section 3 that determined whether an object has text on it using OCR on background-masked instance images to flag whether text is present.
- **Presence of hole(s)**: We used additional metadata from Section 3, indicating if each instance segmentation contains a hole, to flag if a hole is present.

All fine-grained analysis results for YOLACT are shown in Table 2.

With respect to *image quality* and *object size*, our findings reinforce those of prior work. Specifically, like prior work [18], the algorithm typically performs better on images with higher quality. Like other prior work [21, 33], algorithms typically perform worse for smaller objects. However, our findings extend those reported in [21, 33] since we define object sizes differently; i.e., they use smaller thresholds of 32^2 and 96^2. To our knowledge, our work is the first to offer insights

Table 2: Fine-grained analysis on the performance of FSIS and FSOD models on VizWiz-FewShot presented in mAP.

		Image Quality		Object Size			Presence of Text		Presence of Holes	
	shots	Medium	High	Small	Medium	Large	Yes	No	Yes	No
FSIS YOLACT	$k=1$	1.24	**2.11**	1.38	**2.19**	1.74	**1.83**	1.62	1.48	**1.99**
	$k=3$	**3.31**	3.19	2.24	3.44	**3.80**	**3.26**	2.84	2.91	**3.21**
	$k=5$	3.72	**4.29**	2.64	3.88	**5.19**	3.78	**4.05**	3.06	**4.22**
	$k=10$	6.11	**6.50**	3.94	6.53	**7.30**	**6.16**	5.28	5.82	**6.29**
FSOD DeFRCN	$k=1$	**2.46**	2.22	2.67	2.96	**3.57**	**4.97**	2.29	0.875	**3.99**
	$k=3$	4.97	**5.26**	6.13	5.41	**6.81**	**7.93**	7.60	2.15	**7.60**
	$k=5$	**10.69**	10.27	6.83	**16.95**	8.90	**13.48**	12.70	2.56	**9.78**
	$k=10$	12.82	**13.62**	12.23	**18.18**	11.48	**17.45**	15.96	5.37	**12.49**
FSOD YOLACT	$k=1$	1.36	**2.23**	1.49	**2.16**	1.93	**2.06**	1.72	1.71	**2.10**
	$k=3$	**3.42**	3.36	2.51	3.42	**4.05**	**3.44**	3.05	3.30	**3.33**
	$k=5$	3.99	**4.51**	3.02	3.80	**5.24**	3.78	**4.31**	3.40	**4.40**
	$k=10$	6.19	**6.70**	4.43	6.36	**7.31**	**6.18**	5.60	6.09	**6.45**

into performance on larger objects due to the novel presence of such larger instances in our dataset.

With respect to the *presence of text*, overall the performance is slightly better for instances that contain text. Initially, we found this surprising. We expected the opposite trend since we suspected that the limited prevalence of text in prior datasets would have led algorithm designers to not consider the presence of text in their algorithm designs. We suspect part of the reason for our finding is that, if the text on the objects is clear enough to be visible, then the image is high quality. Additionally, the high frequency information from text regions in instance segmentations may be valuable predictive cues, despite the absence of the ability to recognize the text as text. Finally, the presence of text has a strong correlation with particular categories, which may influence our findings.

Finally, with respect to the *presence of holes*, the performance is consistently worse for objects that contain holes. The presence of holes raises the task complexity dramatically by requiring algorithms to go beyond locating object boundaries to also have a semantic understanding of all pixels within the object boundary. According to our analysis in Section 3 and the Supplementary Materials, objects with larger sizes tend to have more coverage by holes, including due to occlusion. Therefore, we suspect that the poor performance that we observed for larger sized objects could be correlated with the poor performance we observing with our analysis here on objects with holes.

4.2 Few-Shot Object Detection

We benchmarked two FSOD algorithms for which code is publicly-available. First, we chose Decoupled Faster R-CNN (*DeFRCN*) with its default hyperparameters [29], since it is the state-of-the-art FSOD model. It follows a two-stage fine-tuning paradigm. For our k-shot experiments, we randomly sample k images to use for fine-tuning the model. We also benchmark the YOLACT model used for FSIS by converting its segmentation results into bounding boxes.

Overall performance: Overall results are shown in Table 1. These results resemble those observed for FSIS. Specifically, both algorithms perform poorly on our dataset and much worse on our dataset than reported for the original dataset on which they were evaluated. These results reinforce that our new dataset offers distinct challenges from existing datasets for FSOD algorithms.

Fine-grained analysis: We perform the same fine-grained analysis conducted for FSIS with the two benchmarked FSOD models, and results are also reported in Table 2. While we observe that the level of *image quality* does not correlate with algorithm performance, we do observe performance trends for the other three factors. Moreover, these trends match those discussed for FSIS. Specifically, both benchmarked models tend to perform the worst for small objects, perform better when text is present, and perform worse when holes are present.

Table 3: Generalization of models trained on MS COCO for few-shot object detection to matching categories in our VizWiz-Fewshot-OD dataset.

		mAP				mAP_{50}			
model	testing set	1	3	5	10	1	3	5	10
DeFRCN	MS COCO	6.63	12.32	14.20	16.69	12.50	21.69	24.87	29.15
	VizWiz	1.32	3.43	2.17	4.57	2.74	5.86	3.39	6.53

Cross-dataset analysis: Finally, we evaluated DeFRCN's generalization performance across datasets.[8] To do, so we randomly selected 20 of the 37 categories found in both MS COCO and VizWiz-FewShot-OD-25^i as novel classes. Next, we trained DeFRCN on the remaining 60 MS COCO classes and then fine-tuned it with k-shot images randomly sampled from the 20 novel classes in MS COCO. The resulting model was evaluated on both the MS COCO test set as well as our VizWiz-FewShot-OD-25^i test set. Results are shown in Table 3. We observe significant gaps between scores on MS COCO and our dataset revealing that the algorithm generalizes poorly when encountering the domain shift between the two datasets. These findings reinforce that images in our dataset offers distinct algorithmic challenges from those observed in MS COCO.

5 Conclusions

We introduce the VizWiz-FewShot dataset to facilitate the community in designing few-shot learning models for object detection and instance segmentation that work well for the diverse set of challenges that emerge in real-world applications. Our benchmarking of top few-shot localization algorithms reveal that valuable directions for future work are to better support objects that contain holes, very small and very large objects, and objects that lack text.

Acknowledgments. This project was supported in part by a National Science Foundation SaTC award (#2148080) and gift funding from Microsoft AI4A. We thank Leah Findlater and Yang Wang for contributing to this research idea and the anonymous reviewers for their valuable feedback to improve this work.

[8] Of note, we also conducted cross-dataset experiments with YOLACT in the FSIS and FSOD settings however the cross-dataset performance was negligible. We attribute it to unsuccessful training with the chosen hyperparameters, both because the loss plateaued rather than converging with the new YOLACT hyperparameter values used in this paper and the loss exploded when using the original YOLACT values (i.e., the performance of YOLACT reported in the original paper could not be replicated when using the different set of training categories from MS COCO). In summary, the cross-dataset analysis results of YOLACT reinforce our initial findings that YOLACT performance is extremely sensitive to chosen hyperparameters and the training data, with custom tuning for each change.

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. Bhattacharya, N., Li, Q., Gurari, D.: Why does a visual question have different answers? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4271–4280 (2019)
3. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S., et al.: Vizwiz: nearly real-time answers to visual questions. In: Proceedings of the 23nd annual ACM symposium on User interface software and technology. pp. 333–342 (2010)
4. for the Blind, A.F.: Low vision optical devices, https://www.afb.org/node/16207/low-vision-optical-devices
5. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: Real-time instance segmentation. In: ICCV (2019)
6. Chen, C., Anjum, S., Gurari, D.: Grounding answers for visual questions asked by visually impaired people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19098–19107 (2022)
7. Chiu, T.Y., Zhao, Y., Gurari, D.: Assessing image quality issues for real-world problems. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3646–3656 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
9. Desmond, N.: Microsoft's Seeing AI founder Saqib Shaikh is speaking at Sight Tech Global, https://social.techcrunch.com/2020/08/20/microsofts-seeing-ai-founder-saqib-shaikh-is-speaking-at-sight-tech-global/
10. Dong, X., Zheng, L., Ma, F., Yang, Y., Meng, D.: Few-example object detection with model communication. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**, 1–1 (06 2018)
11. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (IJCV) **88**, 303–338 (2009)
12. Eyes, B.M.: Be My Eyes: Our story, https://www.bemyeyes.com/about
13. 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)
14. Fan, Z., Yu, J., Liang, Z., Ou, J., Gao, C., Xia, G., Li, Y.: FGN: fully guided network for few-shot instance segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9169–9178. Computer Vision Foundation / IEEE (2020)
15. Gurari, D., He, K., Xiong, B., Zhang, J., Sameki, M., Jain, S.D., Sclaroff, S., Betke, M., Grauman, K.: Predicting foreground object ambiguity and efficiently crowdsourcing the segmentation (s). International Journal of Computer Vision **126**(7), 714–730 (2018)
16. Gurari, D., Li, Q., Lin, C., Zhao, Y., Guo, A., Stangl, A., Bigham, J.P.: Vizwiz-priv: A dataset for recognizing the presence and purpose of private visual information in images taken by blind people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 939–948 (2019)

17. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: Answering visual questions from blind people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3608–3617 (2018)
18. Gurari, D., Zhao, Y., Zhang, M., Bhattacharya, N.: Captioning images taken by people who are blind. In: ECCV (2020)
19. J.-H. Kim, S. Lim, J.P.H.C.: Korean localization of visual question answering for blind people. SK T-Brain - AI for Social Good Workshop at NeurIPS (2019)
20. Jiaxu, L., Taiyue, C., Xinbo, G., Yongtao, Y., Ye, W., Feng, G., Yue, W.: A comparative review of recent few-shot object detection algorithms (2021)
21. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8419–8428 (Nov 2019)
22. Lee, S., Reddie, M., Tsai, C.H., Beck, J., Rosson, M.B., Carroll, J.M.: The emerging professional practice of remote sighted assistance for people with visual impairments. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–12 (2020)
23. 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)
24. 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)
25. Massiceti, D., Zintgraf, L., Bronskill, J., Theodorou, L., Tobias Harris, M., Cutrell, E., Morrison, C., Hofmann, K., Stumpf, S.: Orbit: A real-world few-shot dataset for teachable object recognition. In: ICCV 2021 (October 2021)
26. Michaelis, C., Ustyuzhaninov, I., Bethge, M., Ecker, A.S.: One-shot instance segmentation. ArXiv (2018)
27. Nguyen, K., Todorovic, S.: Fapis: A few-shot anchor-free part-based instance segmenter. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11094–11103 (2021)
28. 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)
29. Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., Zhang, C.: Defrcn: Decoupled faster r-cnn for few-shot object detection. ArXiv (2021)
30. Stangl, A.J., Kothari, E., Jain, S.D., Yeh, T., Grauman, K., Gurari, D.: Browsewithme: An online clothes shopping assistant for people with visual impairments. In: Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility. pp. 107–118 (2018)
31. Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., Lin, L.: Meta r-cnn: Towards general solver for instance-level low-shot learning. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2019)
32. Zeng, X., Wang, Y., Chiu, T.Y., Bhattacharya, N., Gurari, D.: Vision skills needed to answer visual questions. Proceedings of the ACM on Human-Computer Interaction **4**(CSCW2), 1–31 (2020)
33. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems **PP**, 1–21 (01 2019). https://doi.org/10.1109/TNNLS.2018.2876865