Open-vocabulary vs. Closed-set: Best Practice for Few-shot Object Detection Considering Text Describability

Anonymous Author(s)
Affiliation
Address
email

Abstract

Open-vocabulary object detection (OVD), detecting specific classes of objects using only their linguistic descriptions (e.g., class names) without any image samples, has garnered significant attention. However, in real-world applications, the target class concepts is often hard to describe in text and the only way to specify target objects is to provide their image examples, yet it is often challenging to obtain a good number of samples. Thus, there is a high demand from practitioners for few-shot object detection (FSOD). A natural question arises: Can the benefits of OVD extend to FSOD for object classes that are difficult to describe in text? Compared to traditional methods that learn only predefined classes (referred to in this paper as closed-set object detection, COD), can the extra cost of OVD be justified? To answer these questions, we propose a method to quantify the "text-describability" of object detection datasets using the zero-shot image classification accuracy with CLIP. This allows us to categorize various OD datasets with different text-describability and emprically evaluate the FSOD performance of OVD and COD methods within each category. Our findings reveal that: i) there is little difference between OVD and COD for object classes with low text-describability under equal conditions in OD pretraining; and ii) although OVD can learn from more diverse data than OD-specific data, thereby increasing the volume of training data, it can be counterproductive for classes with low-text-describability. These findings provide practitioners with valuable guidance amidst the recent advancements of OVD methods.

1 Introduction

Object detection plays a central role in research field of computer vision with a wide range of real-world applications [25, 33, 23, 38, 3, 53, 1, 54, 49, 20]. Historically, the problem is considered within a closed-set setting, where detectors are designed to identify only the predefined categories of objects encountered during the training process. Recently, the interest in open-vocabulary object detection (OVD) has been growing significantly. Leveraging large models [4, 26, 32, 10] that have learned a large amount of text or image-text pairs, it allows for the detection of specific classes of objects based solely on their linguistic descriptions (e.g., class names) without the need for image samples, making it "zero-shot" [48, 19, 24, 52, 42, 14, 11].

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

However, in real-world applications of object detection, there are often scenarios where the target classes are difficult to describe with words, such as various types of anomalies in industrial anomaly detection, or lesions that are difficult to identify in medical images. In these cases, it is only possible to specify the target objects by showing image examples, presenting a problem not directly addressed by OVD.

In reality, there is often the additional challenge of not having enough samples available; if sufficient samples were available, the standard supervised learning would work well. Therefore, there are high expectations from practitioners for few-shot object detection (few-shot OD), which can learn to detect objects from only a few examples.

While various approaches have been tried so far, the best approach to FSOD to date is a rather mediocre one that relies on transfer learning, where a detector pre-trained on some OD data is finetuned with a few-shot examples of the target objects [39, 36, 31]. This applies to the traditional OD in the closed setting as well as OVD; while OVD is originally designed for zero-shot detection, existing studies have also attempted to apply their OVD methods to few-shot OD settings, where the same finetuning is the standard [19, 24, 28, 45]. It should be noted that recent studies have tried to extend OVD to deal with visual prompts— examples to convey concepts that are hard to describe with words [11, 50, 14], but broadly speaking, this can be considered a type of few-shot OD.

Considering the above demands for FSOD and the recent advancements of OVD, a natural question that arises is *whether the benefits of OVD extend to few-shot OD for object classes that are difficult to describe with words.* Is it superior enough to justify the higher computational costs compared to traditional object detection methods that only learn predetermined classes (referred to as closed-set OD, or COD, in this paper)? What specific advantages do OVD methods offer, which are characterized by similarity calculations in the feature space enabling open-set recognition, the introduction of knowledge from large models (like BERT [4] or CLIP [32]), and the increased volume and variety of training data they enable?

To answer these questions, it is essential to understand the difficulty of describing object classes in text. In this paper, we propose a method to quantify the "text-describability" of OD datasets based on the zero-shot image classification accuracy of target object classes using CLIP. Using this method, we categorize various OD datasets by their text-describability; see Fig. 2. We then experimentally evaluate the performance of OVD and COD methods in FSOD across the introduced dataset categories.

The results of our experiments show that while OVD significantly outperforms COD under few-shot conditions for easily text-describable classes as expected, there is little difference between the two for classes that are hard to describe in text. Moreover, while OVD can learn from more diverse data, its utility is significant for easily describable classes but can be counterproductive for harder-to-describe classes. These findings are expected to provide some guidance to practitioners amidst the recent advances in various OD methods.

2 Related Works

2.1 Open-vocabulary Object Detection

Open-vocabulary object detection (OVD) is an emerging framework for object detection [48, 8, 51, 19, 24, 47, 52, 15, 42] that has seen significant progress in recent years. Unlike traditional methods (i.e., closed-set object detection (COD)), which can only identify predefined object categories [33, 1, 3], OVD allows the detection of objects not seen during training. This is achieved using linguistic knowledge from large models such as BERT [4] and CLIP [32]. To facilitate this capability, existing methods establish a shared feature space between vision and language modalities. They achieve this either by distilling outputs from text encoders [48, 8] or by applying text embeddings from pre-trained vision-language models (VLMs) to the classification weights for each category [52, 29, 15, 42].

2.2 Few-shot Object Detection

As it is often difficult to acquire large volumes of training data for object detection [21, 34, 9, 16], training a detector with only a few examples of target objects, known as few-shot object detection (FSOD) [13, 46, 40, 39, 31, 36], has garnered considerable attention. Existing methods for FSOD can be categorized into two approaches: meta-learning [13, 46, 40, 43] and finetuning [39, 41, 31, 7, 36]. The former approach originally attempts to acquire a "meta-skill" to detect new object classes from only a few samples through the learning of base classes. The latter approach simply involves pre-training on base classes and subsequently training on novel classes, expecting the usual benefits of transfer learning. Recent studies have reported that the finetuning-based approach outperforms the meta-learning-based despite its simplicity [39, 36, 31]. Additionally, Wang *et al.* reported that freezing model parameters except for the final task-specific heads yielded improvements [39]. Sun *et al.* improved this frozen-based approach by employing cosine similarity as classification scores and further added contrastive loss for a RoI head [36].

FSOD has primarily been studied within the framework of COD. However, in the research of OVD, it has become a norm to report the FSOD performance of OVD methods, in addition to their primary application in zero-shot scenarios [19, 24]. In this context, utilizing both textual information and few-shot labeled image examples is expected to improve performance compared to using either one alone. The above insight gained from FSOD in COD seems also applicable to FSOD in OVD. In fact, existing research has shown that finetuning with few-shot examples (where all models, including the text encoder, are subject to training) has become the standard method.

2.3 Recent FSOD Benchmarks

Existing FSOD has historically repurposed popular datasets like VOC [6] and COCO [21] as its benchmarks [13, 40, 46, 39, 31, 36], dividing them into disjoint two splits: base categories and novel categories. Specifically, PASCAL VOC is partitioned into 15 base and 5 novel categories, while COCO is divided into 60 base and 20 novel categories. Whereas these are well-maintained and useful benchmarks, the base and novel categories are sampled from the same dataset, which may be inadequate for evaluating model behaviors in real-world applications with varied target domains. To explore FSOD effectiveness in more diverse scenarios, recent studies have developed Cross-Domain FSOD (CD-FSOD), assessing performance across multiple image domains [17, 44][1]. Lee *et al.* [17] and Xiong *et al.* [44] compiled 10 and 3 datasets from different image domains, respectively, evaluating state-of-the-art FSOD methods. They reported traditional FSOD approaches [39, 36, 31] underperformed in the domains distinct from their base category training, highlighting the importance of diverse domain benchmarks. Their studies provided detailed evaluations using various detectors, but OVD were not investigated.

3 Exploring Best Practice for Few-shot Object Detection

3.1 Closed-set and Open-vocabulary Object Detection

The conventional approach to object detection, referred to as closed-set object detection (COD), operates in the setting where detectors are trained to identify only predefined object categories present in the training data [33, 38, 1, 3, 49, 20]. Figure 1(a) illustrates the model architecture for COD, which features a trainable layer as the final classification head, with dimensions corresponding to the number of target categories.

In contrast, open-vocabulary object detection (OVD) [48, 8, 52, 19, 29, 47, 42] operates in an open-set setting, leveraging a text encoder, usually derived from pre-trained large models such as BERT [4] or CLIP [32]. Figure 1(b) depicts the general architecture of OVD methods. OVD is characterized by the similarity calculation at the classification head, where text and image features are compared,

[1] While Lee *et al.* [17] introduced a similar concept and called it as Multi-domain Few-shot Object Detection (MoFSOD), we consider it identical to CD-FSOD.

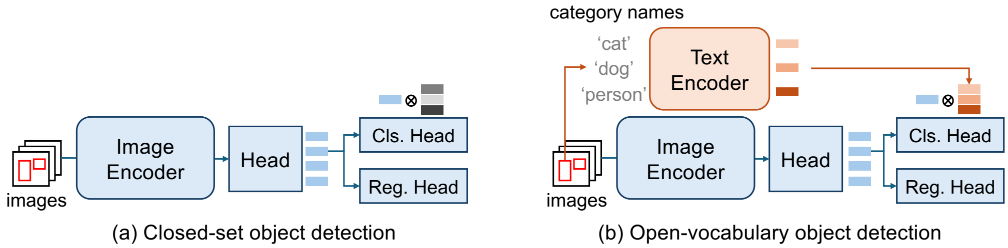

Figure 1: An overview of model architectures for (a) closed-set object detection (COD) and (b) open-vocabulary object detection (OVD).

facilitating open-set recognition. This structure enables the incorporation of textual knowledge into detection and increases the volume and variety of training data, as the models can be trained with more general datasets, such as image-caption pairs [35, 30], rather than data specifically designed for object detection.

3.2 Limitations with Existing OVD Benchmarks

We are investigating which is more suitable for FSOD between COD and OVD, particularly in cases where object categories are difficult to describe in text and the only option is to present image examples—situations where OVD may not have a significant advantage. If there is an advantage, we expect it to stem from one or more of the three characteristics of OVD mentioned earlier. These questions are critical for practitioners tackling real-world FSOD problems, especially given the recent surge in OVD research.

It is important to note that existing research on OVD has already reported on the performance of FSOD [19, 24]. However, these studies do not include comparisons of COD and OVD under the same conditions. More importantly, there is an issue with how datasets for training and testing are selected in current OVD studies, which is crucial for addressing the questions above.

OVD is characterized by pre-training on web-scale data, such as by using BERT or CLIP. In such cases, preventing train-test leakage for common object categories frequently found on the web is extremely difficult. This means that the object categories for which zero-shot/few-shot performance is being tested may have already been pre-trained. As a result, existing OVD research often does not avoid leakage and takes the stance that if the "dataset" is different—even if the same object class is being trained—it meets the zero-shot/few-shot requirements. Although this may seem counterintuitive, it is acceptable (or even advantageous) if the goal is to deploy detectors in scenarios with similar image domains and object categories as the training data; the aim is to create a detector that can identify any object as long as it is named.

However, we are focused on detecting object classes that are hard to describe and are necessarily rare on the web, either because the images themselves are rare or because they are not linked to useful text information. This means there is little to no leakage between train and test. Consequently, the few-shot performance for easily describable object categories reported in existing research is likely not useful for predicting the performance of the same detectors under our conditions of interest—where object categories are difficult to describe and there is no leakage between train and test. In other words, we cannot answer the aforementioned questions with the results of existing research.

3.3 Categorizing Datasets with Their Text Describability

To address the aforementioned limitations, it is essential to assess how easily the object classes in an individual object detection dataset[2] can be described by text. Only then can we explore the relationship between detector performance and the text-describability of the object classes. For

[2]To be precise, it is more about the tasks, i.e., the target object class list. For clarity, we refer to them as datasets here.

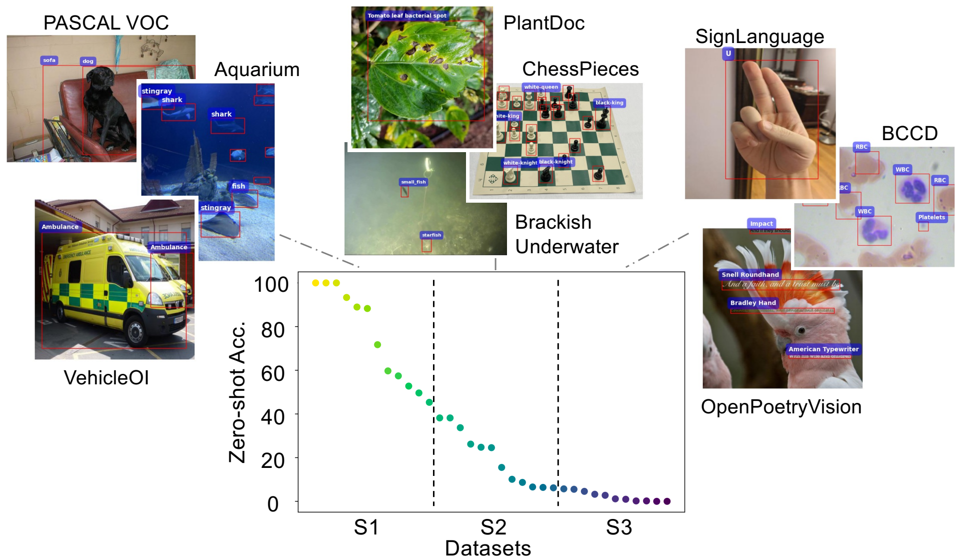

Figure 2: Datasets (35 in total from ODinW [18]) sorted by our metric for the difficulty of describing object classes in text. The datasets are categorized and ranked from S1 to S3, indicating decreasing text-describability.

example, we can experimentally determine which OVD or COD methods perform better on datasets that are challenging to describe in text.

How can we measure the text-describability of a single dataset? We propose using the zero-shot performance of CLIP as a "proxy indicator." This method involves preparing a collection of datasets $\{D_i\}_{i=1\dots n}$ and calculating the zero-shot classification accuracy for each dataset D_i in a comparable manner, thereby relativizing its text-describability.

Specifically, for the image input to CLIP, we use the image regions specified by the ground truth bounding boxes for each object class provided by the respective datasets. For text input, we first extract the list C_i of object class names from each dataset D_i, create their union $\cap C_i$, consolidate duplicates, and compile a class list C spanning all datasets. We then use prompts (e.g., "an image of {class name}") based on these classes as text inputs. For each dataset D_i, we perform classification on the common class set C using CLIP. The average classification accuracy a_i on the dataset-specific classes C_i (treating classifications into classes in $C \setminus C_i$ as errors) is used as the verbalizability indicator for D_i. Considering a classification problem on the common class set C aims to provide a comparable indicator even among datasets with different class counts.

In our experiments, we use ODinW (Object Detection in the Wild) [18] for dataset collection, which is a standard approach in recent OVD research [19, 24][3]. This collection includes 35 diverse datasets selected from the 100 available in Roboflow [2], each of which simulates a distinct real-world application of object detection.

Figure 2 shows how the 35 datasets are sorted using the proposed CLIP-based measure. For statistical evaluation of the detectors' performance, we divided the 35 datasets into three splits (12/12/11 each), labeled S1, S2, and S3, as detailed in the supplementary material. The datasets in S1, S2, and S3 exhibit decreasing CLIP performance, indicating they become less text-describable. As shown in Fig. 2, Split S1 includes datasets with common objects, such as the 20 categories of PASCAL VOC [6] and common vehicle categories in Open Images [16]. Split S2 comprises datasets with lower CLIP performance, such as aquatic life in underwater images and fine-grained plant diseases. Split S3

[3]Existing OVD research typically selects 13 out of 35 datasets and uses the average detection accuracy on these to compare methods. Most of these datasets belong to S1 and S2 in our classification, indicating they are easily verbalizable and do not effectively measure performance on less verbalizable datasets.

contains datasets like blood cell detection in medical images and sign language detection represented by alphabetical strings.

Remark It should be noted that CLIP's zero-shot performance does not directly correspond to the difficulty of verbalizing target objects. The significant variation in CLIP's zero-shot accuracy across different image classification datasets, as reported in the original paper [32], is likely due to whether the target object classes are included in CLIP's training data. In other words, CLIP's zero-shot performance depends on the abundance of image and class name text pairs in its training data.

CLIP's training data is widely collected from the web. When data for a particular object class is scarce, there can be two reasons: either the object is difficult to describe, making image-text pairs less likely to exist, or the images themselves are rare due to their specialized domain. Thus, CLIP's performance indicators may combine the difficulty of verbalization and the rarity of images, resulting in only a partial correlation with verbalizability.

However, considering our objective, this might be acceptable. We are interested in how OVD methods perform on data types they have not pre-trained on. Since CLIP's training data is broadly sourced from the web, the training data for OVD should be similar to some extent. Therefore, despite the aforementioned issues, we believe that linking CLIP's performance to the evaluation of OVD methods' performance is useful. Further analyses will be left for future study.

4 Experiments

To answer the above questions, we experimentally evaluate several representative OVD and COD methods in the standard few-shot setting. To ensure the reproducibility of our results, we will make all the code used in our experiments publicly available; see the supplementary material.

4.1 Compared Methods

Base Detectors We consider four state-of-the-art object detectors: two designed for closed-set object detection (COD)—Dynamic Head (DyHead) [3] and Faster RCNN [33], and two for open-vocabulary object detection (OVD)—GLIP(A) [19] and F-ViT [42]. DyHead [3] and Faster RCNN [33] are simple yet effective methods for COD, representing one-stage and two-stage detectors, respectively. We use Swin-T [27] with Feature Pyramid Network (FPN) [22] as their backbones.

GLIP(A) [19] is an open-vocabulary detector based on DyHead. It leverages BERT [4] as a pre-trained text encoder, to employ its text embeddings as the classification head of the detector. Following the original paper [19], we utilize Swin-T with FPN as the image encoder. In Sec. 4.3.3, we additionally evaluate GLIP, built on the GLIP(A) architecture but with two modifications over GLIP(A). 1) GLIP is pre-trained on a more extensive data that includes resources for phrase grounding (GoldG [12]) and image-caption pairs (CC [35] and SBU [30]). 2) GLIP incorporates deep fusion modules to enhance the integration of image and text information through cross-attention. These enhancements expectedly expand the vocabulary of visual concepts and allow the model to learn visual features more effectively conditioned on text inputs, both leading to improved OVD performance.

F-ViT [42] is an open-vocabulary detector based on Faster RCNN, using frozen CLIP [32, 5] both for the image and text encoders. Before being frozen, the image encoder employs contrastive learning to align dense features of local regions with global features of corresponding crop images. This enables tailored region-level representations for object detection tasks, improving the use of pre-trained CLIP. Following the original paper [42], we use EVA-CLIP [37] for the image and text encoders.

Methods for FSOD Finetuning Fully finetuning all trainable layers (Full-FT) serves as a baseline in many FSOD studies [46, 39, 31, 36]. Additionally, we evaluate two state-of-the-art finetuning approaches for FSOD: TFA [39] and FSCE [36]. TFA (Two-stage Fine-tuning Approach) [39] initially trains all parameters on pre-training phase as usual. Subsequently, only the last prediction heads (i.e., the last layers for classification, regression, centerness, and a projection for text embeddings) are finetuned with few training samples, while the remaining parameters are kept frozen. FSCE (Few-Shot object detection via Contrastive proposals Encoding) [36] builds upon a frozen-based approach similar to TFA. It enhances TFA by 1) unfreezing Region Proposal Network (RPN) and RoI head, 2) increasing the number of proposals in RPN passed to RoI head, 3) using cosine similarity as classification scores, and 4) adding contrastive proposal encoding loss to its prediction head. We apply FSCE only to Faster RCNN and F-ViT, considering that it is tailored for two-stage detectors as it adjust the number of RPN proposals.

4.2 Datasets and Evaluation Protocols

Object Detection Pre-training Unless stated otherwise, we utilize Object365-V1 (O365) [34], which comprises 0.61M images across 365 general object categories, as the pre-training dataset for all the detectors[4]. For GLIP(A) and GLIP, we use their publicly available pre-trained weights from the official repository[5]. Note that this pre-training process is distinctly separate from backbone-level training performed in CLIP [32], BERT [4], etc.

Evaluation of FSOD Performance As previously mentioned, we use the ODinW dataset [18], which consists of 35 individual object detection (OD) datasets, to evaluate the FSOD performance of the above OD methods; see Sec. 3.3 for details of ODinW. We report the average precision (AP) for each method over the intersection over union (IoU) range [0.50:0.95], averaged across datasets within each of the three splits—S1, S2, and S3—each characterized by different levels of text-describability.

For the few-shot configuration, we follow a sampling method employed in previous studies [19, 13]. Specifically, in K-shot settings, we randomly sample the target dataset to ensure that there are at least K images containing one or more ground truth bounding boxes for each category. We consider $K = [1, 3, 5, 10]$ settings. In all experiments, we repeat this sampling process five times using different random seeds and report the averaged performance.

4.3 Results

4.3.1 Comparison of COD and OVD Methods

Table 1 shows the performance of the compared four OD methods on the proposed three splits of ODinW, each with varying numbers K of shots. All methods employ the full-FT approach for FSOD.

It is observed that OVD methods (highlighted in the table) significantly outperform COD methods in the S1 and S2 splits. This is consistent for both one-stage methods (i.e., DyHead and GLIP(A)) and two-stage methods (i.e., Faster RCNN and F-ViT). This result is expected, as OVD methods are designed to detect objects described in text in a zero-shot setting, a capability that also benefits the few-shot setting. Although the performance gap between OVD and COD narrows as K increases, OVD methods consistently show superior performance in S1 and S2 with $K = 10$.

Another observation is that the performance gap between OVD and COD methods narrows in the S3 split. Figure 3 illustrates the AP ratios of an OVD method compared to its counterpart COD method, highlighting this trend. Specifically, it shows that for S3, GLIP(A)'s performance relative to DyHead's drops to around 1.0, indicating nearly equivalent performance; their APs differ by only about 1.0 AP with $K \geq 3$ (e.g., 39.7 vs. 39.2 at $K = 3$).

Moreover, Faster RCNN clearly outperforms its counterpart, F-ViT, with $K = 10$ in the S2 split and with $K \geq 3$ in the S3 split. Recall that the datasets in S3 are characterized by low text-describability, such as sign language detection and OCR tasks to identify font names. On these datasets, the superiority of the OVD methods seen in S1 and S2 diminishes. In fact, the COD methods perform even better by a noticeable margin.

[4] GLIP [19] reported the number of training images for O365 as 0.66M, but the provided dataset links have expired and cannot be verified. We will use a † symbol to indicate this in the results below.

[5] https://github.com/microsoft/GLIP

Table 1: Few-shot OD performance of COD (closed-set object detection) and OVD (open-vocabulary object detection) methods on the S1, S2, and S3 splits of the 35 ODinW datasets with different numbers K of shots. The values represent the average precision, averaged over the datasets within each split. OVD methods are shaded in gray; IE and TE represent image encoder and text encoder, respectively.

Method	Backbone (#param.)		$K=1$			$K=3$		
	IE	TE	S1	S2	S3	S1	S2	S3
DyHead	Swin-T(28M)	-	29.0 ±0.8	22.2 ±0.6	23.8 ±1.1	39.2 ±1.6	33.9 ±1.4	39.7 ±0.8
GLIP(A)	Swin-T(28M)	BERT(110M)	37.4 ±1.7	28.5 ±0.8	25.6 ±1.3	44.6 ±0.7	37.1 ±0.7	39.2 ±0.5
Faster RCNN	Swin-T(28M)	-	21.7 ±2.7	19.8 ±1.1	21.9 ±1.1	36.2 ±1.6	31.4 ±1.2	38.1 ±0.8
F-ViT	CLIP-ViT-B/16(86M)	CLIP(63M)	40.1 ±1.1	24.6 ±0.9	22.9 ±1.3	45.5 ±2.5	32.9 ±0.8	32.0 ±0.9
Method	Backbone (#param.)		$K=5$			$K=10$		
	IE	TE	S1	S2	S3	S1	S2	S3
DyHead	Swin-T(28M)	-	42.5 ±1.6	36.3 ±1.1	42.9 ±1.7	48.1 ±1.2	41.2 ±1.5	48.7 ±1.1
GLIP(A)	Swin-T(28M)	BERT(110M)	49.0 ±0.5	40.2 ±0.7	43.6 ±0.7	52.3 ±1.1	44.5 ±1.0	49.9 ±0.7
Faster RCNN	Swin-T(28M)	-	40.1 ±2.0	36.0 ±0.6	42.8 ±0.5	45.7 ±1.0	39.9 ±0.9	48.9 ±1.7
F-ViT	CLIP-ViT-B/16(86M)	CLIP(63M)	47.7 ±2.6	36.6 ±1.4	35.2 ±1.3	49.6 ±1.5	40.2 ±0.9	38.7 ±1.2

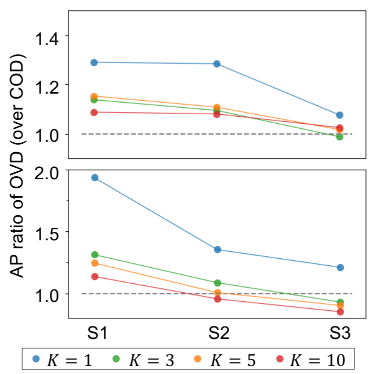

Figure 3: AP ratio of OVD/COD. DyHead vs. GLIP(A) (top) and Faster RCNN vs. F-ViT (bottom).

Table 2: Detection accuracy of state-of-the-art finetuning approaches for FSOD. Results on a $K=3$ are shown. OVD methods are shaded in gray.

Method	Finetuning	S1	S2	S3
DyHead	Full-FT	39.2 ±1.6	33.9 ±1.4	39.7 ±0.8
	TFA [39]	36.3 ±0.7	23.1 ±1.0	16.0 ±0.7
GLIP(A)	Full-FT	44.6 ±0.7	37.1 ±0.7	39.2 ±0.5
	TFA [39]	34.5 ±0.5	19.2 ±0.6	10.7 ±0.5
Faster RCNN	Full-FT	36.2 ±1.6	31.4 ±1.2	38.1 ±0.8
	TFA [39]	29.9 ±1.3	22.0 ±1.1	15.4 ±0.9
	FSCE [36]	36.7 ±0.6	28.1 ±2.4	30.5 ±1.9
F-ViT	Full-FT	45.5 ±2.5	32.9 ±0.8	32.0 ±0.9
	TFA [39]	23.6 ±0.6	8.8 ±0.2	5.0 ±0.1
	FSCE [36]	44.7 ±1.1	32.4 ±1.2	34.3 ±0.4

4.3.2 Impact of Few-shot Finetuning Methods

We next examine the impact of the fine-tuning methods employed for few-shot learning. Table 2 presents the results for $K=3$ using the same four OD methods with different FSOD fine-tuning approaches. It is observed that TFA [39] performs the worst regardless of OVD or COD. Notably, its performance gap compared to Full-FT (i.e., fine-tuning all trainable parameters) increases progressively from S1 to S3.

FSCE [36], applicable to both Faster RCNN and F-ViT, exhibits similar behavior to TFA, except that F-ViT performs better on S3 with FSCE than with Full-FT. These findings suggest that TFA and FSCE, both recent FSOD fine-tuning methods, do not outperform the standard Full-FT. This holds true regardless of whether the method is COD or OOD and the level of text-describability. This result extends the findings of Lee et al.'s study [17] from COD to OVD, showing that fine-tuning only high-layer parameters improves FSOD performance only when the domain gap between train and test datasets is minimal; otherwise, it negatively impacts performance, and fine-tuning all parameters yields the best results.

4.3.3 Impact of Pre-training Data

In FSOD, the detector is initially trained on OD tasks, typically using a large OD dataset and then finetuned with few-shot samples for the target OD task. We examined the impact of this pretraining stage on FSOD with different levels of text-describability.

Table 3: Detection accuracy across varying amounts of pre-training data. All models are finetuned with Full-FT under a $K = 3$ setting. G and C represents grounding datasets (GoldG [12]) and image-caption pairs (CC [35] and SBU [30]), respectively. OVD methods are shaded in gray. See Sec. 4.2 for the † indicator.

Method	Backbone (#param.)		Pre-training	#Images	S1	S2	S3
	IE	TE					
DyHead	Swin-T(28M)	-	COCO+O365	2K (1%)	29.6 ± 1.8	27.0 ± 0.6	31.1 ± 0.5
				20K (10%)	35.1 ± 1.0	29.8 ± 1.3	33.7 ± 0.8
				0.10M (50%)	40.1 ± 1.4	33.6 ± 0.8	36.6 ± 0.4
				0.20M (100%)	40.8 ± 1.0	34.0 ± 1.2	37.9 ± 0.3
			O365	0.61M	39.2 ± 1.6	33.9 ± 1.4	39.7 ± 0.8
GLIP(A)	Swin-T(28M)	BERT(110M)	O365†	0.66M	44.6 ± 0.7	37.1 ± 0.7	39.2 ± 0.5
GLIP	Swin-T(28M)	BERT(110M)	O365†+G+C	5.46M	50.4 ± 0.4	39.6 ± 1.2	34.9 ± 0.6

Specifically, we used DyHead from COD and studied the effects of the amount of pre-training OD data. We randomly selected 0.10M images from the COCO dataset (0.12M in total) and 0.61M images from the Objects365 (O365) dataset, combining them to create a 0.20M image dataset. We then created scaled subsets by extracting $x\%$ of images from this combined dataset, maintaining a consistent 1:1 image ratio between COCO and O365. DyHead was trained on these subsets, followed by few-shot adaptation on the S1, S2, and S3 subsets.

The results, shown in Table 3, indicate that generally, more pre-training data leads to better FSOD performance. However, a closer examination reveals that the effect is more pronounced for S1 and less so for S3. This likely occurs because the overlap (in terms of object categories and image domains) with the pre-training data decreases in the order of S1, S2, and S3. When targeting S3, although more pre-training data is beneficial, the performance gains diminish compared to S1.

OVD has an advantage over COD in that it can utilize more general image-text pair data, not limited to OD-specific data. We have observed that OVD significantly outperforms COD in S1 and S2 (rows 5 and 6 of the table, copied from Table 1). This performance gap is expected to widen with the inclusion of non-OD data. However, can this advantage be observed in S3 as well?

To answer this question, we expanded the training data for GLIP(A) under the same conditions for FSOD, resulting in a model referred to as GLIP; see Sec. 4.1 for details. The results, shown in row 7 of Table 3, indicate improved accuracy in S1 and S2. Since this method is exclusively applicable to OVD, OVD demonstrates a clear advantage over COD here. However, intriguingly, Table 3 shows that for S3, GLIP performs worse than GLIP(A) and even falls behind DyHead, a COD method. This suggests that it is safer to use COD for datasets with characteristics like S3. This further supports the above conclusion that FSOD on S3 shows no significant difference between OVD and COD, thus not justifying the extra cost of OVD.

5 Summary and Conclusion

In this paper, we have addressed the problem of few-shot object detection (FSOD), focusing on the comparison between open-vocabulary object detection (OVD) and closed-set object detection (COD). We first proposed a method to quantify the difficulty of describing target object classes in text using zero-shot image classification accuracy with CLIP. This has enabled us to empirically evaluate COD and OVD methods under equal conditions on various datasets with varying levels of text-describability. Our results provide several key findings. Firstly, for datasets with high text-describability, OVD significantly outperforms COD, as expected. However, when the classes are difficult to describe in text, the superiority of OVD diminishes. Additionally, pre-training on a larger amount of data, which is uniquely beneficial for OVD, can be counterproductive for datasets with low text-describability. These results suggest that for FSOD on datasets where object classes are hard to describe in text, COD methods are recommended over OVD methods. This guidance is valuable for practitioners who are navigating the recent advancements in OVD methods and seeking to optimize their FSOD approaches for specific datasets.

References

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-End Object Detection with Transformers. In *Proc. ECCV*, 2020.

[2] F. Ciaglia, F. S. Zuppichini, P. Guerrie, M. McQuade, and J. Solawetz. Roboflow 100: A Rich, Multi-Domain Object Detection Benchmark, 2022.

[3] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang. Dynamic Head: Unifying Object Detection Heads With Attentions. In *Proc. CVPR*, 2021.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proc. ACL*, 2019.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. ICLR*, 2021.

[6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010.

[7] B.-B. Gao, X. Chen, Z. Huang, C. Nie, J. Liu, J. Lai, G. Jiang, X. Wang, and C. Wang. Decoupling Classifier for Boosting Few-shot Object Detection and Instance Segmentation. In *Proc. NeurIPS*, 2022.

[8] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *Proc. ICLR*, 2022.

[9] A. Gupta, P. Dollar, and R. Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *Proc. CVPR*, 2019.

[10] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proc. ICML*, 2021.

[11] Q. Jiang, F. Li, Z. Zeng, T. Ren, S. Liu, and L. Zhang. T-Rex2: Towards Generic Object Detection via Text-Visual Prompt Synergy. *arXiv*, 2403.14610, 2024.

[12] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. In *Proc. ICCV*, 2021.

[13] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. Few-shot Object Detection via Feature Reweighting. In *Proc. ICCV*, 2019.

[14] P. Kaul, W. Xie, and A. Zisserman. Multi-Modal Classifiers for Open-Vocabulary Object Detection. In *Proc. ICML*, 2023.

[15] W. Kuo, Y. Cui, X. Gu, A. Piergiovanni, and A. Angelova. F-VLM: Open-Vocabulary Object Detection upon Frozen Vision and Language Models. In *Proc. ICLR*, 2023.

[16] A. Kuznetsova, H. Rom, N. Alldrin, J. R. R. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv*, 1811.00982, 2018.

[17] K. Lee, H. Yang, S. Chakraborty, Z. Cai, G. Swaminathan, A. Ravichandran, and O. Dabeer. Rethinking few-shot object detection on a multi-domain benchmark. In *Proc. ECCV*, 2022.

[18] C. Li*, H. Liu*, L. H. Li, P. Zhang, J. Aneja, J. Yang, P. Jin, Y. J. Lee, H. Hu, Z. Liu, et al. ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models. In *Proc. NeurIPS*, 2022.

[19] L. H. Li*, P. Zhang*, H. Zhang*, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao. Grounded Language-Image Pre-training. In *Proc. CVPR*, 2022.

[20] Y. Li, H. Mao, R. B. Girshick, and K. He. Exploring Plain Vision Transformer Backbones for Object Detection. In *Proc. ECCV*, 2022.

[21] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. ECCV*, 2014.

[22] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *Proc. CVPR*, 2017.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. In *Proc. ICCV*, 2017.

[24] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv*, 2303.05499, 2023.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proc. ECCV*, 2016.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *1907.11692*, 2019.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proc. ICCV*, 2021.

[28] A. Madan, N. Peri, S. Kong, and D. Ramanan. Revisiting Few-Shot Object Detection with Vision-Language Models. *arXiv*, 2312.14494, 2022.

[29] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple Open-Vocabulary Object Detection with Vision Transformers. In *Proc. ECCV*, 2022.

[30] V. Ordonez, G. Kulkarni, and T. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Proc. NeurIPS*, 2011.

[31] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang. DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection. In *Proc. ICCV*, 2021.

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. ICML*, 2021.

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *Proc. NeurIPS*, 2015.

[34] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, J. Li, X. Zhang, and J. Sun. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In *Proc. ICCV*, 2019.

[35] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proc. ACL*, 2018.

[36] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang. FSCE: Few-Shot Object Detection via Contrastive Proposal Encoding. In *Proc. CVPR*, 2021.

[37] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao. EVA-CLIP: Improved Training Techniques for CLIP at Scale. *arXiv preprint arXiv:2303.15389*, 2023.

[38] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: Fully Convolutional One-stage Object Detection. In *Proc. ICCV*, 2019.

[39] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu. Frustratingly Simple Few-Shot Object Detection. In *Proc. ICML*, 2020.

[40] Y.-X. Wang, D. Ramanan, and M. Hebert. Meta-Learning to Detect Rare Objects. In *Proc. ICCV*, 2019.

[41] J. Wu, S. Liu, D. Huang, and Y. Wang. Multi-Scale Positive Sample Refinement for Few-Shot Object Detection. In *Proc. ECCV*, 2020.

[42] S. Wu, W. Zhang, L. Xu, S. Jin, X. Li, W. Liu, and C. C. Loy. CLIPSelf: Vision transformer distills itself for open-vocabulary dense prediction. In *Proc. ICLR*, 2024.

[43] Y. Xiao and R. Marlet. Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild. In *Proc. ECCV*, 2020.

[44] W. Xiong. CD-FSOD: A Benchmark for Cross-domain Few-shot Object Detection. In *Proc. ICASSP*, 2023.

[45] Y. Xu, M. Zhang, C. Fu, P. Chen, X. Yang, K. Li, and C. Xu. Multi-modal Queried Object Detection in the Wild. In *Proc. NeurIPS*, 2023.

[46] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin. Meta R-CNN : Towards General Solver for Instance-level Low-shot Learning. In *Proc. ICCV*, 2019.

[47] L. Yao, J. Han, X. Liang, D. Xu, W. Zhang, Z. Li, and H. Xu. DetCLIPv2: Scalable Open-Vocabulary Object Detection Pre-Training via Word-Region Alignment. In *Proc. CVPR*, 2023.

[48] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. Open-Vocabulary Object Detection Using Captions. In *Proc. CVPR*, 2021.

[49] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, and H.-Y. Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *Proc. ICLR*, 2023.

[50] X. Zhang, Y. Wang, and A. Boularias. Detect Everything with Few Examples, 2024.

[51] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li, et al. RegionClip: Region-based Language-Image Pretraining. In *Proc. CVPR*, 2022.

[52] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting Twenty-thousand Classes using Image-level Supervision. In *Proc. ECCV*, 2022.

[53] X. Zhou, V. Koltun, and P. Krähenbühl. Probabilistic two-stage detection. *arXiv*, 2103.07461, 2021.

[54] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *Proc. ICLR*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 3.3.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] While we have not extensively explored this aspect, we are confident that our work does not possess any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section A in the supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 in the main paper and Section B in the supplementary materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section B in the supplementary materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]