Few-Shot Object Detection with Foundation Models

Guangxing Han
Columbia University
guangxinghan@gmail.com

Ser-Nam Lim
University of Central Florida
sernam@ucf.edu

Abstract

Few-shot object detection (FSOD) aims to detect objects with only a few training examples. Visual feature extraction and query-support similarity learning are the two critical components. Existing works are usually developed based on ImageNet pre-trained vision backbones and design sophisticated metric-learning networks for few-shot learning, but still have inferior accuracy. In this work, we study few-shot object detection using modern foundation models. First, vision-only contrastive pre-trained DINOv2 model is used for the vision backbone, which shows strong transferable performance without tuning the parameters. Second, Large Language Model (LLM) is employed for contextualized few-shot learning with the input of all classes and query image proposals. Language instructions are carefully designed to prompt the LLM to classify each proposal in context. The contextual information include proposal-proposal relations, proposal-class relations, and class-class relations, which can largely promote few-shot learning. We comprehensively evaluate the proposed model (FM-FSOD) in multiple FSOD benchmarks, achieving state-of-the-arts performance.

1. Introduction

Learning to recognize and localize unseen classes without large-scale of training is crucial to achieve human-level vision intelligence. Few-shot object detection (FSOD) [3, 17, 22, 66, 69], aiming to detect novel objects with only a few visual training examples, serves as a valuable benchmark for these endeavors. However, the development of FSOD models has been slow recently and the performance are far worse than data-abundant models. In contrast, some other open-set object detection settings, especially open-vocabulary object detection [13, 27, 33, 57, 65] has made significant progress, evolving from traditional strictly defined zero-shot models to modern open-vocabulary models capable of detecting any object defined by natural language description. Recently, these models have achieved performance comparable to data-abundant models.

We argue that the key to the success of open-vocabulary object detection is the intensive use of pre-trained large-scale vision-language models, like CLIP [40], which can learn aligned feature representations for both the vision and language modalities. While the majority of previous FSOD works utilize ImageNet pre-trained models, most of the efforts are dedicated to the development of better few-shot fine-tuning techniques (e.g., contrastive learning [35, 43], feature/data augmentation [60, 68]), and sophisticated human-designed metric-learning networks (e.g., complicated query-support feature fusion networks [3, 5, 12, 14, 16, 17, 20, 69] with alignment/GCN/Transformers). Although some recent works, like MM-FSOD [15] and DE-ViT [69] propose to use modern foundation models, like CLIP [40] or DINOv2 [37] in their models, they do not provide comprehensive evaluations or analyses across different foundation models, and more importantly, both of them have heavily designed few-shot classification networks. The insufficient utilization of modern foundation models hinders further improvement for FSOD models.

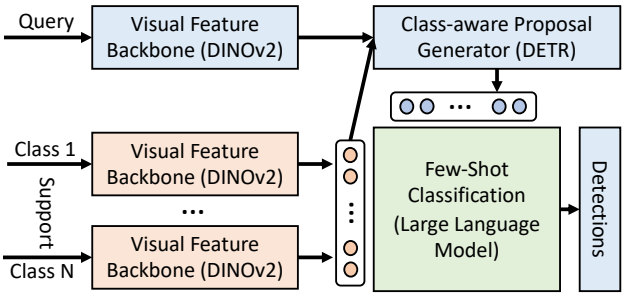

Figure 1. Overview of the proposed model. The diagram shows the high-level framework of our model for FSOD. The visual feature backbone and query-support few-shot classification network are the two critical components for FSOD. In this work, we study FSOD with Foundation Models, and propose to use the frozen self-supervised DINOv2 as the visual backbone and leverage the strong in-context learning capability of Large Language Model for contextualized few-shot proposal classification. Our model achieves strong FSOD performance and reduces human effort to design sophisticated few-shot learning models.

In this work, we study FSOD using modern foundation

models for both visual feature extraction and few-shot proposal classification. As shown in Figure 1, the visual feature backbone and query-support few-shot classification network are the two critical components for FSOD. First, the pre-trained vision backbone should possess not only strong discriminative ability across various semantic concepts but also robust patch-level spatial localization ability, making it ideal for downstream localization-sensitive tasks. Based on this motivation, we comprehensively evaluate multiple pre-trained vision foundation models, including MAE [19], CLIP [40], SAM [24] and DINOv2 [37], and with different detection architectures, RCNN-based framework ViTDet [28] and Transformer-based framework Deformable DETR [71]. Our conclusion is that DINOv2, pre-trained with both image-level and patch-level self-supervised objectives, and equipped with Transformer-based detection framework achieves the best performance. Moreover, unlike some previous works [12, 17] that require updating the visual feature backbone during training and thus can only support very few classes in a single feed-forward pass, our model does not need to fine-tune the visual backbone. This enables contextualized few-shot learning with a larger set of classes.

Second, few-shot classification for object proposals[1] is another key component for FSOD. We propose to generate support-class-aware proposals by applying cross-attention between the query image features and class prototypes. But the proposals are still noisy. The key challenge of FSOD is the few-shot learning with noisy proposals. Previous works [12, 17, 51, 69] propose several methods for this problem, ranging from simple dot product to more sophisticated deep neural networks with the aim to improve the few-shot classification with noisy proposals. In this work, we propose to leverage the strong in-context learning capabilities of pre-trained Large Language Models (LLMs) [2, 9, 25, 46] for contextualized few-shot proposal classification in FSOD. Motivated by recent multimodal LLMs [1, 6, 32, 38, 48, 49, 62, 67], we carefully design language instructions to prompt the LLM to classify each proposal, and provide the mapping between the categories and their visual prototypes as part of the input instructions. Our model can automatically exploit various contextual information between proposals and classes through the LLM, including proposal-proposal relations, proposal-class relations, and class-class relations. The extracted context information can largely promote few-shot proposal classification from the same query image. As for model training, we fine-tune the LLM with meta-learning. In each training episode, we randomly sample some visual samples for each category to calculate the prototypes, which serves as strong data augmentations during model training. We evaluate our method, termed as FD-FSOD, on two widely used FSOD benchmarks and also conduct extensive ablation studies to demonstrate the effectiveness of our model.

Our contributions can be summarized as:

- We study few-shot object detection based on modern foundation models for both visual feature extraction and contextualized few-shot proposal classification.
- Fully-Transformer based detection framework together with DINOv2 backbone achieves strong generalization for both data-abundant classes and few-shot classes.
- The LLM with in-context language instructions can simplify the modeling of query-support few-shot classification network, and automatically learn rich contextual information to facilitate the few-shot learning.
- Our proposed model FD-FSOD achieves state-of-the-arts or strong performance on both the PASCAL VOC and MSCOCO FSOD benchmarks.

2. Related Works

2.1. Few-Shot Object Detection

Few-shot object detection (FSOD) aims to detect unseen novel objects using a few training examples (*a.k.a.*, support images). Besides the few-shot training data, we usually have another data-abundant base classes to assist the training which do not have any overlap with the novel objects. Existing works can be roughly categorised into the following two groups: (1) Fine-tuning-based methods [43, 51, 56, 68, 70]. They usually have two stages for training. Firstly, the object detectors is trained over base classes only. Then, the pre-trained detection models are fine-tuned over few-shot novel classes. During few-shot fine-tuning, some training strategies like re-sampling [51] and re-weighting [31] are utilized to train models with the unbalanced combination of many-shot base-classes dataset and few-shot novel-classes dataset. (2) Meta-learning-based methods [12, 14, 16, 17, 20, 22, 61]. Class-agnostic few-shot detection models [12, 14, 16, 17, 20, 22, 61] are learned over base classes, which can be generalized to novel classes without fine-tuning. The metric-learning-based methods have been demonstrated to be effective by learning a generalizable class-agnostic metric-space over base classes. These methods are usually based on a siamese network architecture and calculate the similarity between the query image regions and few-shot support images using metric-learning [23]. The most simplest method is using dot-product [51]. Subsequent works design more sophisticated deep neural networks to improve the accuracy of similarity learning, including multiple feature fusion networks [12, 59, 61], feature alignment [16], GCN [14], and non-local attention/Transformer [5, 8, 10, 17, 20, 50]). More recently, DE-VIT [69] does not use the original visual features, and uses the maps of similarities between the proposal features and a set of prototypes for classification.

[1] We use proposals here for simplicity, which can also be object queries in DETR-style framework.

2.2. Foundation Models

Foundation Models (FMs) are large-scale machine learning models trained over a vast amounts of data. After training, they can be adapted to a variety of downstream tasks. Recently, Foundation Models have made significant progress in Natural Language Processing (NLP), Computer Vision (CV) and Vision-Language Pre-training (VLP).

Large language models (LLMs) have gained significant attention in the field of NLP and Artificial General Intelligence (AGI) due to their impressive capabilities for language generation. LLMs have demonstrated strong emergent capabilities [54], including in-context learning [2], instruction following [53], and chain-of-thought reasoning [45, 55], which have revolutionized the field of NLP and AGI. Subsequent works [1, 32, 48] introduce other modalities, mostly vision information, into LLMs and thus build multimodal LLMs. In multimodal LLMs, a vision encoder (e.g., Frozen CLIP encoder in LLaVA [32]) is used to encode visual information into hidden representations. Then a trainable projection layer (e.g., Perceiver in [1], Q-Former in [26], or a linear/MLP layer in [32]) is followed to convert image features into the language embedding space. Finally, LLMs take all of the visual tokens and textual tokens in a sequence and make predictions according to the instruction. LLMs can be frozen in [1, 48], full fine-tuned in [32] or updated with LoRA [21]. Some recent work [6, 7, 38, 49, 62, 67] further introduce bounding box locations into the input instruction or the output of LLMs, enabling region-level fine-grained image comprehension for multimodal LLMs. But most of them only evaluate on simple grounding tasks like RefCOCO [64], and do not have satisfying performance on dense object detection tasks on MSCOCO and LVIS [6, 49, 58]. Different from these previous work, we do not use the LLM to predict bounding box location, but only prompt the LLM to classify each of the proposal in a fixed order with the visual lookup table in-context, which can largely ease the learning for the LLM.

On the other hand, large Vision Foundation Models have also made big progress to build stronger and generalizable vision generalist models. One approach is to employ self-supervised learning and scale the training process with a large dataset to learn a strong visual feature backbone [19, 29, 37, 63]. Another line of work [40, 44] utilizes vision-language paired training data to learn transferable visual and text representations through cross-modal contrastive learning. Some recent work (e.g., Painter [52]) reformulates various pixel-level vision tasks (including depth estimation, human keypoint detection, semantic segmentation and etc) into inpainting task, given a few examples of input and output image for a certain task. SAM [24] is a zero-shot segmentation model which can predict segmentation masks, given a query image and a prompt (e.g., box, points, text, or mask) specifying what to segment in an image. We focus on detection task, and a stronger vision backbone has a significant impact on downstream few-shot learning tasks.

3. The Proposed Approach

3.1. Task Definition

In few-shot object detection (FSOD), we have two sets of classes $C = C_{base} \cup C_{novel}$ and $C_{base} \cap C_{novel} = \emptyset$, where base classes C_{base} have plenty of visual training examples per class, and novel classes C_{novel} (*a.k.a.*, support classes) only have very few visual training examples per class (*a.k.a.*, support images). For K-shot (*e.g.*, $K = 1, 5, 10$) object detection, we have exactly K bounding box annotations for each novel class $c \in C_{novel}$ as the training data. The goal of FSOD is to use the few-shot visual examples to detect novel classes, with the assistance of data-abundant base-classes training data, and also keep strong performance on the base classes.

3.2. The Model Architecture

We propose to study FSOD with foundation models in this work. The idea is to make full use of the knowledge in the pre-trained large-scale vision/language foundation models for downstream few-shot learning tasks, and simplify the human efforts for model design.

As shown in Figure 2, our model mainly consists of the following three submodules: (1) *Visual Feature Extraction* to extract feature representations for both query images and few-shot support images, (2) *Proposal Generation* to generate support-class-aware object regions from the query image, and (3) *Few-Shot Proposal Classification* to classify each of the proposal given the mapping of the categories and their visual prototypes. The following details the architecture and model design for each component.

Visual Feature Extraction. In FSOD, we have a query image $I_q \in \mathbb{R}^{H_{I_q} * W_{I_q} * 3}$ and N-way K-shot support set $\mathcal{S} = \{\{I_s^{j,i}\}_{i=1}^{K}\}_{j=1}^{N}$ as inputs, and $I_s^{j,i} \in \mathbb{R}^{H_{I_s} * W_{I_s} * 3}$. For the query image, we use the Vision Transformers (ViTs) to extract the feature representation $f_q = \mathcal{F}(I_q)$ and keep all the local patch representations for the following object localization. Then for the support set, we similarly use the same ViTs to extract the feature representation $f_s^{j,i} = \mathcal{F}(I_s^{j,i})$ for each support image $I_s^{j,i}$. In practice, the support image is cropped around the target object with some image context pixels [12]. We use RoIAlign [18] to calculate the representation of the object given the bounding box annotation of the object $\overline{f}_s^{j,i} = RoIAlign(f_s^{j,i}, box_s^{j,i})$. Then the prototype for each class is the average of the K-shot support features $\{\hat{f}_s^j = \frac{\sum_{i=1}^{K} \overline{f}_s^{j,i}}{K}\}_{j=1}^{N}$.

In this work, we use the pre-trained frozen DINOv2 [37] as our feature backbone for the following two reasons. (1)

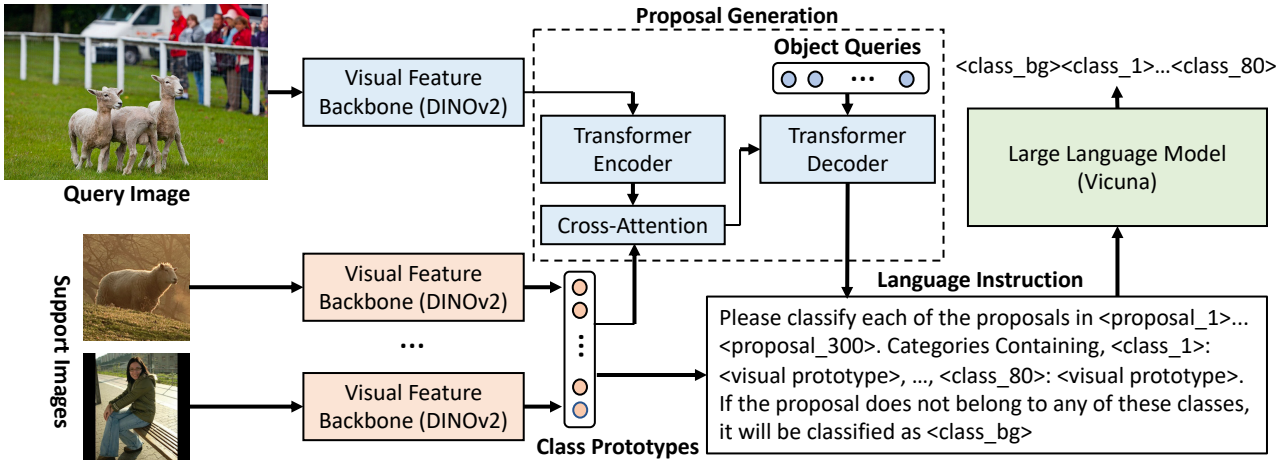

Figure 2. The overall architecture of our proposed model. Our model consists of three submodules: (1) *Visual Feature Extraction* to extract query image features and few-shot class prototypes, (2) *Proposal Generation* to generate support-class-aware object regions from the query image, and (3) *Few-Shot Proposal Classification* to classify each proposal. Our method make full use of modern Foundation Models for FSOD, which achieves strong performance without the need to design sophisticated few-shot learning modules.

DINOv2 is a vision-only self-supervised learning models, trained on a large scale of curated image dataset. Both global image-level and local patch-level self-supervised objectives are jointly used to train the feature backbone. The local patch loss can enforce the model to be localization-sensitive, which is friendly to downstream detection tasks. (2) The DINOv2 model is pre-trained over a large scale of image dataset. In order to the keep the original knowledge in DINOv2, we freeze the feature backbone during training. Our experiments show that fine-tuning some layers of DINOv2 does not improve the performance. Moreover, freezing the backbone allows us to pre-calculate the support features for each class, which enables in-context few-shot classification with a broader set of classes. (3) A potential concern of using DINOv2 is that the few-shot novel classes used for testing might have been seen during DINOv2 pre-training. We argue that the pre-training only learns image representation. How to efficiently transfer the foundation models to downstream tasks is still challenging, especially when the pre-training self-supervised learning tasks and downstream detection task have a large discrepancy.

Proposal Generation. After extracting visual features using DINOv2, we use the Transformer encoder-decoder architecture [4, 71] for proposal generation. Specifically, we first use multi-layer Transformer encoder over query image patch tokens extracted by DINOv2 in the above step. With the Transformer encoder module, each patch token features are enriched with global contextual information. Multi-scale deformable attention are used for fast convergence following [71].

In order to generate the support-class-aware proposals, we calculate cross-attention between the class prototypes $\{\hat{f}_s^j\}_{j=1}^N$ extracted from the DINOv2 backbone and the refined query image features from the Transformer encoder, generating support-class-aware query image features.

Then in the Transformer decoder, a sequence of randomly initialized object queries $Q = \{q_i\}_{i=1}^M$ together with the support-class-aware query image features are taken as inputs. Several self-attention and cross-attention layers are employed to refine the representations of object queries $\hat{Q} = \{\hat{q}_i\}_{i=1}^M$, gradually converging them to the corresponding objects. Bounding box locations of each object query $B = \{b_i\}_{i=1}^M, b_i = [x, y, w, h]$ are calculated using simple linear layers on the top of Transformer decoder. In this way, we can generate a small number ($M = 300$ in our model) of support-class-aware object queries (or proposals) for the following few-shot classification module.

Few-Shot Proposal Classification. After obtaining the proposals from the query images along with the prototype representation for each class, few-shot classification is another critical module to classify the proposals to be one of the support classes or the 'empty' class [4]. Previous methods [12, 14, 16, 17, 69] design sophisticated deep neural networks for similarity learning between the noisy proposals and support classes. In this work, we propose to leverage the strong in-context learning capability of LLMs for contextualized few-shot proposal classification, which can improve the accuracy of few-shot classification by introducing context information and simplify human efforts for designing complicated metric-learning networks.

Specifically, we first add several class tokens (e.g., from $<$*class_1*$>$ to $<$*class_80*$>$ in MSCOCO dataset), and

the background class token *<class_bg>* to the LLM tokenizer. Then we design the following language instructions for the LLM to perform classification for each of the proposals. *"Please classify each of the proposals in <proposal_1>...<proposal_300>. Categories Containing, <class_1>: <visual prototype>... <class_80>: <visual prototype>. If the proposal does not belong to any of these classes, it will be classified as <class_bg>."* Then, we replace the placeholder *<proposal_1>* with the corresponding proposal feature followed by a trainable projection layer to convert the dimension to that of the word embeddings in LLM, and replace *<visual prototype>* with the corresponding class prototype followed by another projection layer. The rest of the language instructions are tokenized by the LLM tokenizer with the newly-introduced class tokens. We use Vicuna [9] as our default language model, which is a decoder-only LLM, instruction-tuned from Llama [46]. The LLM takes the encoded features of the above instruction as inputs, and generate the *<class_id>* tokens for each of the proposal by implicitly looking up the category mapping table defined in the input instruction. Moreover, the output tokens of proposal classification maintain the same order as the proposals in the input instruction. After decoding the generated tokens by the LLM, we get the final detections by fusing the classification results of the LLM and the predictions in the proposal generation module.

By providing the aforementioned language instructions and prompting the LLM to classify each of the proposals, our method is simple by design. More importantly, our model takes the input of all proposals together with the category mapping table of all classes, which can automatically exploit the multiple relations among proposals and classes, including proposal-proposal relations, proposal-class relations, and class-class relations. The extracted context information can largely promote few-shot proposal classification from the same query image.

Recently, there are a large number of concurrent works [6, 7, 49, 62, 67] propose to incorporate the spatial localization ability into LLMs for region-level fine-grained image comprehension. Their methods performs well on simple grounding tasks like RefCOCO, but are struggling on more complicated dense detection tasks on MSCOCO and LVIS [6, 49, 58]. Different from these methods, we use the LLM in a different way for detection tasks, and only prompt the LLM to classify the proposals. This simplifies the problem of generating *unordered bounding boxes* (*classification tokens* + *spatial location tokens*) to only generating *ordered classification tokens* for proposal queries. Therefore our method largely eases the learning for the LLM.

3.3. The Training Framework

We have the following three steps for model training.

In the first step, we pre-train the proposal generation module, Deformable DETR [71] with the frozen DINOv2 backbone on base classes. We follow the original loss functions defined in DETR [4] by first finding the optimal bipartite matching between the predicted objects set and ground-truth objects set, and then optimizing the model towards this optimal assignment.

In the second step, we learn the whole model on base classes. The proposal generation module is initialized from the trained model in the first step. The LLM is initialized from Vicuna model. In order to obtain the ground-truth labels of the proposals for LLM training, we use the bipartite matching in DETR to assign labels to the proposals. Then, we can train our LLM end-to-end using the next-token prediction loss, calculated over the ground-truth proposal labels. In this step, the proposal generation module is also fine-tuned with the DETR loss.

In the third step, we fine-tune our model on novel classes. Similarly as the first and second step, we first fine-tune the proposal generation module with down-sampled base classes and novel classes. Then, we fine-tune the LLM using base classes and up-sampled novel classes following [36] because fine-tuning LLM needs more training data.

4. Experimental Results

4.1. Datasets

We evaluated our model on two widely used FSOD benchmarks, the MSCOCO [30] and PASCAL VOC dataset [11] following the evaluation protocol defined in [51].

PASCAL VOC. Following previous works in [22, 51], we have three random partitions of base and novel categories. In each partition, the twenty PASCAL VOC categories are split into fifteen base classes and five novel classes. We have the exact same few-shot images for model training/testing as [43, 51], and report AP50 results under shots 1, 2, 3, 5, and 10.

MSCOCO. We use the twenty PASCAL VOC categories as novel classes and the remaining sixty categories are base classes. We have the exact same few-shot images for model training/testing as [43, 51], and report the detection accuracy AP/AP50/AP75 under shots 1, 2, 3, 5, 10 and 30 following [14, 39, 51].

We report the full results on the two FSOD benchmarks in Section 4.3, and use the MSCOCO dataset under 10-shot for the ablation study in Section 4.4.

4.2. Implementation Details

We provide the implementation details for both the model architecture and model training.

Model Architecture. (1) For the *Visual Feature Extraction*, We use the frozen DINOv2 [37] ViT as our feature extractor, and conduct experiments with all of the ViT-S/B/L (small, base, large) model sizes. We use the official released

checkpoint and codebase from Facebook Research. (2) For the *Proposal Generation*, we use the two-stage Deformable DETR with iterative bounding box refinement in our model. A cross-attention layer between the class prototypes and query image features is added in-between the Transformer encoder and decoder. We develop our model based on open-sourced codebase detrex [41] and follow their implementations and use the same hyper-parameters, for example, using 300 object queries in all of our experiments. (3) For the *Few-Shot Proposal Classification*, we use the most recent version of Vicuna [9] (version 1.5) which is instruction fine-tuned from Llama 2 [47] and use Vicuna-7B by default in our experiments following most of the previous multimodal LLMs works [6, 6, 32, 38, 62]. We have two projection layers in our model: one connecting proposals to the LLM and the other connecting class prototype to the LLM. The two projection layers are simply linear layers with different input dimensions, and both of them are randomly initialized from scratch. Note that following previous DETR based models [4, 71], during evaluation, we assign each object query to the class with the highest score or the second highest score if the class with the highest score is <class_bg>, and always assign the bipartite-matched class to each of the object query during training.

Model Training. (1) We implement the first step training as traditional supervised training, and we exactly follow the training details (e.g., learning rate scheduler and training epochs) as the detrex codebase. (2) We implement the second step training as meta-training. Specifically, in each training episode, we randomly sample a N-way K-shot support set together with a query set from the base classes dataset. N is set to be 60 with all the base classes, and K is set to be 30 by default. In practice, we have a pool of support images for each class, cropped around the ground-truth bounding box with some context. The support features are pre-calculate for all classes before training because our feature backbone DINOv2 is frozen all the time. We train the model with 3 epochs, using cosine scheduler with a peak learning rate of 2e-5 and Adam optimizer. Most of our models are trained with 8 A100 80G GPUs with a batch size of 8 per gpu. (3) We conduct fine-tuning similar to (1) and (2), but with smaller training epochs. The N and K are also changed accordingly for each few-shot setting.

4.3. Main Results

We evaluate our proposed method on the two widely used FSOD benchmarks. The main results are shown in the Table 1 and Table 3 for the MSCOCO and PASCAL VOC dataset respectively. We compare our model with a large number of existing works. The existing methods can be roughly divided into two groups according to the detection framework: RCNN based methods (e.g., FCT [17], DiGeo [36]), and DETR based methods (e.g., Meta-DETR [66], FS-DETR [3]). Our method belongs to the second group with the Deformable DETR detection framework.

From Table 1 we can find that our method outperforms the traditional Faster R-CNN based FSOD models [17, 36] significantly, especially for the settings with a relatively large number of shots. For example, our method outperforms FCT [17] by 15.6 AP points on 30-shot, and 10.6 AP points on 10-shot. Using smaller number of shots, the performance gain is smaller, but nontrivial. For example, we outperform FCT [17] by 3.1 AP points on 2-shot, and 4.6 AP points on 3-shot. This verified the effectiveness of utilizing large Foundation Models for FSOD. The DETR based methods [3, 66] usually have better results compared to Faster R-CNN based methods. We similarly observe large performance gain on large number of shots. But both of the two methods [3, 66] outperform our models at 1-shot and 2-shot AP50 metric. We argue that this is because LLM is hard to tune with such small number of training data, and we do not utilize external dataset for FSOD training like [3]. Interesting future works are to introduce external dataset for detection training, and explore efficient training with LLM under small data. Similar performance comparisons can be concluded in the Table 3 of PASCAL VOC 3 splits.

Recently, another strong baseline DE-VIT [69] is proposed, which is Faster R-CNN based, and also utilize DINOv2 as the feature backbone. We directly compare DE-VIT with our method under different DINOv2 model sizes. DE-VIT only provides evaluation results under 10-shot and 30-shot. Our method outperforms DE-VIT on 30-shot, but is inferior to DE-VIT on 10-shot. However, novel classes evaluation is only part of our goal. We need to keep strong performance on base classes as well, which is also called generalized few-shot object detection (G-FSOD). The evaluation results of G-FSOD on MSCOCO 10-shot and 30-shot are shown in the Table 2. Our method has strong performance on both many-shot base classes and few-shot novel classes. The AP scores of our method outperform DE-VIT by more than 10 point on both of the 10-shot and 30-shot settings, and over all model sizes. DiGeo [36], using the same Faster R-CNN framework, also outperforms DE-VIT. DE-VIT transforms multi-class classification into multiple binary classifications with a shared binary classifier. The shared classifier improves generalization to nAP, but potentially harms bAP without learning discriminative knowledge for base classes. Our method shows strong performance on both base and novel classes.

We also perform detection visualization and failure case analysis in the Figure 3. Our model can detect most of the objects of various sizes, under different illuminations, and can detect some of the partially occluded objects. But our method is still struggling in detecting objects in the dark, and with extremely small size or occluded by others. Future work can exploit efficient data augmentation methods

Table 1. Few-shot object detection performance on the MSCOCO dataset with novel classes only. Please find the Table 2 for the full evaluations of both base classes and novel classes on 10-shot and 30-shot.

Method	1-shot			2-shot			3-shot			5-shot			10-shot			30-shot		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
TFA w/ fc [51]	2.9	5.7	2.8	4.3	8.5	4.1	6.7	12.6	6.6	8.4	16.0	8.4	10.0	19.2	9.2	13.4	24.7	13.2
TFA w/ cos [51]	3.4	5.8	3.8	4.6	8.3	4.8	6.6	12.1	6.5	8.3	15.3	8.0	10.0	19.1	9.3	13.7	24.9	13.4
Xiao et al. [59]	3.2	8.9	1.4	4.9	13.3	2.3	6.7	18.6	2.9	8.1	20.1	4.4	10.7	25.6	6.5	15.9	31.7	15.1
MPSR [56]	2.3	4.1	2.3	3.5	6.3	3.4	5.2	9.5	5.1	6.7	12.6	6.4	9.8	17.9	9.7	14.1	25.4	14.2
Fan et al. [12]	4.2	9.1	3.0	5.6	14.0	3.9	6.6	15.9	4.9	8.0	18.5	6.3	9.6	20.7	7.7	13.5	28.5	11.7
FSCE [43]	–	–	–	–	–	–	–	–	–	–	–	–	11.9	-	10.5	16.4	-	16.2
QA-FewDet [14]	4.9	10.3	4.4	7.6	16.1	6.2	8.4	18.0	7.3	9.7	20.3	8.6	11.6	23.9	9.8	16.5	31.9	15.5
Meta Faster R-CNN [16]	5.1	10.7	4.3	7.6	16.3	6.2	9.8	20.2	8.2	10.8	22.1	9.2	12.7	25.7	10.8	16.6	31.8	15.8
FCT [17]	5.6	-	-	7.9	-	-	11.1	-	-	14.0	-	-	17.1	-	-	21.4	-	-
DiGeo [36]	–	–	–	–	–	–	–	–	–	–	–	–	10.3	18.7	9.9	14.2	26.2	14.8
Meta-DETR [66]	7.5	12.5	7.7	-	-	-	13.5	21.7	14.0	15.4	25.0	15.8	19.0	30.5	19.7	22.2	35.0	22.8
FS-DETR [3]	7.0	13.6	7.5	8.9	17.5	9.0	10.0	18.8	10.0	10.9	20.7	10.8	11.3	21.7	11.1	-	-	-
DE-VIT [69] ViT-S	–	–	–	–	–	–	–	–	–	–	–	–	27.1	43.1	28.5	26.9	43.1	28.4
DE-VIT [69] ViT-B	–	–	–	–	–	–	–	–	–	–	–	–	33.2	51.4	35.5	33.4	51.4	35.7
DE-VIT [69] ViT-L	–	–	–	–	–	–	–	–	–	–	–	–	34.0	53.0	37.0	34.0	52.9	37.2
FM-FSOD (Ours) ViT-S	4.5	6.1	5.0	9.4	12.8	10.1	14.6	20.2	15.7	18.7	26.4	20.0	24.3	34.7	26.0	31.6	45.0	33.6
FM-FSOD (Ours) ViT-B	5.0	6.6	5.4	10.1	13.5	10.9	14.8	20.4	16.0	21.0	28.4	22.7	26.8	38.4	28.4	36.8	51.3	39.4
FM-FSOD (Ours) ViT-L	5.7	7.8	6.2	11.0	15.1	11.5	15.7	21.8	16.8	21.9	30.4	23.2	27.7	38.6	30.1	37.0	51.3	39.7

Table 2. Evaluations of both base classes and novel classes on the MSCOCO dataset.

		10-shot			30-shot		
		AP	bAP	nAP	AP	bAP	nAP
DiGeo [36]		32.0	39.2	10.3	33.1	39.4	14.2
DE-VIT [69]	ViT-S	24.8	24	27.1	24.9	24.2	26.9
DE-VIT [69]	ViT-B	29.5	28.3	33.2	29.7	28.5	33.4
DE-VIT [69]	ViT-L	30.6	29.4	34.0	30.6	29.5	34.0
FM-FSOD (Ours)	ViT-S	34.6	38.1	24.2	38.1	40.3	31.6
FM-FSOD (Ours)	ViT-B	37.9	41.6	26.8	42.7	44.7	36.8
FM-FSOD (Ours)	ViT-L	40.0	44.2	27.7	43.1	45.2	37.0

to solve this problem, for example, using the most recent text-to-image generation models [42].

4.4. Ablation Studies

We perform extensive ablation studies on the model architecture and training method in Table 4. We can have the following conclusions: (1) Using the ViT based RCNN detection framework ViTDet, it is important to fine-tune the feature backbone. By comparing the models with frozen backbones and tuned backbones, we can find large performance drop of the former. The reason is that the detection head is too shallow, and do not have enough capabilities for downstream tasks. (2) Using the Transformer-based detection framework, Deformable DETR, can significantly improve the performance for both base and novel classes, even with frozen feature backbone. This is due to the additional learning capacities brought by the Transformer-based detection head. If we also tune the backbone, we can only get small improvement especially for the strong DINOv2 based model, but tuning the backbone brings significant computational burden. This is because the support features cannot be pre-calculated if the parameters of the backbone are tuned. (3) DINOv2 based models show better performance compared with the models using other pre-trained vision encoders, e.g., MAE [19], CLIP [40], SWIN [34]

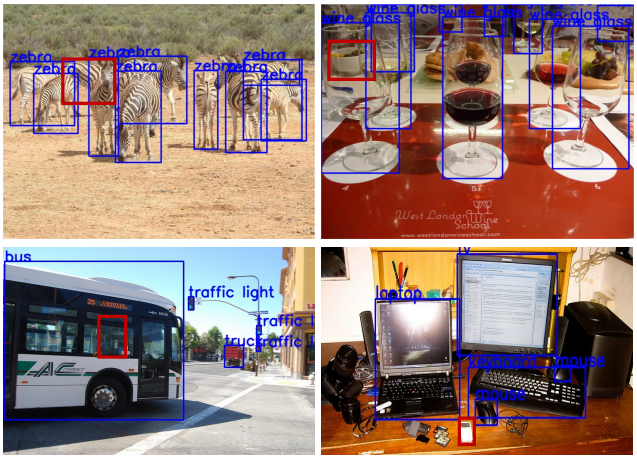

Figure 3. Detection Visualization and Failure Case Analysis. Blue means our detection results and red means false negatives. We use our 30-shot fine-tuned G-FSOD model for visualization.

Table 3. Few-shot object detection performance (AP50) on the PASCAL VOC dataset with novel classes only.

	Novel Set 1					Novel Set 2					Novel Set 3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
TFA w/ fc [51]	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA w/ cos [51]	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
Xiao et al. [59]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
MPSR [56]	41.7	42.5	51.4	55.2	61.8	24.4	29.3	39.2	39.9	47.8	35.6	41.8	42.3	48.0	49.7
Fan et al. [12]	37.8	43.6	51.6	56.5	58.6	22.5	30.6	40.7	43.1	47.6	31.0	37.9	43.7	51.3	49.8
FSCE [43]	44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5
QA-FewDet [14]	42.4	51.9	55.7	62.6	63.4	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5
Meta Faster R-CNN [16]	43.0	54.5	60.6	66.1	65.4	27.7	35.5	46.1	47.8	51.4	40.6	46.4	53.4	59.9	58.6
FCT [17]	49.9	57.1	57.9	63.2	67.1	27.6	34.5	43.7	49.2	51.2	39.5	54.7	52.3	57.0	58.7
FS-DETR [3]	45.0	48.5	51.5	52.7	56.1	37.3	41.3	43.4	46.6	49.0	43.8	47.1	50.6	52.1	56.9
Meta-DETR [66]	40.6	51.4	58.0	59.2	63.6	37.0	36.6	43.7	49.1	54.6	41.6	45.9	52.7	58.9	60.6
FM-FSOD (Ours) ViT-S	41.6	49.0	55.8	61.2	67.7	34.7	37.6	47.6	52.5	58.7	39.5	47.8	54.4	57.8	62.6
FM-FSOD (Ours) ViT-B	40.9	52.8	59.5	68.3	71.4	33.5	36.1	48.1	53.6	59.3	41.9	52.6	54.9	62.8	68.2
FM-FSOD (Ours) ViT-L	40.1	53.5	57.0	68.6	72.0	33.1	36.3	48.8	54.8	64.7	39.2	50.2	55.7	63.4	68.1

Table 4. Ablation study of major components on COCO 10-shot setting. † Deformable DETR. ‡ We only fine-tune the Transformer blocks in the final stage as defined in ViTDet.

	Framework	Freeze backbone	LLM	10-shot AP	10-shot bAP	10-shot nAP
MAE ViT-B	ViTDet	✓		12.1	14.0	6.4
MAE ViT-B	ViTDet			35.5	41.9	16.1
CLIP ViT-B	ViTDet	✓		10.6	11.7	7.4
CLIP ViT-B	ViTDet			29.1	35.6	9.5
SAM ViT-B	ViTDet	✓		18.5	20.8	11.4
SAM ViT-B	ViTDet	partial‡		32.5	39.0	12.7
SAM ViT-B	ViTDet			34.4	42.5	10.2
DINOv2 ViT-B	ViTDet	✓		29.7	31.9	23.1
DINOv2 ViT-B	ViTDet	partial		35.0	41.3	16.1
DINOv2 ViT-B	ViTDet			35.9	43.7	12.5
MAE ViT-B	D-DETR†	✓		20.4	23.4	11.5
MAE ViT-B	D-DETR			35.3	40.7	18.8
CLIP ViT-B	D-DETR	✓		23.2	26.6	13.1
CLIP ViT-B	D-DETR			32.3	37.8	15.5
SWIN-B	D-DETR	✓		36.0	40.3	23.0
SWIN-B	D-DETR			39.5	45.3	22.0
SAM ViT-B	D-DETR	✓		25.5	29.0	15.3
SAM ViT-B	D-DETR	partial		31.3	36.9	14.5
SAM ViT-B	D-DETR			34.3	41.5	12.6
DINOv2 ViT-B	D-DETR	✓		36.5	40.2	25.4
DINOv2 ViT-B	D-DETR	partial		38.0	44.5	18.5
DINOv2 ViT-B	D-DETR			38.0	44.8	17.3
FM-FSOD ViT-S	D-DETR	✓	✓	34.6	38.1	24.2
FM-FSOD ViT-B	D-DETR	✓	✓	37.9	41.6	26.8
FM-FSOD ViT-L	D-DETR	✓	✓	40.0	44.2	27.7

and SAM [24], especially for novel classes. This verifies the effectiveness of large scale self-supervised pre-training at both global and local levels. (4) Using LLM as few-shot learner can further improve the performance, compared with the Deformable DETR only model. This verifies the effectiveness of our model by introducing additional context information and prior knowledge for few-shot learning.

Table 5. Ablation study of contextual modeling with LLM. † We calculate the total evaluation time on ∼5k images with 8 A100.

	Default model	Use LLM for each proposal separately	Use LLM for each class separately
Running Speed†	25 mins	3 days	20 hours
COCO 10-shot AP	37.9	36.8	37.4

(5) We further ablate the importance of our contextualized few-shot learning in Table 5. We show the experiments of using LLM to classify for each proposal separately, and using LLM to classify for each class separately. The performance decreases slightly in the two models. More importantly, the running speed of the two models is much slower. This shows the effectiveness of our contextual modeling.

5. Conclusion

In this work, we study few-shot object detection using modern foundation models. First, the pre-trained DINOv2 model is used for the vision backbone and is frozen during training, which achieves strong performance for both base classes and novel classes. Second, Large Language Model (LLM) is employed for contextualized few-shot learning, taking input of all proposals and the visual lookup table for all classes. Language instructions are carefully designed to prompt the LLM to classify each proposal in context. The in-context language instructions with LLMs can simplify the modeling of query-support few-shot learning network, and automatically exploit rich contextual information among all the proposals and classes to facilitate the few-shot learning. We comprehensively evaluate the proposed model (FM-FSOD) on both MSCOCO and PASCAL VOC benchmarks, achieving state-of-the-arts performance.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2, 3

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2, 3

[3] Adrian Bulat, Ricardo Guerrero, Brais Martinez, and Georgios Tzimiropoulos. Fs-detr: Few-shot detection transformer with prompting and without re-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11793–11802, 2023. 1, 6, 7, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 4, 5, 6

[5] Ding-Jie Chen, He-Yen Hsieh, and Tyng-Luh Liu. Adaptive image transformer for one-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12247–12256, 2021. 1, 2

[6] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 2, 3, 5, 6

[7] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. 3, 5

[8] Tung-I Chen, Yueh-Cheng Liu, Hung-Ting Su, Yu-Cheng Chang, Yu-Hsiang Lin, Jia-Fong Yeh, Wen-Chin Chen, and Winston Hsu. Dual-awareness attention for few-shot object detection. *IEEE Transactions on Multimedia*, pages 1–1, 2021. 2

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 2, 5, 6

[10] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *Advances in Neural Information Processing Systems*, pages 21981–21993. Curran Associates, Inc., 2020. 2

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[12] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020. 1, 2, 3, 4, 7, 8

[13] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1

[14] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3263–3272, 2021. 1, 2, 4, 5, 7, 8

[15] Guangxing Han, Long Chen, Jiawei Ma, Shiyuan Huang, Rama Chellappa, and Shih-Fu Chang. Multi-modal few-shot object detection with meta-learning-based cross-modal prompting. *arXiv preprint arXiv:2204.07841*, 2022. 1

[16] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 780–789, 2022. 1, 2, 4, 7, 8

[17] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5321–5330, 2022. 1, 2, 4, 6, 7, 8

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 3, 7

[20] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*, pages 2725–2734, 2019. 1, 2

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

[22] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8420–8429, 2019. 1, 2, 5

[23] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2019. 2

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3, 8

[25] Anis Koubaa. Gpt-4 vs. gpt-3.5: A concise showdown. 2023. 2

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3

[27] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 1

[28] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. 2

[29] Han Lin, Guangxing Han, Jiawei Ma, Shiyuan Huang, Xudong Lin, and Shih-Fu Chang. Supervised masked knowledge distillation for few-shot transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19649–19659, 2023. 3

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2, 3, 6

[33] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 7

[35] Jiawei Ma, Guangxing Han, Shiyuan Huang, Yuncong Yang, and Shih-Fu Chang. Few-shot end-to-end object detection via constantly concentrated encoding across heads. In *European Conference on Computer Vision*, pages 57–73. Springer, 2022. 1

[36] Jiawei Ma, Yulei Niu, Jincheng Xu, Shiyuan Huang, Guangxing Han, and Shih-Fu Chang. Digeo: Discriminative geometry-aware learning for generalized few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3208–3218, 2023. 5, 6, 7

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 5

[38] Shraman Pramanick, Guangxing Han, Rui Hou, Sayan Nag, Ser-Nam Lim, Nicolas Ballas, Qifan Wang, Rama Chellappa, and Amjad Almahairi. Jack of all tasks, master of many: Designing general-purpose coarse-to-fine vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 6

[39] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8681–8690, 2021. 5

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 7

[41] Tianhe Ren, Shilong Liu, Feng Li, Hao Zhang, Ailing Zeng, Jie Yang, Xingyu Liao, Ding Jia, Hongyang Li, He Cao, et al. detrex: Benchmarking detection transformers. *arXiv preprint arXiv:2306.07265*, 2023. 6

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 7

[43] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7352–7362, 2021. 1, 2, 5, 7, 8

[44] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 3

[45] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023. 3

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 5

[47] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6

[48] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 2, 3

[49] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an

open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 2, 3, 5

[50] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[51] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning (ICML)*, 2020. 2, 5, 7, 8

[52] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023. 3

[53] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 3

[54] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 3

[55] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 3

[56] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *European Conference on Computer Vision*, pages 456–472. Springer, 2020. 2, 7, 8

[57] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7031–7040, 2023. 1

[58] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*, 2023. 3, 5

[59] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European Conference on Computer Vision*, 2020. 2, 7, 8

[60] Jingyi Xu, Hieu Le, and Dimitris Samaras. Generating features with increased crop-related diversity for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19713–19722, 2023. 1

[61] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9577–9586, 2019. 2

[62] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023. 2, 3, 5, 6

[63] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 3

[64] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 3

[65] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021. 1

[66] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P Xing. Meta-detr: Image-level few-shot detection with inter-class correlation exploitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 6, 7, 8

[67] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. 2, 3, 5

[68] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13008–13017, 2021. 1, 2

[69] Xinyu Zhang, Yuting Wang, and Abdeslam Boularias. Detect every thing with few examples. *arXiv preprint arXiv:2309.12969*, 2023. 1, 2, 4, 6, 7

[70] Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8782–8791, 2021. 2

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2, 4, 5, 6