YOLO-World: Real-Time Open-Vocabulary Object Detection

Tianheng Cheng[3,2,⋆], Lin Song[1,⋆,✉], Yixiao Ge[1,2,†], Wenyu Liu[3], Xinggang Wang[3,✉], Ying Shan[1,2]

⋆ **equal contribution** † **project lead** ✉ **corresponding author**

[1] Tencent AI Lab [2] ARC Lab, Tencent PCG

[3] School of EIC, Huazhong University of Science & Technology

Abstract

The You Only Look Once (YOLO) series of detectors have established themselves as efficient and practical tools. However, their reliance on predefined and trained object categories limits their applicability in open scenarios. Addressing this limitation, we introduce YOLO-World, an innovative approach that enhances YOLO with open-vocabulary detection capabilities through vision-language modeling and pre-training on large-scale datasets. Specifically, we propose a new Re-parameterizable Vision-Language Path Aggregation Network (RepVL-PAN) and region-text contrastive loss to facilitate the interaction between visual and linguistic information. Our method excels in detecting a wide range of objects in a zero-shot manner with high efficiency. On the challenging LVIS dataset, YOLO-World achieves 35.4 AP with 52.0 FPS on V100, which outperforms many state-of-the-art methods in terms of both accuracy and speed. Furthermore, the fine-tuned YOLO-World achieves remarkable performance on several downstream tasks, including object detection and open-vocabulary instance segmentation. Code and models are available at: `https://github.com/AILab-CVC/YOLO-World`.

1. Introduction

Object detection has been a long-standing and fundamental challenge in computer vision with numerous applications in image understanding, robotics, and autonomous vehicles. Tremendous works [17, 27, 43, 45] have achieved significant breakthroughs in object detection as the development of deep neural networks. Despite the success of these methods, they remain limited as they only detect objects with a fixed vocabulary, *e.g.*, 80 categories in the COCO [26] dataset. Once object categories are defined and labeled, trained detectors can only detect those specific categories, thus limiting the ability and applicability of open scenarios.

Recent works [9, 14, 48, 53, 59] have explored the prevalent vision-language models [19, 39] to address open-

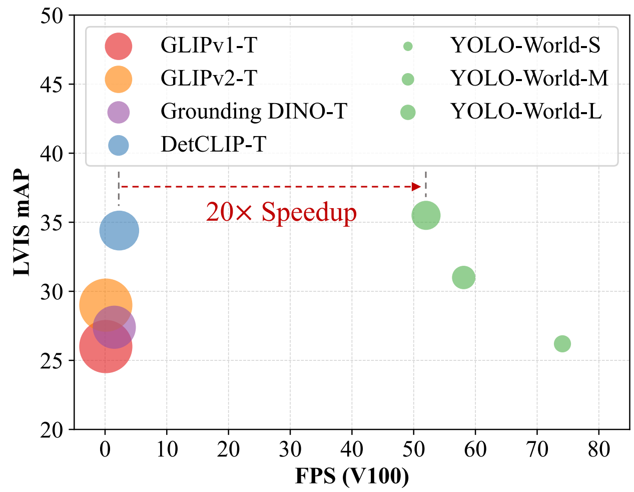
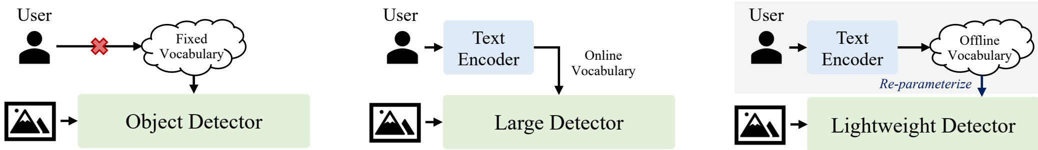


Figure 1. **Speed-and-Accuracy Curve.** We compare YOLO-World with recent open-vocabulary methods pre-trained similar datasets (*i.e.*, Objects365 [46] and GoldG [21]) in terms of speed and accuracy. All models are evaluated on the LVIS `minival` (using *Fixed AP* [4]) and inference speeds are measured on one NVIDIA V100 w/o TensorRT. The size of the circle represents the model's size.

vocabulary detection [59] by distilling vocabulary knowledge from language encoders, *e.g.*, BERT [6]. However, these distillation-based methods are much limited due to the scarcity of training data with a limited diversity of vocabulary, *e.g.*, OV-COCO [59] containing 48 base categories. Several methods [24, 30, 56, 57, 60] reformulate object detection training as region-level vision-language pre-training and train open-vocabulary object detectors at scale. However, those methods still struggle for detection in real-world scenarios, which suffer from two aspects: (1) heavy computation burden and (2) complicated deployment for edge devices. Previous works [24, 30, 56, 57, 60] have demonstrated the promising performance of pre-training large detectors while pre-training small detectors to endow them with open recognition capabilities remains unexplored.

In this paper, we present YOLO-World, aiming for high-efficiency open-vocabulary object detection, and ex-



(a) Traditional Object Detector (b) Previous Open-Vocabulary Detector (c) YOLO-World

Figure 2. **Comparison with Detection Paradigms. (a) Traditional Object Detector**: These object detectors can only detect objects within the fixed vocabulary pre-defined by the training datasets, *e.g.*, 80 categories of COCO dataset [26]. The fixed vocabulary limits the extension for open scenes. **(b) Previous Open-Vocabulary Detectors:** Previous methods tend to develop large and heavy detectors for open-vocabulary detection which intuitively have strong capacity. In addition, these detectors simultaneously encode images and texts as input for prediction, which is time-consuming for practical applications. **(c) YOLO-World:** We demonstrate the strong open-vocabulary performance of lightweight detectors, *e.g.*, YOLO detectors [20, 42], which is of great significance for real-world applications. Rather than using online vocabulary, we present a *prompt-then-detect* paradigm for efficient inference, in which the user generates a series of prompts according to the need and the prompts will be encoded into an offline vocabulary. Then it can be re-parameterized as the model weights for deployment and further acceleration.

plore large-scale pre-training schemes to boost the traditional YOLO detectors to a new open-vocabulary world. Compared to previous methods, the proposed YOLO-World is remarkably efficient with high inference speed and easy to deploy for downstream applications. Specifically, YOLO-World follows the standard YOLO architecture [20] and leverages the pre-trained CLIP [39] text encoder to encode the input texts. We further propose the Re-parameterizable Vision-Language Path Aggregation Network (RepVL-PAN) to connect text features and image features for better visual-semantic representation. During inference, the text encoder can be removed and the text embeddings can be re-parameterized into weights of RepVL-PAN for efficient deployment. We further investigate the pre-training scheme for YOLO detectors through region-text contrastive learning on large-scale datasets, which unifies detection, grounding, and image-text data into region-text pairs. The pre-trained YOLO-World with abundant region-text pairs demonstrates a strong capability for open-vocabulary detection and training more data leads to greater improvements in open-vocabulary capability.

In addition, we explore a *prompt-then-detect* paradigm to further improve the efficiency of open-vocabulary object detection in real-world scenarios. As illustrated in Fig. 2, traditional object detectors [17, 20, 23, 41–43, 52] concentrate on the fixed-vocabulary (close-set) detection with predefined and trained categories. While previous open-vocabulary detectors [24, 30, 56, 60] encode the prompts of a user for online vocabulary with text encoders and detect objects. Notably, those methods tend to employ large detectors with heavy backbones, *e.g.*, Swin-L [32], to increase the open-vocabulary capacity. In contrast, the *prompt-then-detect* paradigm (Fig. 2 (c)) first encodes the prompts of a user to build an offline vocabulary and the vocabulary varies with different needs. Then, the efficient detector can infer the offline vocabulary on the fly without re-encoding the prompts. For practical applications, once we have trained the detector, *i.e.*, YOLO-World, we can pre-encode the prompts or categories to build an offline vocabulary and then seamlessly integrate it into the detector.

Our main contributions can be summarized into three folds:

- We introduce the YOLO-World, a cutting-edge open-vocabulary object detector with high efficiency for real-world applications.
- We propose a Re-parameterizable Vision-Language PAN to connect vision and language features and an open-vocabulary region-text contrastive pre-training scheme for YOLO-World.
- The proposed YOLO-World pre-trained on large-scale datasets demonstrates strong zero-shot performance and achieves 35.4 AP on LVIS with 52.0 FPS. The pre-trained YOLO-World can be easily adapted to downstream tasks, *e.g.*, open-vocabulary instance segmentation and referring object detection. Moreover, the pre-trained weights and codes of YOLO-World will be open-sourced to facilitate more practical applications.

2. Related Works

2.1. Traditional Object Detection

Prevalent object detection research concentrates on fixed-vocabulary (close-set) detection, in which object detectors are trained on datasets with pre-defined categories, *e.g.*, COCO dataset [26] and Objects365 dataset [46], and then detect objects within the fixed set of categories. During the past decades, the methods for traditional object detection can be simply categorized into three groups, *i.e.*, region-based methods, pixel-based methods, and query-based methods. The region-based methods [12, 13, 17, 27, 44], such as Faster R-CNN [44], adopt a two-stage framework for proposal generation [44] and RoI-wise (Region-of-Interest) classification and regression. The pixel-based methods [28, 31, 42, 49, 62] tend to be one-stage detectors, which perform classification and regression over pre-defined anchors or pixels. DETR [1] first explores object

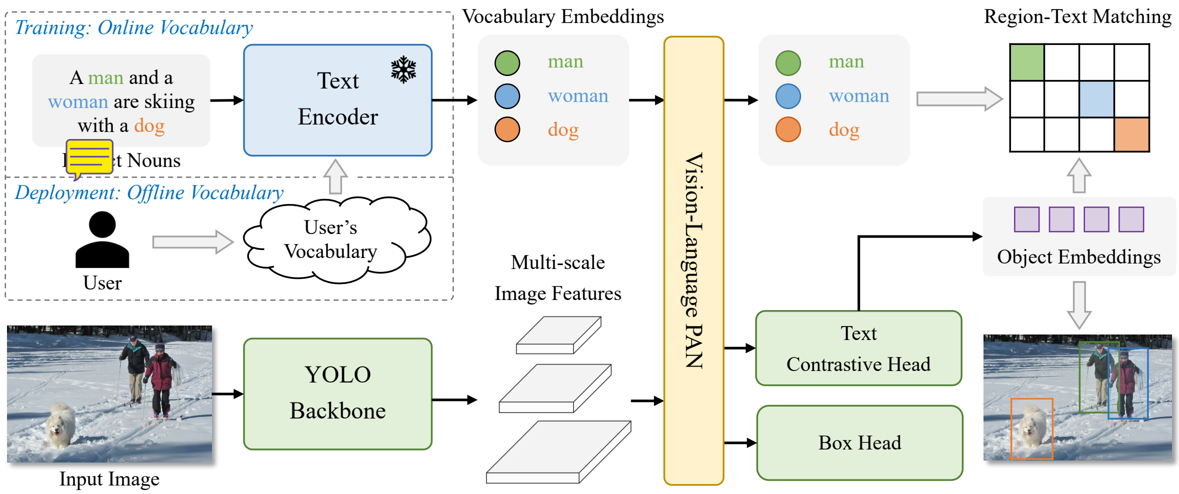


Figure 3. **Overall Architecture of YOLO-World.** Compared to traditional YOLO detectors, YOLO-World as an open-vocabulary detector adopts text as input. The *Text Encoder* first encodes the input text input text embeddings. Then the *Image Encoder* encodes the input image into multi-scale image features and the proposed *RepVL-PAN* exploits the multi-level cross-modality fusion for both image and text features. Finally, YOLO-World predicts the regressed bounding boxes and the object embeddings for matching the categories or nouns that appeared in the input text.

detection through transformers [50] and inspires extensive query-based methods [65]. In terms of inference speed, Redmon *et al*. presents YOLOs [40–42] which exploit simple convolutional architectures for real-time object detection. Several works [11, 23, 33, 52, 55] propose various architectures or designs for YOLO, including path aggregation networks [29], cross-stage partial networks [51], and re-parameterization [7], which further improve both speed and accuracy. In comparison to previous YOLOs, YOLO-World in this paper aims to detect objects beyond the fixed vocabulary with strong generalization ability.

2.2. Open-Vocabulary Object Detection

Open-vocabulary object detection (OVD) [59] has emerged as a new trend for modern object detection, which aims to detect objects beyond the predefined categories. Early works [9, 14, 48, 53, 58] follow the standard OVD setting [59] by training detectors on the base classes and evaluating the novel (unknown) classes. Nevertheless, this open-vocabulary setting can evaluate the capability of detectors to detect and recognize novel objects, it is still limited for open scenarios and lacks generalization ability to other domains due to training on the limited dataset and vocabulary. Detic [64] incorporates the image classification datasets [5] to extend the object categories for large vocabulary detection. Inspired by vision-language pre-training [19, 39], recent works [35, 36, 56, 57, 60, 63] formulate open-vocabulary object detection as image-text matching and exploit large-scale image-text data to increase the training vocabulary at scale. OWL-ViTs [35, 36] fine-tune the simple vision transformers [8] with detection and large-scale automatic labeled datasets and build the simple open-vocabulary detectors. GLIP [24] presents a pre-training framework for open-vocabulary detection based on phrase grounding and evaluates in a zero-shot setting. Grounding DINO [30] incorporates the grounded pre-training [24] into detection transformers [61] with cross-modality fusions. Several methods [25, 56, 57, 60] unify detection datasets and image-text datasets through region-text matching and pre-train detectors with large-scale image-text pairs, achieving promising performance and generalization. However, these methods often use heavy detectors like ATSS [62] or DINO [61] with Swin-L [32] as a backbone, leading to high computational demands and deployment challenges. In contrast, we present YOLO-World, aiming for efficient open-vocabulary object detection with real-time inference and easier downstream application deployment. Differing from ZSD-YOLO [54], which also explores open-vocabulary detection [59] with YOLO through language model alignment, YOLO-World introduces a novel YOLO framework with an effective pre-training strategy, enhancing open-vocabulary performance and generalization.

3. Method

3.1. Pre-training Formulation: Region-Text Pairs

The traditional object detection methods, including the YOLO-series [20], are trained with instance annotations $\Omega = \{B_i, c_i\}_{i=1}^N$, which consist of bounding boxes $\{B_i\}$ and category labels $\{c_i\}$. In this paper, we reformulate the instance annotations as region-text pairs $\Omega = \{B_i, t_i\}_{i=1}^N$, where t_i is the corresponding text for the region B_i. Specifically, the text t_i can be the category name, noun phrases, or object descriptions. YOLO-World adopts the image I and texts T as input and outputs predicted boxes $\{\hat{B}_k\}$ and the corresponding object embeddings $\{e_k\}$ ($e_k \in \mathbb{R}^D$).

3.2. Model Architecture

The overall architecture of the proposed YOLO-World is illustrated in Fig. 3, which consists of a *YOLO detector*, a *Text Encoder*, and a *Re-parameterizable Vision-Language Path Aggregation Network* (RepVL-PAN). Given the input text, the text encoder in YOLO-World encodes the text into text embeddings. The image encoder in the YOLO detector extracts the multi-scale features from the input image. Then we leverage the RepVL-PAN to enhance both text and image representation by exploiting the cross-modality fusion between image features and text embeddings.

YOLO Detector. YOLO-World is mainly developed based on YOLOv8 [20], which contains a Darknet backbone [20, 43] as the image encoder, a path aggregation network (PAN) for multi-scale feature pyramids, and a head for bounding box regression and object embeddings.

Text Encoder. Given the text T, we adopt the text encoder pre-trained by CLIP [39] to extract the corresponding text embeddings $W = \texttt{TextEncoder}(T) \in \mathbb{R}^{C\times D}$, where C is the number of nouns and D is the embedding dimension. The CLIP text encoder offers better visual-semantic capabilities for connecting visual objects with texts compared to text-only language encoders [6]. When the input text is a caption, we adopt the simple noun extraction algorithm to extract the noun phrases and feed them into the text encoder.

Text Contrastive Head. Following previous works [20], we adopt the decoupled head with two 3×3 convs to regress bounding boxes $\{b_k\}_{k=1}^K$ and object embeddings $\{e_k\}_{k=1}^K$, where K denotes the number of objects. We present a text contrastive head to obtain the object-text similarity $s_{k,j}$ by:

$$s_{k,j} = \alpha \cdot \texttt{L2-Norm}(e_k) \cdot \texttt{L2-Norm}(w_j)^\top + \beta, \quad (1)$$

where $\texttt{L2-Norm}(\cdot)$ is the L2 normalization and $w_j \in W$ is the j-th text embeddings. In addition, we add the affine transformation with the learnable scaling factor α and shifting factor β. Both the L2 norms and the affine transformations are important for stabilizing the region-text training.

Training with Online Vocabulary. During training, we construct an online vocabulary T for each mosaic sample containing 4 images. Specifically, we sample all positive nouns involved in the mosaic images and randomly sample some negative nouns from the corresponding dataset. The vocabulary for each mosaic sample contains at most M nouns, and M is set to 80 as default.

Inference with Offline Vocabulary. At the inference stage, we present a *prompt-then-detect* strategy with an offline vocabulary for further efficiency. As shown in Fig. 3, the user can define a series of custom prompts, which might include captions or categories. We then utilize the text encoder to encode these prompts and obtain offline vocabulary embeddings. The offline vocabulary allows for avoiding computation for each input and provides the flexibility to adjust the vocabulary as needed.

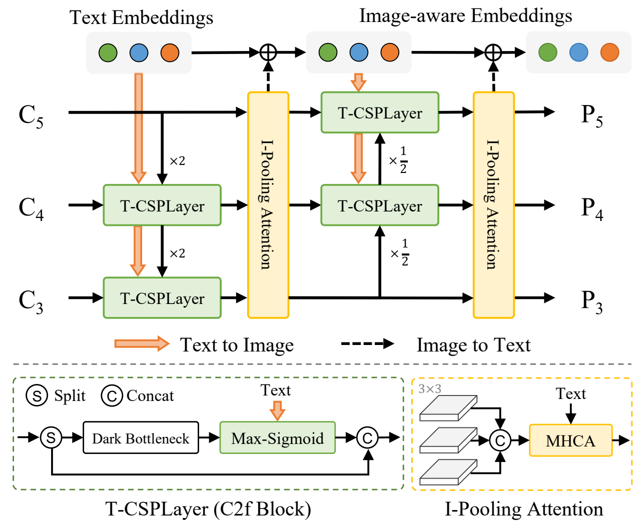


Figure 4. **Illustration of the RepVL-PAN.** The proposed RepVL-PAN adopts the *Text-guided CSPLayer* (T-CSPLayer) for injecting language information into image features and the *Image Pooling Attention* (I-Pooling Attention) for enhancing text embeddings.

3.3. Re-parameterizable Vision-Language PAN

Fig. 4 shows the structure of the proposed RepVL-PAN which follows the top-down and bottom-up paths in [20, 29] to establish the feature pyramids $\{P_3, P_4, P_5\}$ with the multi-scale image features $\{C_3, C_4, C_5\}$. Furthermore, we propose the Text-guided CSPLayer (T-CSPLayer) and Image-Pooling Attention (I-Pooling Attention) to further enhance the interaction between image features and text features, which can improve the visual-semantic representation for open-vocabulary capability. During inference, the offline vocabulary embeddings can be re-parameterized into weights of convolutional or linear layers for deployment.

Text-guided CSPLayer. As Fig. 4 illustrates, the cross-stage partial layers (CSPLayer) are utilized after the top-down or bottom-up fusion. We extend the CSPLayer (also called $\texttt{C2f}$) of [20] by incorporating text guidance into multi-scale image features to form the Text-guided CSPLayer. Specifically, given the text embeddings W and image features $X_l \in \mathbb{R}^{H\times W\times D}$ ($l \in \{3,4,5\}$), we adopt the *max-sigmoid attention* after the last dark bottleneck block to aggregate text features into image features by:

$$X_l' = X_l \cdot \delta(\max_{j\in\{1..C\}}(X_l W_j^\top))^\top, \quad (2)$$

where the updated X_l' is concatenated with the cross-stage features as output. The δ indicates the sigmoid function.

Image-Pooling Attention. To enhance the text embeddings with image-aware information, we aggregate image features to update the text embeddings by proposing the Image-Pooling Attention. Rather than directly using cross-attention on image features, we leverage max pooling on multi-scale features to obtain 3×3 regions, resulting in a total of 27 patch tokens $\tilde{X}\in\mathbb{R}^{27\times D}$. The text embeddings are then updated by:

$$W' = W + \texttt{MultiHead-Attention}(W, \tilde{X}, \tilde{X}) \quad (3)$$

3.5. Pre-training Schemes

In this section, we present the training schemes for pre-training YOLO-World on large-scale detection, grounding, and image-text datasets.

Learning from Region-Text Contrastive Loss. Given the mosaic sample I and texts T, YOLO-World outputs K object predictions $\{B_k, s_k\}_{k=1}^{K}$ along with annotations $\Omega = \{B_i, t_i\}_{i=1}^{N}$. We follow [20] and leverage task-aligned label assignment [10] to match the predictions with ground-truth annotations and assign each positive prediction with a text index as the classification label. Based on this vocabulary, we construct the region-text contrastive loss $\mathcal{L}_{\text{con}}$ with region-text pairs through cross entropy between object-text (region-text) similarity and object-text assignments. In addition, we adopt IoU loss and distributed focal loss for bounding box regression and the total training loss is defined as: $\mathcal{L}(I) = \mathcal{L}_{\text{con}} + \lambda_I \cdot (\mathcal{L}_{\text{iou}} + \mathcal{L}_{\text{dfl}})$, where λ_I is an indicator factor and set to 1 when input image I is from detection or grounding data and set to 0 when it is from the image-text data. Considering image-text datasets have noisy boxes, we only calculate the regression loss for samples with accurate bounding boxes.

Pseudo Labeling with Image-Text Data. Rather than directly using image-text pairs for pre-training, we propose an automatic labeling approach to generate region-text pairs. Specifically, the labeling approach contains three steps: (1) *extract noun phrases*: we first utilize the noun extraction algorithm to extract noun phrases from the text; (2) *pseudo labeling*: we adopt a pre-trained open-vocabulary detector, *e.g.*, GLIP [24], to generate pseudo boxes for the given noun phrases for each image, thus providing the coarse region-text pairs. (3) *filtering*: We employ the pre-trained CLIP [39] to evaluate the relevance of image-text pairs and region-text pairs, and filter the low-relevance pseudo annotations and images. We further filter redundant bounding boxes by incorporating methods such as Non-Maximum Suppression (NMS). We suggest the readers refer to the appendix for the detailed approach. With the above approach, we sample and label 246k images from CC3M [47] with 821k pseudo annotations to construct the CC3M-Lite dataset.

Dataset	Type	Vocab.	Images	Anno.
Objects365V1 [46]	Detection	365	609k	9,621k
GQA [18]	Grounding	-	621k	3,681k
Flickr [38]	Grounding	-	149k	641k
CC3M† [47]	Image-Text	-	246k	821k

Table 1. **Pre-training Data.** The specifications of the datasets used for pre-training YOLO-World.

4. Experiments

In this section, we demonstrate the effectiveness of the proposed YOLO-World by pre-training it on large-scale datasets and evaluating YOLO-World in a zero-shot manner on both LVIS benchmark and COCO benchmark (Sec. 4.2). We also evaluate the fine-tuning performance of YOLO-World on COCO, LVIS for object detection.

4.1. Implementation Details

The YOLO-World is developed based on the MMYOLO toolbox [3] and the MMDetection toolbox [2]. Following [20], we provide three variants of YOLO-World for different latency requirements, *e.g.*, small (S), medium (M), and large (L). We adopt the open-source CLIP [39] text encoder with pre-trained weights to encode the input text. Unless specified, we measure the inference speeds of all models on one NVIDIA V100 GPU without extra acceleration mechanisms, *e.g.*, FP16 or TensorRT.

4.2. Pre-training

Experimental Setup. At the pre-training stage, we adopt the AdamW optimizer [34] with an initial learning rate of 0.002 and weight decay of 0.05. YOLO-World is pre-trained for 100 epochs on on 32 NVIDIA V100 GPUs with a total batch size of 512. During pre-training, we follow previous works [20] and adopt color augmentation, random affine, random flip, and mosaic with 4 images for data augmentation. The text encoder is frozen during pre-training.

Pre-training Data. For pre-training YOLO-World, we mainly adopt detection or grounding datasets including Objects365 (V1) [46], GQA [18], Flickr30k [38], as specified in Tab. 1. Following [24], we exclude the images from the COCO dataset in GoldG [21] (GQA and Flickr30k). The annotations of the detection datasets used for pre-training contain both bounding boxes and categories or noun phrases. In addition, we also extend the pre-training data with image-text pairs, *i.e.*, CC3M-Lite, which we have sampled and labeled 246k images from CC3M [47].

Zero-shot Evaluation. After pre-training, we evaluate the proposed YOLO-World on the LVIS dataset [15] in a zero-shot manner. The LVIS dataset contains 1203 object categories, which is much more than the categories of the pre-training detection datasets and can measure the performance on large vocabulary detection. Following previous works [21, 24, 56, 57], we mainly evaluate on LVIS `minival` [21] and report the *Fixed AP* [4] for comparison. The maximum number of predictions is set to 1000.

Main Results on LVIS Object Detection. In Tab. 2, we compare the proposed YOLO-World with recent state-of-the-art methods [21, 30, 56, 57, 60] (pre-trained on similar datasets) on LVIS benchmark in a zero-shot manner. Considering the computation burden and model parameters, we mainly compare with those methods based on lighter backbones, *e.g.*, Swin-T [32]. Remarkably, YOLO-World outperforms previous state-of-the-art methods in terms of zero-shot performance and inference speed. Compared to GLIP, GLIPv2, and Grounding DINO, which incorporate more data, *e.g.*, Cap4M (CC3M+SBU [37]), YOLO-World pre-trained on O365 & GolG obtains better performance even with fewer model parameters. Compared to DetCLIP, YOLO-World achieves comparable performance (35.4 *v.s.* 34.4) while obtaining 20× increase in inference speed. *The experimental results also demonstrate that small models,* e.g.*, YOLO-World-S with 13M parameters, can be used for vision-language pre-training and obtain strong open-vocabulary capabilities.*

4.3. Ablation Experiments

We provide extensive ablation studies to analyze YOLO-World from two primary aspects, *i.e.*, pre-training and architecture. Unless specified, we mainly conduct ablation experiments based on YOLO-World-L and pre-train Objects365 with zero-shot evaluation on LVIS `minival`.

Pre-training Data. In Tab. 3, we evaluate the performance of pre-training YOLO-World using different data. Compared to the baseline trained on Objects365, adding GQA can significantly improve performance with an 8.4 AP gain on LVIS. This improvement can be attributed to the richer textual information provided by the GQA dataset, which can enhance the model's ability to recognize large vocabulary objects. Adding part of CC3M samples (8% of the full datasets) can further bring 0.5 AP gain with 1.3 AP on rare objects. Tab. 3 demonstrates that adding more data can effectively improve the detection capabilities on large-vocabulary scenarios. Furthermore, as the amount of data increases, the performance continues to improve, highlighting the benefits of leveraging larger and more diverse datasets for training.

Ablations on RepVL-PAN. Tab. 4 demonstrates the effectiveness of the proposed RepVL-PAN of YOLO-World, including Text-guided CSPLayers and Image Pooling Attention, for the zero-shot LVIS detection. Specifically, we adopt two settings, *i.e.*, (1) pre-training on O365 and (2) pre-training on O365 & GQA. Compared to O365 which only contains category annotations, GQA includes rich texts, particularly in the form of noun phrases. As shown in Tab. 4, the proposed RepVL-PAN improves the baseline (YOLOv8-PAN [20]) by 1.1 AP on LVIS, and the improvements are remarkable in terms of the rare categories (AP_r) of LVIS, which are hard to detect and recognize. In addition, the improvements become more significant when YOLO-World is pre-trained with the GQA dataset and experiments indicate that the proposed RepVL-PAN works better with rich textual information.

Text Encoders. In Tab. 5, we compare the performance of using different text encoders, *i.e.*, BERT-base [6] and CLIP-base (ViT-base) [39]. We exploit two settings during pre-training, *i.e.*, frozen and fine-tuned, and the learning rate for fine-tuning text encoders is a 0.01× factor of the basic learning rate. As Tab. 5 shows, the CLIP text encoder obtains superior results than BERT (+10.1 AP for rare categories in LVIS), which is pre-trained with image-text pairs and has better capability for vision-centric embeddings. Fine-tuning BERT during pre-training brings significant improvements (+3.7 AP) while fine-tuning CLIP leads to a severe performance drop. We attribute the drop to that fine-tuning on O365 may degrade the generalization ability of the pre-trained CLIP, which contains only 365 categories and lacks abundant textual information.

4.4. Fine-tuning YOLO-World

In this section, we further fine-tune YOLO-World for close-set object detection on the COCO dataset and LVIS dataset to demonstrate the effectiveness of the pre-training.

Experimental Setup. We use the pre-trained weights to initialize YOLO-World for fine-tuning. All models are fine-tuned for 80 epochs with the AdamW optimizer and the initial learning rate is set to 0.0002. For the LVIS dataset, we follow previous works [9, 14, 64] and fine-tune YOLO-World on the LVIS-base (common & frequent) and evaluate it on the LVIS-novel (rare). In addition, we fine-tune the text encoder on LVIS with a learning factor of 0.01.

COCO Object Detection. We compare the pre-trained YOLO-World with previous YOLO detectors [20, 23, 52] in Tab. 6. For fine-tuning YOLO-World on the COCO dataset, we remove the proposed RepVL-PAN for further acceleration considering that the vocabulary size of the COCO dataset is small. In Tab. 6, it's evident that our approach can achieve decent zero-shot performance on the COCO dataset, which indicates that YOLO-World has strong generalization ability. Moreover, YOLO-World after fine-tuning 80 epochs on the COCO `train2017` demonstrates higher performance compared to previous methods trained from scratch with amounts of epochs ($\geq$ 300 epochs).

LVIS Object Detection. In Tab. 7, we evaluate the fine-tuning performance of YOLO-World on the standard LVIS dataset. Firstly, compared to the oracle YOLOv8 [20]

Method	Backbone	Params	Pre-trained Data	FPS	AP	AP_r	AP_c	AP_f
MDETR [21]	R-101 [16]	169M	GoldG	-	24.2	20.9	24.3	24.2
GLIP-T [24]	Swin-T [32]	232M	O365,GoldG	0.12	24.9	17.7	19.5	31.0
GLIP-T [24]	Swin-T [32]	232M	O365,GoldG,Cap4M	0.12	26.0	20.8	21.4	31.0
GLIPv2-T [60]	Swin-T [32]	232M	O365,GoldG	0.12	26.9	-	-	-
GLIPv2-T [60]	Swin-T [32]	232M	O365,GoldG,Cap4M	0.12	29.0	-	-	-
Grounding DINO-T [30]	Swin-T [32]	172M	O365,GoldG	1.5	25.6	14.4	19.6	32.2
Grounding DINO-T [30]	Swin-T [32]	172M	O365,GoldG,Cap4M	1.5	27.4	18.1	23.3	32.7
DetCLIP-T [56]	Swin-T [32]	155M	O365,GoldG	2.3	34.4	26.9	33.9	36.3
YOLO-World-S	YOLOv8-S	13M (77M)	O365,GoldG	74.1 (19.9)	26.2	19.1	23.6	29.8
YOLO-World-M	YOLOv8-M	29M (92M)	O365,GoldG	58.1 (18.5)	31.0	23.8	29.2	33.9
YOLO-World-L	YOLOv8-L	48M (110M)	O365,GoldG	52.0 (17.6)	35.0	27.1	32.8	38.3
YOLO-World-L	YOLOv8-L	48M (110M)	O365,GoldG,CC3M-Lite	52.0 (17.6)	35.4	27.6	34.1	38.0

Table 2. **Zero-shot Evaluation on LVIS.** We evaluate YOLO-World on LVIS `minival` [21] in a zero-shot manner. We report the *Fixed AP* [4] for a fair comparison with recent methods. The FPS is evaluated on one NVIDIA V100 GPU w/o TensorRT. The parameters and FPS of YOLO-World are evaluated for both the re-parameterized version (w/o bracket) and the original version (w/ bracket).

Pre-trained Data	AP	AP_r	AP_c	AP_f
O365	23.5	16.2	21.1	27.0
O365,GQA	31.9	22.5	29.9	35.4
O365,GoldG	32.5	22.3	30.6	**36.0**
O365,GoldG,CC3M-Lite	**33.0**	**23.6**	**32.0**	35.5

Table 3. **Ablations on Pre-training Data.** We evaluate the zero-shot performance on LVIS of pre-training YOLO-World with different amounts of data.

GQA	T→I	I→T	AP	AP_r	AP_c	AP_f	Δt(ms)
✗	✗	✗	22.4	14.5	20.1	26.0	+0.0
✗	✓	✗	23.2	15.2	20.6	27.0	+1.5
✗	✓	✓	23.5	16.2	21.1	27.0	+1.9
✓	✗	✗	29.7	21.0	27.1	33.6	+0.0
✓	✓	✓	**31.9**	**22.5**	**29.9**	**35.4**	+1.9

Table 4. **Ablations on Re-parameterizable Vision-Language Path Aggregation Network.** We evaluate the zero-shot performance on LVIS of the proposed Vision-Language Path Aggregation Network. T→I and I→T denote the Text-guided CSPLayers and Image-Pooling Attention, respectively.

Text Encoder	Frozen?	AP	AP_r	AP_c	AP_f
BERT-base	Frozen	14.6	3.4	10.7	20.0
BERT-base	Fine-tune	18.3	6.6	14.6	23.6
CLIP-base	Frozen	**22.4**	**14.5**	**20.1**	**26.0**
CLIP-base	Fine-tune	19.3	8.6	15.7	24.8

Table 5. **Text Encoder in YOLO-World.** We ablate different text encoders in YOLO-World through the zero-shot LVIS evaluation.

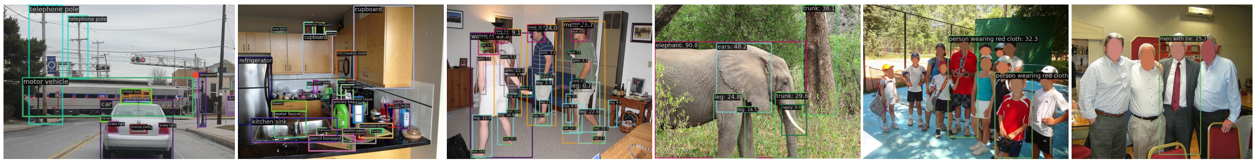
trained on the full LVIS datasets, YOLO-World achieves significant improvements, especially for larger models, *e.g.*, YOLO-World-L outperforms YOLOv8-L by 7.2 AP and 10.2 AP_r. improvements can demonstrate the effectiveness of the proposed pre-training strategy for large-vocabulary detection. eover, YOLO-World, as an efficient one-stage detector, outperforms previous state-of-the-art two-stage methods [9, 14, 22, 53, 64] on the overall per-

Method	Pre-train	Epochs	AP	AP_{50}	AP_{75}	FPS
Training from scratch.						
YOLOv6-S [23]	✗	300	43.7	60.8	47.0	442
YOLOv6-M [23]	✗	300	48.4	65.7	52.7	277
YOLOv6-L [23]	✗	300	50.7	68.1	54.8	166
YOLOv7-T [52]	✗	300	37.5	55.8	40.2	404
YOLOv7-L [52]	✗	300	50.9	69.3	55.3	182
YOLOv7-X [52]	✗	300	52.6	**70.6**	57.3	131
YOLOv8-S [20]	✗	500	44.4	61.2	48.1	386
YOLOv8-M [20]	✗	500	50.5	67.3	55.0	238
YOLOv8-L [20]	✗	500	52.9	69.9	57.7	159
Zero-shot transfer.						
YOLO-World-S	O+G	0	37.6	52.3	40.7	-
YOLO-World-M	O+G	0	42.8	58.3	46.4	-
YOLO-World-L	O+G	0	44.4	59.8	48.3	-
YOLO-World-L	O+G+C	0	45.1	60.7	48.9	-
Fine-tuned w/ RepVL-PAN.						
YOLO-World-S	O+G	80	45.9	62.3	50.1	-
YOLO-World-M	O+G	80	51.2	68.1	55.9	-
YOLO-World-L	O+G+C	80	53.3	70.1	58.2	-
Fine-tuned w/o RepVL-PAN.						
YOLO-World-S	O+G	80	45.7	62.3	49.9	373
YOLO-World-M	O+G	80	50.7	67.2	55.1	231
YOLO-World-L	O+G+C	80	**53.3**	70.3	**58.1**	156

Table 6. **Comparison with YOLOs on COCO Object Detection.** We fine-tune the YOLO-World on COCO `train2017` and evaluate on COCO `val2017`. The results of YOLOv7 [52] and YOLOv8 [20] are obtained from MMYOLO [3]. 'O', 'G', and 'C' denote pertaining using Objects365, GoldG, and CC3M†, respectively. The FPS is measured on one NVIDIA V100 w/ TensorRT.

formance without extra designs, *e.g.*, learnable prompts [9] or region-based alginments [14].

n-Vocabulary Instance Segmentation In this section, we further fine-tune YOLO-World for segmenting objects under the open-vocabulary setting Considering that YOLO-World has strong transfer and generalization capabilities,



(a) LVIS Vocabulary (b) User's Vocabulary (c) Referring Detection

Figure 5. **Visualization Results of YOLO-World.** We visualize the outputs of YOLO-World under three settings: (a) zero-shot inference on LVIS, (b) detection with user prompts (categories), *e.g.*, the parts of humans, and (c) referring detection, *e.g.*, men with a tie. Please zoom in to see detailed labels.

Method	AP	AP_r	AP_c	AP_f
ViLD [14]	27.8	16.7	26.5	34.2
RegionCLIP [63]	28.2	17.1	-	-
Detic [64]	26.8	17.8	-	-
FVLM [22]	24.2	18.6	-	-
DetPro [9]	28.4	20.8	27.8	32.4
BARON [53]	29.5	**23.2**	29.3	32.5
YOLOv8-M	23.1	8.4	21.3	31.5
YOLOv8-L	26.9	10.2	25.4	35.8
YOLO-World-M	28.8	15.9	24.6	39.0
YOLO-World-L	**34.1**	20.4	**31.1**	**43.5**

Table 7. **Comparison with Open-Vocabulary Detectors on LVIS.** We train YOLO-World on the LVIS-base (including common and frequent) report the *bbox AP*. The YOLOv8 are trained on the full LVIS datasets (including base and novel) along with the class balanced sampling.

we directly fine-tune YOLO-World on a subset of data with mask annotations and evaluate the segmentation performance under large-vocabulary settings. cifically, we benchmark open-vocabulary instance segmentation of YOLO-World on the standard LVIS dataset and fine-tune on LVIS-base (comprising 866 categories) and COCO dataset (comprising 80 categories). . 8 shows the experimental results of extending YOLO-World for open-vocabulary instance segmentation. Specifically, we adopt two fine-tuning strategies: (1) fine-tuning the segmentation head and (2) fine-tuning all modules. Under strategy (1), YOLO-World still retains the zero-shot capabilities acquired from pre-training, allowing it to generalize to unseen categories. Strategy (2) enables YOLO-World to fit the LVIS dataset better while it may result in the degradation of the zero-shot capabilities. In addition, Tab. 8 shows fine-tuning on LVIS-base obtains better performance compared to that based on COCO. However, the ratios between AP and AP_r (AP_r/AP) are nearly unchanged, *e.g.*, the ratios on COCO and LVIS-base are 76.5% and 74.3%. Considering that the detector is frozen, we attribute the performance gap to the fact that the LVIS dataset provides dense segmentation annotations, which are beneficial for learning the segmentation head.

4.5. Visualizations

Fig. 5 shows the visualization results of pre-trained YOLO-World-L under three settings: (a) we perform zero-shot inference with LVIS categories; (b) we input the custom prompts with fine-grained categories, *e.g.*, head, leg, and arm; (c) referring detection. The visualizations also demonstrate that YOLO-World has a strong generalization ability for open-vocabulary scenarios along with referring ability.

Model	Module	Data	AP	AP_r	AP_c	AP_f
YOLO-World-M	*Seg Head*	COCO	12.3	9.1	10.9	14.6
YOLO-World-L	*Seg Head*	COCO	16.2	12.4	15.0	19.2
YOLO-World-M	*Seg Head*	LVIS	16.7	12.6	14.6	20.8
YOLO-World-L	*Seg Head*	LVIS	19.1	14.2	17.2	23.5
YOLO-World-M	*All*	LVIS	25.9	13.4	24.9	32.6
YOLO-World-L	*All*	LVIS	**28.7**	**15.0**	**28.3**	**35.2**

Table 8. **Open-Vocabulary Instance Segmentation.** We evaluate YOLO-World for open-vocabulary instance segmentation under the two settings. We fine-tune the segmentation head or all modules of YOLO-World and report *Mask AP* for comparison.

5. Conclusion

We present YOLO-World, a cutting-edge real-time open-vocabulary detector aiming to improve efficiency and open-vocabulary capability in real-world applications. In this paper, we have reshaped the prevalent YOLOs as a vision-language YOLO architecture for open-vocabulary pre-training and detection and proposed RepVL-PAN, which connects vision and language information with the network and can be re-parameterized for efficient deployment. We further present the effective pre-training schemes with detection, grounding and image-text data to endow YOLO-World with a strong capability for open-vocabulary detection. Experiments can demonstrate the superiority of YOLO-World in terms of speed and open-vocabulary performance and indicate the effectiveness of vision-language pre-training on small models, which is insightful for future research. We hope YOLO-World can serve as a new benchmark for addressing real-world open-vocabulary detection.

Acknowledgement. This work was partially supported by National Natural Science Foundation of China (NSFC) under Grant No. 62276108.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 2

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[3] MMYOLO Contributors. MMYOLO: OpenMMLab YOLO series toolbox and benchmark. https://github.com/open-mmlab/mmyolo, 2022. 5, 7

[4] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross B. Girshick. Evaluating large-vocabulary object detectors: The devil is in the details. *CoRR*, abs/2102.01066, 2021. 1, 6, 7

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 3

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 1, 4, 6

[7] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[9] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14064–14073, 2022. 1, 3, 6, 7, 8

[10] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R. Scott, and Weilin Huang. TOOD: task-aligned one-stage object detection. In *ICCV*, pages 3490–3499. IEEE, 2021. 5

[11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 3

[12] Ross B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015. 2

[13] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 2

[14] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. 1, 3, 6, 7, 8

[15] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 5

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 1, 2

[18] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 5

[19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916, 2021. 1, 3

[20] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8. https://github.com/ultralytics/ultralytics, 2023. 2, 3, 4, 5, 6, 7

[21] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - modulated detection for end-to-end multi-modal understanding. In *ICCV*, pages 1760–1770, 2021. 1, 5, 6, 7

[22] Weicheng Kuo, Yin Cui, Xiuye Gu, A. J. Piergiovanni, and Anelia Angelova. F-VLM: open-vocabulary object detection upon frozen vision and language models. *CoRR*, abs/2209.15639, 2022. 7, 8

[23] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, and Xiaolin Wei. Yolov6: A single-stage object detection framework for industrial applications. *CoRR*, abs/2209.02976, 2022. 2, 3, 6, 7

[24] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, pages 10955–10965, 2022. 1, 2, 3, 5, 6, 7

[25] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2023. 3

[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 1, 2

[27] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017. 1, 2

[28] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017. 2

[29] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 3, 4

[30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. *CoRR*, abs/2303.05499, 2023. 1, 2, 3, 6, 7

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016. 2

[32] 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 *ICCV*, pages 9992–10002, 2021. 2, 3, 6, 7

[33] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, and Shilei Wen. PP-YOLO: an effective and efficient implementation of object detector. *CoRR*, abs/2007.12099, 2020. 3

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[35] Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. In *ECCV*, 2022. 3

[36] Matthias Minderer, Alexey A. Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *NeurIPS*, 2023. 3

[37] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, pages 1143–1151, 2011. 6

[38] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *Int. J. Comput. Vis.*, pages 74–93, 2017. 5

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 2, 3, 4, 5, 6

[40] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, pages 6517–6525, 2017. 3

[41] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 2

[42] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 2, 3

[43] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1, 2, 4

[44] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1137–1149, 2017. 2

[45] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1137–1149, 2017. 1

[46] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pages 8429–8438, 2019. 1, 2, 5

[47] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018. 5

[48] Cheng Shi and Sibei Yang. Edadet: Open-vocabulary object detection using early dense alignment. In *ICCV*, pages 15678–15688, 2023. 1, 3

[49] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *ICCV*, pages 9626–9635, 2019. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 3

[51] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of CNN. In *CVPRW*, pages 1571–1580, 2020. 3

[52] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *CVPR*, pages 7464–7475, 2023. 2, 3, 6, 7

[53] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, pages 15254–15264, 2023. 1, 3, 7, 8

[54] Johnathan Xie and Shuai Zheng. ZSD-YOLO: zero-shot YOLO detection using vision-language knowledgedistillation. *CoRR*, 2021. 3

[55] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, and Baohua Lai. PP-YOLOE: an evolved version of YOLO. *CoRR*, abs/2203.16250, 2022. 3

[56] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. In *NeurIPS*, 2022. 1, 2, 3, 6, 7

[57] Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *CVPR*, pages 23497–23506, 2023. 1, 3, 6

[58] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary DETR with conditional matching. In *ECCV*, pages 106–122, 2022. 3

[59] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pages 14393–14402, 2021. 1, 3

[60] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *NeurIPS*, 2022. 1, 2, 3, 6, 7

[61] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023. 3

[62] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pages 9756–9765, 2020. 2, 3

[63] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *CVPR*, pages 16772–16782, 2022. 3, 8

[64] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, pages 350–368, 2022. 3, 6, 7, 8

[65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. 3