Enhanced Object Detection: A Study on Vast Vocabulary Object Detection Track for V3Det Challenge 2024

Peixi Wu
University of Science and Technology of China
wupeixi@mail.ustc.edu.cn

Bosong Chai*
Zhejiang University
chaibosong@mail.zju.edu.cn

Xuan Nie
Northwestern Polytechnical University
xnie@nwpu.edu.cn

Longquan Yan
Northwest University
18829512640@163.com

Zeyu Wang
Zhejiang University
wangzeyu2020@zju.edu.cn

Qifan Zhou
Northwestern Polytechnical University
george13@mail.nwpu.edu.cn

Boning Wang
Zhejiang University
1007658022@qq.com

Abstract

In this technical report, we present our findings from the research conducted on the Vast Vocabulary Visual Detection (V3Det) dataset for Supervised Vast Vocabulary Visual Detection task. How to deal with complex categories and detection boxes has become a difficulty in this track. The original supervised detector is not suitable for this task. We have designed a series of improvements, including adjustments to the network structure, changes to the loss function, and design of training strategies. Our model has shown improvement over the baseline and achieved excellent rankings on the Leaderboard for both the Vast Vocabulary Object Detection (Supervised) track and the Open Vocabulary Object Detection (OVD) track of the V3Det Challenge 2024.

1. Introduction

The V3Det dataset [38] is a large-scale, richly annotated dataset featuring detection bounding box annotations for over 13,000 object classes on real images. It includes a hierarchical category structure with detailed class affiliations forming a comprehensive relationship tree. As shown in Fig 1, with 245,000 annotated images and expert-generated descriptions, V3Det is an invaluable resource for advanced object detection research in computer vision.

This workshop has two tracks. The first track (Supervised), called Vast Vocabulary Object Detection, aims to evaluate supervised learning models for object detection across all 13,204 classes in the V3Det dataset. Detecting any object has been a long-term goal in the field of computer vision. Due to the countless diverse objects in the real world, an ideal visual detection system should be capable of detecting a large number of categories and be applicable to open vocabulary categories.

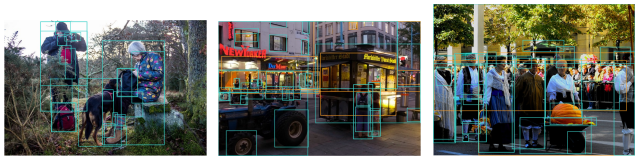

Figure 1: V3Det is a high-quality, precisely annotated object detection dataset with a broad vocabulary, encompassing 13,204 categories. The figure shows annotated image samples from V3Det, featuring more complex and detailed annotations.

Currently widely used object detection datasets such as COCO [23], Objects365 [32], and OpenImages v4 [19], despite providing a large number of images and categories, still have a limited vocabulary. The limited vocabulary of these datasets constrains the training potential of class-generalized detectors, as an ideal detector should be able to recognize new categories beyond those in the training set. Even large vocabulary object detection datasets like LVIS [16] cannot fully represent the complexity of the real world in terms of the number and diversity of categories. V3Det provides the research community with a large vo-

*Bosong Chai is the corresponding author. Bosong Chai and Peixi Wu contributed equally to this work.

cabulary object detection dataset, which can accelerate the exploration of more general visual detection systems. The baseline cascade structure is very suitable for handling the hierarchical category structure of the V3Det dataset. We treat the supervised track I as a traditional object detection task with complex labels, using common detection improvement strategies. By improving the Feature Pyramid Network (FPN) structure, we hope the network can effectively learn deeper semantic information. Additionally, we balance category labels by adjusting the loss function.

The second track (OVD) of the V3Det challenge involves developing object detectors capable of accurately identifying objects from 6,709 base classes and 6,495 novel classes. For base classes, full annotations are provided, while for novel classes, only class names, descriptions, and a few exemplar images are given. The task is to design detectors that can utilize this limited information to detect novel classes effectively during inference, ensuring accurate detection across both base and novel categories. This track requires detectors to possess strong generalization and semantic understanding capabilities to identify new categories without direct annotation information. It can rely on current vision-text models, such as CLIP [29], to extract visual and semantic features from images and text, and establish connections between them.

The baseline EVA model [12], combined with CLIP [29], demonstrates powerful semantic feature extraction capabilities. Due to time constraints and limited computational resources, we rely solely on supervised training for Track II, yet still achieve good detection results even for novel categories. This to some extent indicates that V3Det dataset covers a vast array of annotations from real-world scenarios, with rich semantic information learned by excellent detectors, thus exhibiting good generalization performance.

2. Related Work

2.1. Object Detection

Object detection [13, 14, 2] is one of the most traditional tasks in computer vision, with various applications across different industries such as autonomous driving [41, 39, 28], robotics [9], remote sensing [4]. It takes images as input, localizes, and classifies objects within a given vocabulary. Each detected object is represented by a bounding box with a class label.

Classical CNN-based object detectors can be divided into two main categories: two-stage and one-stage detectors. Two-stage detectors [13, 14, 2, 47] first generate object proposals and then refine them in a second stage, offering higher precision but at the cost of increased complexity. One-stage detectors, such as YOLO [30, 15, 37, 36] and SSD[25], directly classify and regress predefined anchor boxes or search for geometric cues like points [35], centers [11], and corners [20], providing faster but potentially less accurate results. Transformer-based detectors [33, 17, 3, 48] use the self-attention mechanism to capture global contextual information in images, eliminating the need for additional components like anchor boxes and Non-Maximum Suppression (NMS). The end-to-end architecture is simpler, making the training and inference process more straightforward.

Currently, novel detectors based on diffusion are emerging [6, 7]. At the same time, object detection is being combined with large language models (LLM) to achieve open-vocabulary detection [42, 8, 40] and the detection of everything. This approach allows object detection to go beyond just the design of detector architectures, providing models with better adaptability to handle complex scenes and various types of objects.

2.2. Data Augmentation

Data augmentation is a commonly used technique in machine learning and deep learning, aimed at transforming and expanding training data to increase its diversity and richness. In addition to common data augmentation methods such as flipping, jittering, and scaling, effective data augmentation techniques for object detection can be broadly categorized into Cutting-based [46, 10] and Mixing-based [44, 43, 18] methods. There is also the widely used Mosaic method proposed by YOLOv4 [1].

3. Our Method

In this section, we elaborate on the technical details of our method. We made two improvements based on the baseline: (a) adjustments to the model architecture, (b) improvements to the loss function and training strategy. We will introduce each component in the following subsections.

3.1. Baseline Framework

In this challenge, the organizers built two baselines based on MMDetection [5][1] and Detectron2[2]. The baseline EVA[3], based on Detectron2, utilizes a Cascade RCNN with a backbone structure of ViTDet [22]. The pretraining task of EVA involves Masked Image Modeling (MIM), aimed at reconstructing masked image-text aligned visual features generated by the CLIP [29]. This network demonstrates robust generalization performance and stands as the state-of-the-art (SOTA) for many vision tasks. Based on the MMDetection baseline[4], the best-performing model is also based on Cascade R-CNN [2], with a Swin-Transformer [26] as its backbone. The cascade structure is highly suitable for

[1] https://github.com/open-mmlab/mmdetection
[2] https://github.com/facebookresearch/detectron2
[3] Detectron2-V3Det-EVA
[4] MMDetection-V3Det

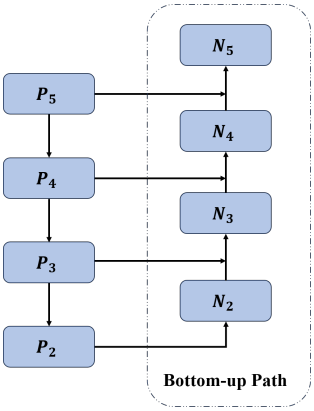

Figure 2: Illustration of PA-FPN structure, FPN with bottom-up path structure from N_2 to N_5.

multi-class detection tasks by progressively refining bounding boxes and classification results. Each stage of the cascade head uses two shared fully connected layers, which helps capture high-level semantic features of the targets at different stages. The IoU thresholds set for each stage ensure that the detection boxes become more precise at each level.

3.2. Model Architecture Adjustment

Backbone. The baseline adopts Swin Transformer [26] as the backbone network for feature extraction, commonly using versions such as Swin-S, Swin-B, and Swin-L[5]. Different versions affect the parameter count, computational cost, and accuracy. Therefore, we have made multiple attempts with different backbones. The baseline pretrained model provided by the organizers uses ImageNet-1K pretrained weights to initialize the backbone. We also attempted to use ImageNet-22K pretrained weights to initialize the Swin-B backbone. We also attempted to use pretrained models with a resolution of 384×384[6]. In addition to using Swin Transformer as the backbone, we also experimented with the basic Vision Transformer models, specifically using ViT-B and ViT-L.

Path Aggregation Feature Pyramid Network (PA-FPN). Although the FPN structure already integrates shallow feature information, the path from shallow features to the top network layers is too long, resulting in low utilization efficiency of shallow features. To effectively capture image semantic information, inspired by PA-Net [24], we add a bottom-up structure into the baseline Cascade R-CNN. This shortens the transmission path from shallow features to the top layers, enhancing the transmission of shallow features within the network, and allowing more shallow features to be effectively utilized. As shown in Fig 2, where the feature map N_2 has the same dimensions as P_2. N_3, N_4, N_5 are obtained through downsampling and fusion. For a high-resolution feature map N_i and a low-resolution feature map P_{i+1}, a new feature map N_{i+1} is generated.

[5]https://github.com/microsoft/Swin-Transformer

[6]swin_base_patch4_window12_384_22k.pth

3.3. Other Improvements

Data Augmentation. In order to enhance the size and quality of training dataset, we employ data augmentation including flipping, jittering, and scaling, on original input images. We tried the data augmentation strategies built into MMDetection-transforms such as Mixup, Cutout, Corrupt, and PhotoMetricDistortion. It is important to note that more data augmentation is not always better, especially in object detection tasks. Excessive data augmentation can lead to shifts or distortions in the original target positions, making it difficult for the model to learn accurate target boundaries. It has been shown [34] that the two-stage algorithm can be used for data augmentation without random geometric transformations in the training phase.

Loss Function. In this section, we introduce the DIoU Loss function for addressing coordinate point interrelationship issues using the L_1 loss function in baseline Cascade R-CNN networks. Inspired by Zhaohui Zheng et al. [45], DIoU Loss considers two key issues: (a) Minimizing the normalized distance between the prediction frame and the target frame to achieve faster convergence. (b) How to make the regression more accurate and faster when there is overlap or even inclusion with the target box. The DIoU Loss function yields values in the range [-1,1], and is defined as follows:

$$R_{DIoU} = \frac{\rho^2(b, b^{gt})}{c^2}, \tag{1}$$

$$L_{DIoU} = 1 - IoU + R_{DIoU}, \tag{2}$$

$\rho(\cdot)$ represents the Euclidean distance. The penalty term R_{DIoU} is defined as the squared Euclidean distance between the central points of b and b_{gt}, normalized by the square of the diagonal length c of the smallest enclosing box covering the two boxes. This formulation ensures that the DIoU loss directly minimizes the distance between the two central points.

Inspired by Li et al. [21], to reduce the economic imbalance of the sample measure in the detection process and the inaccurate detection results caused by the blurred bounding box, we properly introduces the Generalized Focal Loss (GFL) function into the Region Proposal Network (RPN) to balance the proportion of positive and negative samples in the loss function, The GFL function is typically shown in equation (3).

Table 1: The detection results of different models on the V3Det Supervised track I. We show the results on the validation set, with gray indicating the baseline provided by the organizers in MMDetection. The last row represents the baseline provided in Detectron2, which uses EVA pretrained with CLIP. $pretrain$ indicates whether the models are pretrained on the ImageNet 1K or the ImageNet 22K dataset, and $resolution$ indicates whether an input resolution of 224×224 or 384×384 was used. All models are based on Cascade R-CNN.

Backbone	pretrain	resolution	AP_{all}	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	Recall_s	Recall_m	Recall_l	Recall_all
Swin-B	1K	224	43.4	50.2	45.8	12.9	22.9	49.0	23.3	37.0	70.1	64.3
Swin-B	22K	224	43.2	50.4	46.0	13.7	23.7	49.3	30.6	44.8	75.3	69.5
Swin-B	22K	384	43.7	50.6	46.3	13.7	23.9	49.5	30.8	45.1	75.5	69.8
Swin-L	22K	224	37.1	43.4	38.3	8.6	16.3	35.2	24.4	38.2	64.5	58.5
ViT-B	22K	384	40.2	46.6	43.3	10.2	19.5	40.2	30.8	40.7	68.1	69.8
ViT-L	22K	224	30.1	35.9	32.7	9.8	17.1	35.5	23.4	37.4	70.3	64.2
Cascade R-CNN EVA-CLIP			51.1	55.9	53.2	24.4	34.6	56.2	44.3	56.2	78.6	75.3

$$GFL\left(p_{y_l}, p_{y_r}\right) = -\left|y - \left(y_l p_{y_l} + y_r p_{y_r}\right)\right|^{\beta} \times \left(\left(y_r - y\right)\log\left(p_{y_l}\right) + \left(y - y_l\right)\log\left(p_{y_r}\right)\right). \tag{3}$$

y represents the true IoU, while y_l and y_r are the lower and upper bounds of the predicted and true IoU of the bounding boxes. β is an adjustable hyper-parameter controlling the slope of the loss function ($\beta \geq 0$). p_{yl} and p_{yr} are the probability values predicted by the model, satisfying $p_{y_l} + p_{y_r} = 1$. The final prediction $\hat{y}$ is a linear combination of y_l and y_r, enabling classification values to transition from discrete to continuous. The balancing factor in the formula minimizes deviations between predicted and true IoU, while the classification loss function computes errors to enhance the model's understanding of object position and size. GFL employs a focal mechanism, dynamically adjusting weights to balance proportions and facilitate learning differences between positive and negative samples.

Table 2: Detection results of different methods for Supervised track I in the validation set

Method	AP_{all}	Recall_all
Baseline	43.4	64.3
+PA-FPN	42.2	62.6
+DIoU	44.7	69.3
+GFL	43.7	68.4

Table 3: Detection results on the test set for OVD Track II, where n represents novel categories and b represents base categories.

bAP_{50}	nAP_{50}	bAP_{75}	nAP_{75}	bAP	nAP	AP
56.2	28.7	53.2	2.2	50.4	10.3	20.2

Training Techniques. During training, we find that the $json$ format files of more than 30 images in the original dataset do not match the corresponding images. We perform data cleaning and remove such erroneous data. We use Synchronized Batch Normalization to solve the multiple GPU cross-card synchronization problem. For the learning rate setting, we borrowed the training strategy of YOLOv3 [31], and in the first 3000 iterations, we use warm-up to gradually increase the learning rate from 0 to the preset base learning rate, and subsequent iterations with the cosine strategy, which is conducive to the stability of the training process. We use Apex-based hybrid precision training to accelerate the training with as little loss of precision as possible. We also enable auto-scale learning rate, which means that when using different numbers of GPUs and different batch sizes, GPU resources can be effectively utilized and the model can converge quickly.

4. Experiments

In this section, we present the implementation details and give main experimental results and analysis.

4.1. Implementation Details

Following the challenge guidelines, 183,354 images are used as the training set, and 29,821 images are used as the validation set. We train exclusively on the V3Det dataset and do not use any extra data. We train the full models on the training set and evaluate them on the validation set for algorithm validation and hyper-parameter tuning. Finally, we retrain and save the models on the complete training data using the selected hyper-parameters. We implement our model using PyTorch 2.1.0 and conduct our experiments on a system with 4 × H100 GPUs, using a batch size of 48. We use Adam with decoupled weight decay (AdamW) [27] with a learning rate of 0.001. We adopt the COCO Detection Evaluation [23] to measure the performance. The COCO Detection Evaluation includes multiple-scale objects (AP_S, AP_L), where AP_S represents small ob-

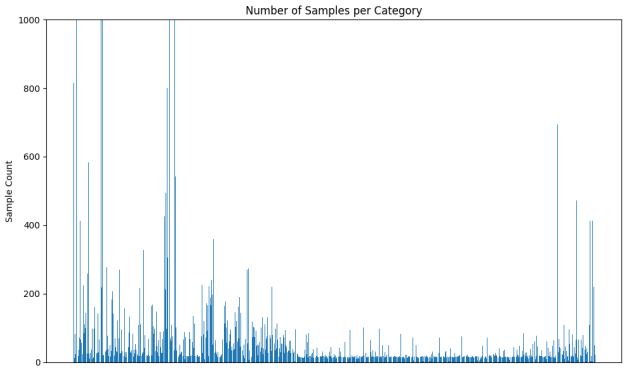

Figure 3: The horizontal axis represents different classes, and the vertical axis represents the number of samples corresponding to each class, with values above 1000 not displayed.

ject AP, with an area $< 32^2$, and AP_L represents large object AP, with an area $> 96^2$. For the Supervised Track I, AP and Recall are used as evaluation metrics for the test set. For the OVD Track II, AP and Recall are calculated separately for the base categories and novel categories.

4.2. Results and Analysis

As shown in Table 1, we are trying various approaches to the model backbone. When using ImageNet 22k pretraining, there is not much change in the AP value of the model, but the Recall has significantly improved. The Recall_all has increased from 64.3% to 69.5%, indicating that the model misses fewer targets. Better pretraining initialization of the backbone is particularly important for object detection tasks. Using a larger model like Swin-L as the backbone introduces additional parameters and computational complexity, resulting in longer inference times. However, despite these drawbacks, the detection performance of the model decreased.

As shown in Table 2, we introduced a series of improvements, including optimizing the loss function of the original detector and modifying its FPN structure. Surprisingly, after incorporating the PA-FPN structure, the model's detection performance, as measured by AP, did not improve but instead decreased by nearly 2%. The PA-FPN structure has been proven effective in many tasks and widely applied in various detection and segmentation tasks. We speculate that this unexpected result may be due to the influence of noise or irrelevant information on the lower-level features, leading to a decrease in the quality of the fused features. The bottom-up structure may cause premature or excessive fusion of features between different levels, resulting in information loss or confusion. The introduction of the bottom-up structure may increase the complexity of the network, making training more challenging and requiring more adjustments and optimizations. Due to time constraints, we did not conduct detailed experiments, and further validation will be carried out gradually.

Certainly, modifying the RPN classification loss function to the GFL function and changing the bounding box regression loss to the GIoU loss function have proven effective. As shown in Fig 3, the V3Det dataset, due to its numerous categories, results in poor learning performance for minority classes during training. GFL introduces adjustable parameters to weight the loss functions for different classes, allowing the model to focus more on challenging samples.GFL introduces adjustment parameters to weight the loss functions of different categories, making the model pay more attention to samples that are difficult to classify.

Regrettably, despite conducting numerous experiments and adjustments, and achieving some improvements over the baseline, our results still could not surpass the reproduced EVA model provided by the organizers based on Detectron2. The EVA model employed the MIM training method, optimizing CLIP and demonstrating powerful performance and superior results. The outstanding performance of the EVA model indicates that merely modifying and designing the model structure is no longer sufficient to achieve significant breakthroughs in the current era of large models. The key to the success of the EVA model lies in its innovative training methods and the effective utilization of pretrained models, which provides a direction for our future research and improvements.

As shown in Table 2, for OVD Track II, we adhered to the traditional supervised object detection transfer learning approach and did not incorporate textual information. According to the competition requirements, we used the Cascade R-CNN model based on MMDetection with Swin-B as the backbone from Track I, retrained on the V3Det train set of base classes, and directly inferred on the test dataset. We were pleasantly surprised to find that this approach also yielded good results. Compared to the baseline, our AP for novel classes improved from 11% to 20%, with AP_{50} reaching 29%. This might be because the V3Det dataset already contains rich semantic information, giving the model a certain degree of generalization ability.

5. Conclusion

In conclusion, this report has presented our study on V3Det Challenge for Vast Vocabulary Object Detection track 2024. In the Supervised Track I, we made various attempts at traditional object detection tasks using different models. For the V3Det dataset, which contains rich semantic information across multiple categories, we observed some improvement in detection results. However, although the performance did not fully meet our expectations, our adjustments could not surpass the results we obtained by

reproducing EVA. This indicates that simply modifying and designing model structures is no longer sufficient in the era of LLM. Our final submission achieved good results on the leaderboard for both Track I and Track II.

References

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[4] B. Chai, X. Nie, Q. Zhou, and X. Zhou. Enhanced cascade r-cnn for multi-scale object detection in dense scenes from sar images. *IEEE Sensors Journal*, 2024.

[5] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[6] S. Chen, P. Sun, Y. Song, and P. Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023.

[7] Z. Chen, R. Gao, T.-Z. Xiang, and F. Lin. Diffusion model for camouflaged object detection. *arXiv preprint arXiv:2308.00303*, 2023.

[8] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:2401.17270*, 2024.

[9] A. Coates and A. Y. Ng. Multi-camera object detection for robotics. In *2010 IEEE International conference on robotics and automation*, pages 412–419. IEEE, 2010.

[10] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[12] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[13] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] Q. Guo, J. Liu, and M. Kaliuzhnyi. Yolox-sar: High-precision object detection system based on visible and infrared sensors for sar remote sensing. *IEEE Sensors Journal*, 22(17):17243–17253, 2022.

[16] A. Gupta, P. Dollar, and R. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[17] L. He and S. Todorovic. Destr: Object detection with split transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9377–9386, 2022.

[18] J.-H. Kim, W. Choo, and H. O. Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.

[19] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

[20] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[21] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020.

[22] Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[27] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] J. Mao, S. Shi, X. Wang, and H. Li. 3d object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131(8):1909–1963, 2023.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[30] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. in 2017 ieee conf. *Comput. Vis. Pattern Recognit. CVPR*, pages 6517–6525.

[31] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[32] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[33] Z. Sun, S. Cao, Y. Yang, and K. M. Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3611–3620, 2021.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[35] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: fully convolutional one-stage object detection. corr abs/1904.01355 (2019), 1904.

[36] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[37] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[38] J. Wang, P. Zhang, T. Chu, Y. Cao, Y. Zhou, T. Wu, B. Wang, C. He, and D. Lin. V3det: Vast vocabulary visual detection dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19844–19854, 2023.

[39] Z. Wang, F. Colonnier, J. Zheng, J. Acharya, W. Jiang, and K. Huang. Tirdet: Mono-modality thermal infrared object detection based on prior thermal-to-visible translation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2663–2672, 2023.

[40] Z. Wang, Y. Li, X. Chen, S.-N. Lim, A. Torralba, H. Zhao, and S. Wang. Detecting everything in the open world: Towards universal object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11433–11443, 2023.

[41] H. Wu, C. Wen, W. Li, X. Li, R. Yang, and C. Wang. Transformation-equivariant 3d object detection for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2795–2802, 2023.

[42] X. Wu, F. Zhu, R. Zhao, and H. 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.

[43] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[44] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[45] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.

[46] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

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

[48] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.