CodeComingSoon's Tech Report of Low-light Object Detection and Instance Segmentation Challenge

Haiyang Xie[1,4,*] Jian Zhao[4,*] Shihua Huang[2,*] Peng Cheng[3,4] Xi Shen[2,†]
Zheng Wang[1,†] Shuai An[5] Caizhi Zhu[6] Xuelong Li[4]
[1]School of Computer Science, Wuhan University
[2]Intellindust
[3]Beijing Forestry University
[4]EVOL Lab, Institute of AI (TeleAI), China Telecom
[5]Institute of AI (TeleAI), China Telecom
[6]Harbin Institute of Technology
whuocean@whu.edu.cn shihuahuang95@gmail.com
shenxiluc@gmail.com wangzwhu@whu.edu.cn
pengcheng2022@bjfu.edu.cn

Abstract

We provide an overview of our approach aimed at tackling the challenges of object detection and instance segmentation in the Low-light Object Detection and Instance Segmentation Challenge. Our strategy involves integrating 18 predictions from different models using the Weighted Box Fusion (WBF) technique [10], which yields outstanding performance in object detection. Furthermore, we utilize a RTMDet [8] to compete in the segmentation track.

1. Method

In this section, we present details of our method for both Object Detection track and Instance Segmentation track.

1.1. Object Detection

Several prior studies [3–5, 11] have endeavored to enhance image cognition performance in extreme conditions. Despite demonstrating superior efficacy compared to their respective baselines, we have observed that employing conventional methodologies on the dataset in this challenge yields comparable effectiveness while being straightforward to implement. Consequently, we adopt a simplified approach by treating the low-light images from the challenge dataset as conventional RGB images.

As shown in Fig. 1, we trained several detectors, including RTMDet [8], YOLOX [6], Dino [12] and Co-DETR [13] on the challenge datasets, and then ensemble the predictions from those models to achieve better results. We employed Weighted Box Fusion [10] as our ensemble method.

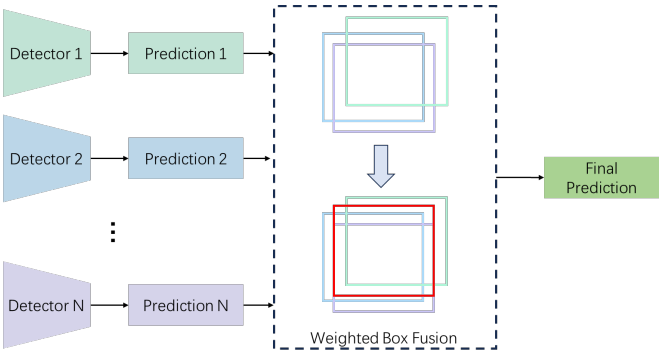

Figure 1. The overview of the proposed object detection framework.we trained several detectors, including RTMDet [8], YOLOX [6], Dino [12] and Co-DETR [13] on the challenge datasets, and then ensemble the predictions from those models to achieve better results. We employed Weighted Box Fusion [10] as our ensemble method.

1.1.1 RTMDet

RTMDet [8] is an efficient real-time object detector that surpasses the YOLO series. Apart from adjusting the number of output classes, we made no modifications to RTMDet. RTMDet-x and RTMDet-l models were chosen due to their high mAP on the COCO dataset.

*Equal Contribution
†Corresponding Author

1.1.2 YOLOX

YOLOX [6] is a highly advanced detector that represents a significant improvement upon the YOLO series. Apart from adjusting the number of output classes, we made no modifications to YOLOX. Taking into account both performance and training costs, we opted for YOLOX-l.

1.1.3 DINO

DINO [12] is an advanced end-to-end object detector. Apart from adjusting the number of output classes, we made no modifications to DINO. DINO-Swin-L model was chosen due to its high mAP on the COCO dataset.

1.1.4 Co-DETR

Co-DETR [13] is a novel training scheme aimed at improving the efficiency and effectiveness of DETR-based detectors. Apart from adjusting the number of output classes, we made no modifications

1.2. Instance Segmentation

We trained a single RTMDet [8] model for instance segmentation without employing any ensemble methods.

2. Implement Details

2.1. Dataset

We solely utilized the challenge dataset for training. Additionally, we attempted to augment our training data by incorporating the COCO dataset which was unprocessed according to [1], preserving annotations with common classes. However, this augmentation did not yield improved results. It is necessary to point out that we still utilized the pretrained weights on the unprocessed [1] COCO dataset to initialize some of the models, aiming to enhance the diversity of our model zoo, which proves advantageous for ensemble methods.

During the initial phase of the challenge, only annotations for the training set were available. Initially, we randomly divided the training set into a proxy training set and a validation set using an 8:2 ratio. Subsequently, we trained the models and optimized the training settings to enhance performance. These settings were then uniformly applied for training on the original complete training dataset, ensuring full utilization of the available data.

2.2. Training

To achieve higher performance, we initialized the model weights using pretrained weights from the COCO [7] Dataset. However, Co-DETR [13] was an exception, as we found that the pretrained weights obtained by training first on Object365 [9] and then on COCO [7] performed better than those from COCO [7].

During the validation and test phases, we retained the weights from the last epoch for evaluation on the official validation and test sets.

We utilized the MMDetection framework [2] to conduct all experiments on 4 machines, each equipped with 8 NVIDIA RTX 3090/4090 GPUs.

Due to the extensive nature of our training process, which involved training over 18 models for ensemble, providing detailed training configurations in this paper may not be feasible. We recommend referring to the config files in our code repository for more comprehensive information.

2.3. Ensemble

Table. 1 briefly describes the type of model and any specific strategies employed. For example, "Dino-Swin-L" signifies the use of the Dino model with the Swin-L Backbone, while "Dino-Swin-L with TTA" indicates the same model enhanced by test-time augmentation (TTA). Additionally, the descriptions encompass different versions of the RTMDet and Co-DETR models, which may incorporate varying parameters like dropout rates or random seeds during the training phase. "obj2coco" indicates that we use pretrained weights obtained by training first on Object365 [9] and then on COCO [7] to initialize the parameters of the model.

These predictions were then utilized in the weighted box fusion to ensemble predictions. The weight of each prediction was determined using a grid search algorithm on the proxy validation set described in 2.1.

For more details about the ensemble process, please refer to the configuration files in our code project.

3. Results

As shown in Table2, our ensemble method for the Object Detection track attained a mean Average Precision (mAP) of 0.76. Additionally, our RTMDet model for the Instance Segmentation track achieved an mAP of 0.58.

4. Conclusion

In this paper, we provide an overview of our approach aimed at tackling the challenges of object detection and instance segmentation in the Low-light Object Detection and Instance Segmentation Challenge. Our Object Detection strategy involves integrating 18 predictions from different models using the Weighted Box Fusion (WBF) technique [10], which yields outstanding performance in the challenge.

References

[1] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images

Table 1. Ensemble Strategy. "Dino-Swin-L" signifies the use of the Dino model with the Swin-L Backbone, while "Dino-Swin-L with TTA" indicates the same model enhanced by test-time augmentation (TTA). Additionally, the descriptions encompass different versions of the RTMDet and Co-DETR models, which may incorporate varying parameters like dropout rates or random seeds during the training phase. "obj2coco" indicates that we use pretrained weights obtained by training first on Object365 [9] and then on COCO [7] to initialize the parameters of the model.

ID	Weight	Description
1	1	Dino-Swin-L
2	1	Dino-Swin-L with TTA
3	1	RTMDet-l
4	1	RTMDet-l
5	1	RTMDet-l
6	1	RTMDet-l
7	1	RTMDet-l
8	1	RTMDet-l
9	1	RTMDet-x
10	1	RTMDet-l
11	1	RTMDet-l
12	1	RTMDet-l
13	1	RTMDet-l
14	1	YOLOX-l with TTA
15	8	Co-DETR
16	8	Co-DETR
17	10	Co-DETR-dropout0.6-obj2coco
18	10	Co-DETR-dropout0.3

Table 2. Results of our methods

	mAP	mAP50
Object Detection	0.76	0.89
Instance Segmentation	0.58	0.79

for learned raw denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11036–11045, 2019. 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. 2

[3] Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *International Journal of Computer Vision*, 131(8):2198–2218, 2023. 1

[4] Linwei Chen, Ying Fu, Shaodi You, and Hongzhe Liu. Hybrid supervised instance segmentation by learning label noise suppression. *Neurocomputing*, 496:131–146, 2022. 1

[5] Ying Fu, Yang Hong, Linwei Chen, and Shaodi You. Le-gan: Unsupervised low-light image enhancement network using attention module and identity invariant loss. *Knowledge-Based Systems*, 240:108010, 2022. 1

[6] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 1, 2

[7] 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. *Computer Vision–ECCV 2014*, 8693:740–755, 2014. 2, 3

[8] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022. 1, 2

[9] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 2, 3

[10] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021. 1, 2

[11] Linwei Chen Ying Fu Yang Hong, Kaixuan Wei. Crafting object detection in very low light. In *BMVC*, 2021. 1

[12] 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. *arXiv preprint arXiv:2203.03605*, 2022. 1, 2

[13] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 1, 2