Road Object Detection Robust to Distorted Objects at the Edge Regions of Images

Wooksu Shin* Donghyuk Choi* Hancheol Park Jeongho Kim
Nota Inc., Republic of Korea
{wooksu.shin,donghyuk.choi,hancheol.park,jeongho.kim}@nota.ai

Abstract

*Fish-eye cameras, capable of capturing wide areas, enable efficient traffic monitoring with only a few cameras. Nevertheless, it still remains challenging to successfully detect objects in images from such cameras. In this work, we analyze key reasons why object detectors frequently make incorrect predictions in such images and propose methods to address them. More specifically, we address the issues of objects being represented as smaller at the edges of images and the distortion of non-target objects (*e.g.*, street signs), which are recognized as target objects (*e.g.*, vehicles). Furthermore, in this work, we propose a road object detector capable of achieving high performance by additionally applying various techniques known to generally enhance detection performance. Our proposed detector achieved second place in Track 4 of the 2024 AI City Challenge with an F1 score of 0.6196. Our code is publicly available at* https://github.com/nota-github/AIC2024_Track4_Nota.

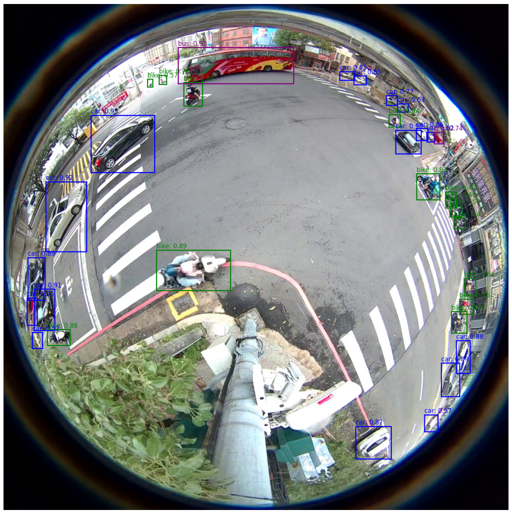


Figure 1. An example of road object detection using fish-eye cameras. The objective of this task is to detect bus, bike, car, pedestrian, and truck objects from images captured by fish-eye cameras.

1. Introduction

Automatic recognition of the locations and categories of key objects on roads (*e.g.*, vehicles and pedestrians) is essential for developing applications that enhance convenience in everyday life, such as smart traffic signal control and real-time traffic congestion analysis. With recent advancements in deep learning-based object detection models, the performance of object detection in images or videos captured by conventional road cameras has been significantly improved. However, despite this progress, there still remains the challenge of requiring a significant number of cameras to monitor roads that span considerably wide areas. To address this issue, as shown in Fig. 1, fish-eye cameras can be utilized to detect road objects in wide areas with fewer cameras. However, utilizing such cameras also poses the problem of distorting the appearance of certain objects, making accurate object detection difficult.

In this work, we discuss various methods for building high-performance road object detection models for images or videos captured by fish-eye cameras. Specifically, we focus on addressing the problems where the state-of-the-art deep learning-based object detectors fail to detect objects in the edge regions of images. We observed that objects are represented smaller at the edge regions of the images. In cases where objects are small, most state-of-the-art object detectors may still fail to detect them. Furthermore, we observed a problem where non-target objects (*e.g.*, street signs) exhibit visual distortions at the edges of the images, resulting in visual similarities with the target objects (*e.g.*, vehicles). As a result, it was observed that non-target objects were frequently detected as target objects.

To address the problem of objects being represented small at the edges of images, we use the sliced inference technique [1]. This involves partitioning the original image into specific-sized slices and resizing each slice to the

[*] These authors contributed equally to this work.

model's input size at the inference step, rather than resizing the original image directly to the model's input size. This approach ensures that objects are either enlarged or not considerably reduced. Additionally, to prevent non-target objects from being detected, pseudo labels are assigned to non-target objects in the training data, enabling the model to learn to distinguish them from the target objects.

Furthermore, in this work, we also use various techniques known to be effective in improving detection performance, regardless of the types of tasks. Experimental results show that the two proposed methods to address issues in the edge regions of images are highly effective in detecting objects on road images captured by fish-eye cameras. Additionally, by applying various techniques together, known to be effective in general, we achieved second place in Track 4 of the 2024 AI City Challenge [20], with an F1 score of 0.6196.

In this paper, our contributions are as follows: (1) We uncover the challenging aspects unique to road object detection tasks based on fish-eye cameras and propose techniques tailored to this task, investigating their effectiveness. (2) We propose a high-performance model for road object detection in fish-eye cameras.

2. Related Work

2.1. Object Detection

It is obvious that using deep learning-based object detectors is crucial for accurately detecting road objects. For a long time, CNN-based networks have been utilized as the architecture of deep learning-based object detection models. Faster R-CNN [14] and You Only Look Once (YOLO) [13] can be considered as conventional examples. Especially, YOLO has been continuously improved from version 1 [13] to the latest version 9 [18], achieving an unprecedented level of high performance, such that it can be used in real-life situations without significant issues. Recently, Transformer-based [17] object detection models, such as DEtection TRansformer (DETR) series [2, 3, 9, 23–25], have been actively proposed and have achieved unprecedented high performance, surpassing CNN-based models.

Conventional CNN-based object detectors predict multiple candidate bounding boxes for an object in an image and then select a final bounding box using non-maximum suppression (NMS). Empirical knowledge is involved in determining candidate bounding boxes and selecting a final bounding box for an object. In contrast, the DETR model [2] performs end-to-end prediction for one object, predicting only one bounding box without our prior knowledge. Moreover, it has been shown to achieve higher performance than CNN-based object detection models, especially for large objects, by capturing relationships between distant pixels through the self-attention mechanism.

Despite various advantages, some weaknesses have been pointed out in the DETR model [2]. First, instead of feature maps of various scales, it uses highly abstracted feature maps, resulting in significant loss of object information. Additionally, during training, only one predicted box per object is considered as a positive bounding box, aiming to minimize the difference with the ground truth box. This leads to insufficient supervision for the encoder to recognize various forms of bounding boxes for one object. Representative structures proposed to address the former issue include Deformable DETR [24] and DETR with Improved deNoising anchOr boxes (DINO) [23]. Various training methods have been proposed to address the latter issue, such as considering diverse positive boxes. Representative models include H-DETR [9], Group DETR [3], and Co-DETR [25]. In this work, we use Co-DETR as the base detector as it has shown the best performance in recent diverse benchmarks.

2.2. Road Object Detection

Various datasets containing plain images captured by conventional cameras have been introduced for the road object detection task [5], and detection techniques tailored to this task have been proposed for a long time [7, 12, 22]. Recently, with the availability of datasets such as FishEye8K benchmark dataset [5], which contain images captured by fish-eye cameras, it has become possible to train and evaluate models for road object detection using images from fish-eye cameras, enabling the development of high-performing models capable of detecting wide areas with few cameras. However, since the datasets have been relatively recently released, there have been relatively few proposed methods specifically tailored to road object detection using images captured by fish-eye cameras. Data augmentation techniques have been introduced to distort plain images to resemble those captured by fish-eye cameras [8]. Nevertheless, specialized methods for this task are still not extensively studied.

3. Proposed Method

In this section, we discuss various methods to successfully detect target objects in road images or videos captured by fish-eye cameras. First, we investigate the task-specific reasons why state-of-the-art deep learning-based detectors frequently make incorrect predictions in this task (§3.1). The key issues are frequently observed at the edge regions of images, and we propose methods to address these task-specific issues. (§3.1). We also discuss various methods known to be effective in general object detection (§3.2). In this work, we improve the performance of detection by ensembling various detectors that combine these individual methods in different ways. In this section, we also elaborate on how we ensemble these detectors (§3.3).

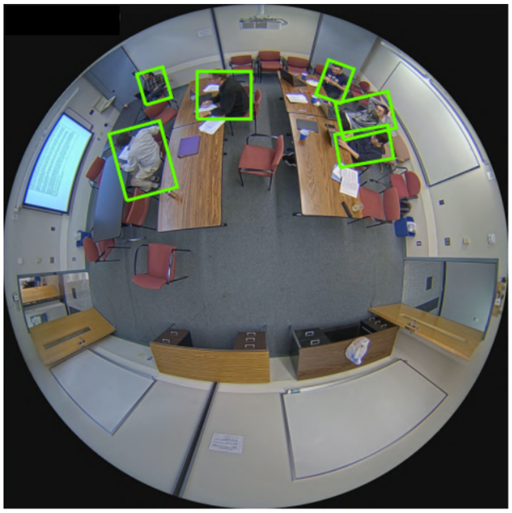


Figure 2. An indoor image captured by a fish-eye camera [21]

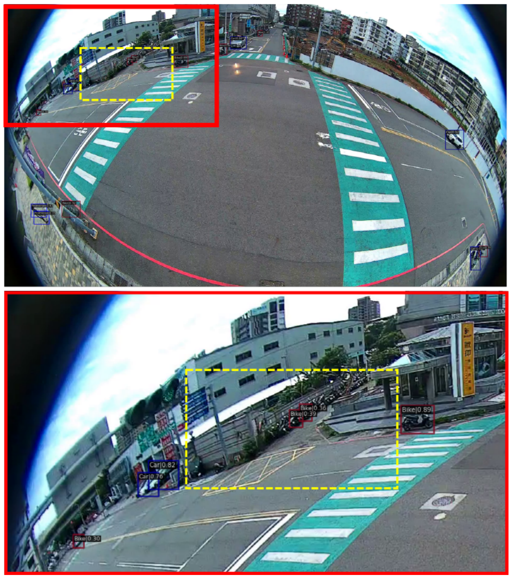


Figure 3. An example of sliced inference. The region corresponding to the red box in the above image is resized to match the model's input size (as shown in the bottom image) before being inputted. As a result, the small objects within the yellow box are greatly enlarged, allowing the model to detect the target objects more accurately.

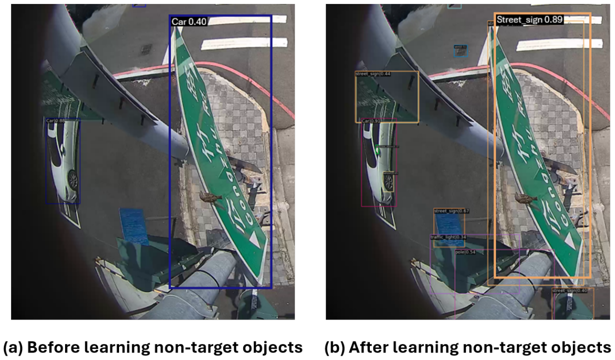
3.1. Task-specific Methods

We investigate the reasons why state-of-the-art object detectors frequently make incorrect predictions in images captured by fish-eye cameras. We observed two key issues in the edge regions of such images. First, we observed that in the edge regions of most road images captured by fish-eye cameras, there are numerous small objects, and detectors often struggle to detect these objects effectively. Due to the characteristics of fish-eye cameras, objects near the edges of images are represented relatively smaller in size compared to those in other areas. As shown in Fig. 2, in indoor settings, the edges of images typically contain walls, reducing the likelihood of objects of interest being present. However, in road environments, where there are no walls and open spaces, objects of interest may be present near the edges of images. Additionally, fish-eye cameras installed on roads cover wider areas compared to indoor settings, leading to a larger number of objects appearing in a single image. Consequently, as the number of objects increases, there is a higher likelihood of multiple objects being located near the edges of the image. Due to these reasons, we observed the presence of numerous small objects in the most of road images.

To effectively detect these small objects, we introduce a sliced inference technique called SAHI (Slicing Aided Hyper Inference) [1]. As shown in Fig. 3, the original image is partitioned into slices of predefined sizes (*i.e.*, the red box in Fig. 3) and each slice is resized to the input size of the detection model for inference. SAHI enables effective object detection for small objects at the inference step for the following reasons: When the original image is smaller than the input size of the model, resizing one slice to the model's input size can make the object size larger than resizing the entire original image to the model's input size. Conversely, when the original image is larger than the input size of the model, shrinking the entire original image to the model's input size is necessary. However, since one slice is usually smaller than the model's input size, resizing it to the model's input size still enlarges the size of each object. Even if the size of one slice is larger than the model's input size, it does not reduce the size of each object compared to resizing the original image to the model's input size. SAHI performs inference by moving slices in a manner similar to convolutional operations, both horizontally and vertically, and aggregates the predicted boxes from each slice. The hyperparameters defined by a user in SAHI are the slice size and the ratio of overlap between each slice (r). In this work, we set the size of the slice to two-thirds of the height and width of the original image and r to 0.25.

Another situation where state-of-the-art detectors frequently fail in object detection is mis-classifying distorted objects. As an example, in plain images, street signs and cars may have little visual similarity, causing most modern detectors to rarely confuse them. However, as shown in Fig. 4 (a), when a street sign is distorted, creating visual similarity with the outline of a car, detectors occasionally mis-predict it as a car.

To address the issue of distorted objects, in this work, we construct a training dataset where pseudo labels are assigned to the most observed objects, excluding the tar-



(a) Before learning non-target objects (b) After learning non-target objects

Figure 4. (a) depicts a mis-prediction of a distorted non-target object. After training on these non-target objects, the issue of incorrect predictions is resolved as shown in (b).

get objects and then train the model using this data (*i.e*., semi-supervised learning). In other words, we enhance the model's object discrimination capability by training it not only on the target detection objects but also on other objects. To assign pseudo labels to as many object categories as possible, we leverage the Co-DETR model that are trained with the large vocabulary instance segmentation (LVIS) dataset [6]. This dataset includes samples of a total of 1,203 types of objects. Since the Co-DETR model currently ranks first for this dataset, we expected to minimize noises in the training data when the model assigns pseudo labels to all objects. By training the model with pseudo-labeled training data, we can prevent issues like street signs being mis-predicted as cars, as shown in Fig. 4 (b).

3.2. General Methods for Object Detection

In this work, we employ various methods known to generally improve the performance of detection models. Data augmentation is known to increase the quantity of data for training, thereby improving detection performance. We utilize basic augmentation methods such as random flip, resize, and random crop, which are commonly used across various detection tasks. Since images captured by fish-eye cameras often exhibit rotation, rotation augmentation is also applied. Data augmentation is applied during the training process.

We also use histogram equalization technique to transform an input image with a narrow range of pixel values, resulting in a high-contrast output image with a wider range of pixel values (see Fig. 5). In other words, it smooths out the pixel distribution of mostly dark or bright images, making them brighter or slightly darker. Histogram equalization is only used during the inference step.

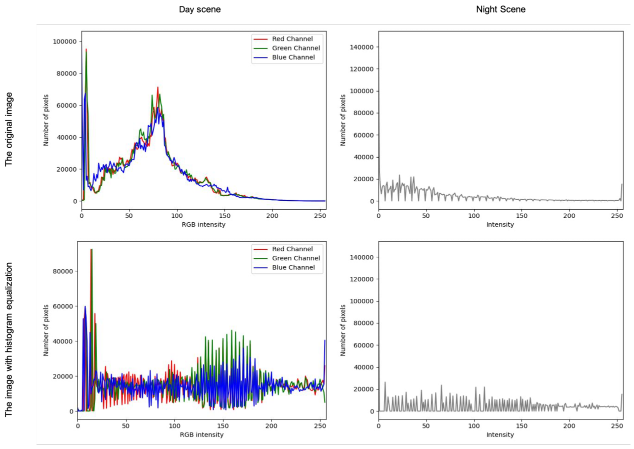


Figure 5. Changes in pixel distribution with histogram equalization

Finally, we utilize super resolution (SR) technique to obtain high-resolution images. In this work, we use pretrained StableSR model [19][1]. The SR is applied during both training and inference steps. In this work, detection models are trained with 1.5 times up-scaled samples, and during the inference step, images are 2 times up-scaled.

3.3. Ensembling Detectors

In this work, various techniques introduced in previous sections are combined in different ways to create various detectors, as shown in Tab. 1. Weighted Boxes Fusion (WBF) [16] is employed to aggregate the predicted bounding boxes from different detectors. As shown in Fig. 6, WBF computes the average coordinates of multiple bounding boxes predicting the same object to generate a single bounding box. The confidence score for the generated bounding box is determined by the average confidence score of the bounding boxes used to create it.

4. Experiments

In this section, we first examine the effectiveness of the task-specific methods that we proposed in this study[2]. We also evaluate the performance of our final detector, which is an ensemble of detectors applied with various methods.

4.1. Datasets and Evaluation Metric

In this work, we utilize the FishEye8k benchmark dataset to train the detectors. The FishEye8k dataset consists of

[1] In this work, we use the checkpoint file "stablesr_768v_000139.ckpt" (https://huggingface.co/Iceclear/StableSR/blob/main/README.md)

[2] Due to the limitation on the number of submissions, in this challenge, we focused solely on evaluating various ensemble combinations, without separately assessing the effectiveness of the general methods used in our work. However, since the task-specific methods proposed in this work had not been validated in previous research, we evaluated their effectiveness separately. The effectiveness of the general methods has already been validated in other studies.

No.	Used Methods
1	Co-DINO (ViT-L) + SAHI
2	Co-DINO (ViT-L) + basic augmentation + SAHI
3	Co-DINO (ViT-L) + image rotation + SAHI
4	Co-DINO (Swin-L) + image rotation + semi-supervision + SAHI
5	Co-DINO (ViT-L) + SAHI + histogram equalization
6	Co-DINO (ViT-L) + basic augmentation + SAHI + histogram equalization
7	Co-DINO (ViT-L) + image rotation + SAHI + histogram equalization
8	Co-DINO (Swin-L) + image rotation + semi-supervision + SAHI + histogram equalization
9	Co-DINO (ViT-L) + SR + SAHI

Table 1. Ensembled detectors in this work. Swin-L [11], ViT-L [4] indicate backbones of DETR models, DINO [23] is an architecture of DETR series. Co-DINO (Swin-L) was pretrained with Objects365 [15] and COCO [10] datasets. Co-DINO (ViT-L) was pretrained with Objects365 [15] and LVIS [6]. All models are fine-tuned with FishEye8K dataset in this work.

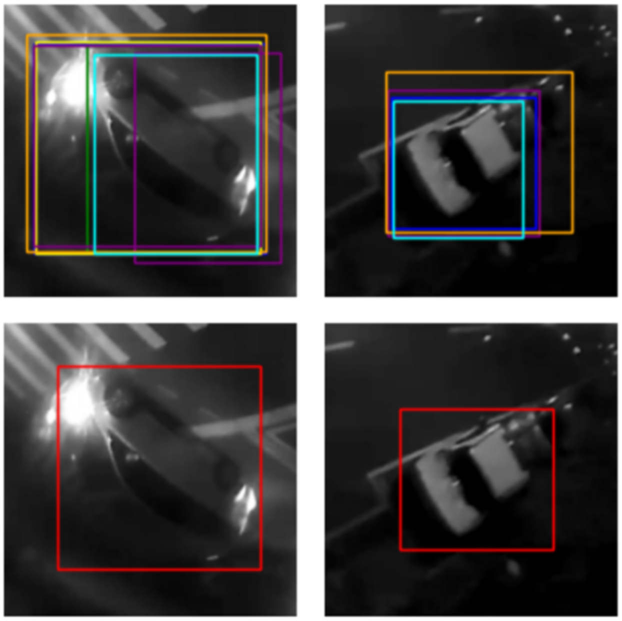


Figure 6. Weighted boxes fusion

Method	F1 Score
Baseline	0.4734
+ Sliced inference	0.5233
+ Semi-supervision	0.5588

Table 2. The results of ablation study on using sliced inference and semi-supervision, respectively

Rank	Team ID	Team Name	Score
1	9	VNPT AI	0.6406
2	**40**	**NetsPresso (ours)**	**0.6196**
3	5	SKKU-AutoLab	0.6194
4	63	UIT_AICLUB	0.6077
5	15	SKKU-NDSU	0.5965
6	33	MCPRL	0.5883
7	156	zzl	0.5828
8	52	DeepDrivePL	0.5825
9	86	NCKU_ACVLAB	0.5637
10	13	FRDC-SH	0.5606

Table 3. Public Top 10 leaderboard for the Challenge Track 4

5,288 training images and 2,712 evaluation images, annotated with a total of 157K bounding boxes, each labeled with one of the five road object classes (Bus, Bike, Car, Pedestrian, Truck). We merged the training and evaluation datasets of FishEye8K to use as training data in this work. For testing, we employ the FishEye1Keval test set, provided as the official evaluation set for Challenge Track 4. This dataset comprises 1,000 images that are not used in creating the FishEye8k dataset. As an evaluation metric, F1 score is used.

4.2. Results

The effectiveness of using task-specific methods As described in Tab. 2, it can be observed that the problem of small objects in the edge regions of images can be effectively addressed through sliced inference techniques. Additionally, by assigning pseudo labels to objects that are not targets for detection and then training the model, it is possible to better distinguish between detection targets and non-detection targets.

Leaderboard As described in Tab. 3, we were able to achieve high performance by combining task-specific methods as well as general methods. We achieved 2nd place in Challenge Track 4.

5. Conclusion

In this paper, we proposed methods to address performance issues caused by distorted or small objects at the edges of images captured by fish-eye cameras. By leveraging various general methods together, we were able to rank highly in Challenge Track 4. However, there are still room for improvement. For example, in some images, objects appeared blurry at the edges, leading to occasional failures in object detection. In future research, we aim to address additional challenges that may arise in the edge regions of images. We will also address the issue of computational complexity in the future work. We utilized a large number of highly complex models to enhance performance, which is not practical. To address this problem, we plan to compress ensembled models into a lightweight single model through techniques such as knowledge distillation and network pruning.

References

[1] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, 2022. 1, 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2020. 2

[3] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6633–6642, 2023. 2

[4] 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 *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*, 2021. 5

[5] Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Erkhembayar Ganbold, Jun-Wei Hsieh, Ming-Ching Chang, Ping-Yang Chen, Byambaa Dorj, Hamad Al Jassmi, Ganzorig Batnasan, Fady Alnajjar, Mohammed Abduljabbar, and Fang-Pang Lin. Fisheye8k: A benchmark and dataset for fisheye camera object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5305–5313, 2023. 2

[6] Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5356–5364, 2019. 4, 5

[7] Malik Haris and Adam Glowacz. Comprehensive review on vehicle detection, classification and counting on highways. *Electronics*, 10, 2023. 2

[8] Yi-Zeng Hsieh, Hau-Ching Chen, and Yi-Hung Yeh. Object detection via fisheye camera. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia (MMAsia)*, 2023. 2

[9] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19702–19712, 2023. 2

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 5

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 5

[12] Prashan Premaratne, Inas Jawad Kadhim, Rhys Blacklidge, and Mark Lee. Comprehensive review on vehicle detection, classification and counting on highways. *Neurocomputing*, 556, 2023. 2

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 2

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[15] 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 (ICCV)*, pages 8430–8439, 2019. 5

[16] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107, 2021. 4

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017. 2

[18] Chien-Yao Wang and Hong-Yuan Mark Liao. YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024. 2

[19] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023. 4

[20] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh

Chakraborty, Sanjita Prajapati, Quan Kong, Norimasa Kobori, Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Ganzorig Batnasan, Fady Alnajjar, Ping-Yang Chen, Jun-Wei Hsieh, Xunlei Wu, Sameer Satish Pusegaonkar, Yizhou Wang, Sujit Biswas, and Rama Chellappa. The 8th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[21] Lu Yang, Liulei Li, Xueshi Xin, Yifan Sun, Qing Song, and Wenguan Wang. Large-scale person detection and localization using overhead fisheye cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19961–19971, 2023. 3

[22] Xipeng Yang, Jin Ye, Jincheng Lu, Chenting Gong, Minyue Jiang, Xiangru Lin, Wei Zhang, Xiao Tan, Yingying Li, Xiaoqing Ye, and Errui Ding. Box-grained reranking matching for multi-camera multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3096–3106, 2022. 2

[23] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*, 2023. 2, 5

[24] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*, 2021. 2

[25] Yu Liu Zhuofan Zong, Guanglu Song. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6748–6758, 2023. 2