Robust Data Augmentation and Ensemble Method for Object Detection in Fisheye Camera Images

Viet Hung Duong[1]
hungdv@vnpt.vn

Duc Quyen Nguyen[1]
quyennd2000@vnpt.vn

Thien Van Luong[2*]
thien.luongvan@phenikaa-uni.edu.vn

Huan Vu[3]
huan.vu@utc.edu.vn

Tien Cuong Nguyen[1]
nguyentiencuong@vnpt.vn

[1] VNPT AI, VNPT Group, Hanoi, Vietnam
[2] Faculty of Computer Science, Phenikaa University, Hanoi, Vietnam
[3] University of Transport and Communications, Hanoi, Vietnam

Abstract

In recent years, traffic surveillance systems have begun leveraging fisheye lenses to minimize the requisite number of cameras for comprehensive coverage of streets and intersections. However, as fisheye images have large radial distortion, they pose new challenges to standard object detection algorithms. In this study, we propose a robust object detection method in traffic scenarios using fisheye cameras. Specifically, we develop a novel data augmentation method, which is applied to VisDrone dataset. Note that we select this dataset for augmentation, since it bears resemblances to the Fisheye8K dataset. Furthermore, we leverage pseudo labels generated by a pre-trained object detection model based on the Fisheye8K and original VisDrone dataset to further enrich the training data. Finally, we utilize various state-of-the-art object detection models trained with different combinations of the proposed augmented data, which are then combined with robust ensemble techniques to further enhance the overall object detection performance. As a result, our proposed method achieves a final F1 score of 64.06% on the 2024 AI City Challenge - Track 4 and ranks first among the competing teams.

1. Introduction

With the escalating demand for intelligent transportation systems and the growing complexity of urban traffic environments, traffic surveillance has become indispensable in modern urban management and safety strategies [46]. The main components of traffic surveillance applications include camera systems and object detection algorithms, which enable automated monitoring, analysis, and management of traffic conditions. While traditional traffic camera systems have predominantly relied on pinhole cameras, which suffer from limited coverage areas, the emergence of fisheye cameras presents a promising alternative. Fisheye cameras offer wide-area coverage with a single camera setup, thereby alleviating the need to install multiple cameras, particularly at road intersections. However, their unique distortion characteristics are really challenging for object detection tasks.

The roots of fisheye cameras trace back to 1908 when [43] first introduced the concept and constructed the first fisheye camera by filling a pinhole camera with water. Subsequently, in 1922, [2] replaced water with a hemispherical lens. Initially employed in automotive surround-view systems, fisheye cameras have gained traction for their wider field of view compared to conventional pinhole cameras, providing additional context and covering blind spots around vehicles. Despite their potential, the absence of publicly available data hindered extensive research on fisheye cameras, particularly in the realm of object detection. Recently, [13] created the Fisheye8K dataset - the first open dataset dedicated to the training and evaluation of road object detection for traffic surveillance, which facilitated work in this branch of research. Due to the optical design of their lenses, images produced by fisheye cameras typically exhibit stronger distortion towards the periphery compared to the center of the image, making objects that are far away from the camera appear shrunk and warped. Furthermore, traditional road object detection challenges, such as class imbalance, occlusion, and viewing perspective, persist in fisheye imagery, exacerbating the complexity of the task.

In this paper, we propose an efficient approach to the vehicle and pedestrian detection problem in the context of

*Corresponding author.

(a) (b)

Figure 1. Predictions made by our proposed framework on the 2024 AI City Challenge - Track 4 test dataset, where red, blue, pink, yellow, and green bounding boxes represent predictions for Cars, Bikes, Pedestrians, Trucks, and Buses, respectively. Figure 1a and Figure 1b demonstrate bounding boxes detected on images captured during daytime and nighttime, respectively.

fisheye camera in the 2024 AI City Challenge - Track 4 [40]. Our main contributions are summarized as follows:

- We propose a novel data augmentation method tailored to the VisDrone dataset [8], which aims to generate synthetic data having similar characteristics to the Fisheye8K dataset. Additionally, we leverage pseudo labels generated by our pre-trained CO-DETR [48] model based on Fisheye8K and the original VisDrone dataset. Subsequently, we combine the synthetic VisDrone data and pseudo data with Fisheye8K to train an ensemble-based object detection model detailed in the following.
- We introduce a novel ensemble model that combines 4 state-of-the-art object detection models, namely, YOLOv9-e [39], YOLOR-W6 [38], InternImage [41], and CO-DETR [48] using the Weighted Boxes Fusion (WBF) [34] method. These selected models are trained with different combinations of VisDrone, synthetic VisDrone, FishEye8k, and pseudo data (see Section 3).
- Finally, we conduct extensive experiments to demonstrate the superior performance of our proposed data augmentation and ensemble method over the state-of-the-art baselines, securing the 1st position in Track 4 of the challenge. Specifically, Figure 1 illustrates the objects detected by our proposed method on samples captured during daytime and nighttime in the 2024 AI City Challenge - Track 4 test dataset. It is shown in this figure that our method accurately detects and classifies objects in the presence of diverse environmental contexts, including diminutive objects situated at the image periphery.

The remainder of the paper is organized as follows. Section 2 provides a review of pioneering works on object detection in fisheye images. Section 3 elaborates on our approach and system architecture. We summarize experiment results and implementation details of the proposed method in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Object Detection

Object Detection is a fundamental task in computer vision that involves identifying and localizing objects within images or videos. In traffic surveillance applications, utilizing object localization algorithms facilitates automatic monitoring and analyzing traffic flow, leading to effective traffic management. Regarding technical design, object detection algorithms can be categorized into two main groups: two-stage and one-stage detectors. In two-stage object detection methods, the process involves two main stages: region proposal and classification. The algorithm first generates a set of candidate object bounding boxes using techniques such as selective search, edge boxes, or region proposal networks. Afterward, the proposed regions of interest are fed into a classifier to predict the presence of objects and refine the bounding boxes' coordinates. Popular two-stage object detection architectures include R-CNN [12], Fast R-CNN [11], Faster R-CNN [30], and Mask R-CNN [14]. In contrast, one-stage detectors are designed to directly estimate the bounding boxes' coordinates and class probabilities in a single pass through the network. Examples of one-stage object detection methods include YOLO [1, 9, 17, 27–29, 37–39], SSD [20], RetinaNet [19], EfficientDet [36], and InternImage [41]. Recently, there has been a paradigm shift in object detection that leverages attention mechanisms and transformer-based architecture initially designed for natural language processing tasks. Several detection architectures based on transformers have gained popularity due to their effectiveness, including DETR [4], DEformable-DETR [47], Swin Transformer [21], and CO-DETR [48].

2.2. Road Detection Datasets

Dedicated datasets play a pivotal role in training and evaluating object detection algorithms, especially in traffic settings. Hence, multiple datasets have been created explicitly for various tasks, such as road detection and autonomous driving. Road detection datasets typically consist of images captured from an overhead view, often extracted from traffic surveillance cameras or drones. Well-known datasets used for road detection tasks include the UA-DETRAC [42], the MIO-TCD [23], the UAV [7], and the VisDrone dataset [8]. In contrast, datasets designed for self-driving scenarios are often created using cameras mounted on vehicles. Examples of object detection datasets created for this task include the KITTI [10], the Eurocity Persons [3], and the Cityscapes dataset [6].

2.3. Object Detection in Fisheye Images

Despite the growing popularity and long development history of fisheye cameras, publicly available datasets for fisheye images remain limited. In 2019, [44] created the WoodScape dataset, the first comprehensive dataset for road detection. However, the dataset was explicitly designed for autonomous driving. The Fisheye8K [13], published in 2023, was the first fisheye image dataset dedicated to traffic surveillance, and it is the foundation of the 2024 AI City Challenge - Track 4 [40]. Fisheye images exhibit strong radial distortion, which can affect the appearance of objects, making their shapes and sizes different from those in perspective images. Thus, it is challenging for traditional object detection algorithms to accurately detect objects in fisheye images due to the distorted representations. Addressing this challenge often requires developing specialized techniques and algorithms tailored to fisheye imagery. In 2018, [5] introduced the spherical CNNs (SCNNs) that were specifically constructed for analyzing spherical images. Afterward, in 2019, [45] created a neural network based on SCNNs that specialized in object detection for panoramic images. In [26], the authors surveyed different object representations and proposed a curved bounding box model that possesses the optimal properties for fisheye images. Despite the development of multiple techniques, only a few were explicitly targeted at road object detection tasks.

3. The Proposed Method

3.1. Data Selection

We approach the competition with a data-centric strategy rather than focusing solely on the model. There are two main problems we needed to address: finding public datasets that are similar to the Fisheye8K [13] dataset to augment our data and finding data augmentation methods to handle the unique characteristics of fisheye camera data. We thoroughly survey several public datasets containing classes relevant to traffic surveillance, such as UA-DETRAC [42], Eurocity Persons [3], Cityscapes [6], MIO-TCD [23], UAV [7], and VisDrone [8]. Note that the classes of interest include truck, pedestrian, motorbike, bus, and car.

Observing datasets like Eurocity Persons, Cityscapes, and UA-DETRAC, we note that the images were predominantly captured from front-facing cameras with low viewing angles and large object sizes, limiting their resemblance to fisheye camera data. On the other hand, datasets like MIO-TCD, UAV, and VisDrone, containing images captured from drones with high viewing angles and numerous small objects, show close similarities with the Fisheye8K dataset. Therefore, we individually combine these three datasets with the Fisheye8K dataset for further experiments.

Through experimental evaluations presented in Section 4.2, we determine that combining the VisDrone dataset with the Fisheye8K dataset yields the best performance (see Table 1 in Section 4). Consequently, we select the VisDrone + Fisheye8K dataset for further experiments.

Additionally, we explore various data augmentation techniques tailored to fisheye camera data, including techniques that transform regular images into fisheye images. One such technique involves transforming regular images into fisheye-like images to mimic the unique characteristics of the Fisheye8K dataset. This approach aims to bridge the gap between datasets captured from different perspectives, enhancing the model's ability to generalize across diverse environments. We refer to this augmented dataset as Synthetic VisDrone. The experiment results in Section 4.3 indicate that while there is not a significant improvement in the overall mAP, the model trained on the Synthetic VisDrone dataset performs exceptionally well in predicting small objects at the edges of the frame. Furthermore, by leveraging ensemble modeling techniques, we further enhance the accuracy of the main model by integrating predictions from the model trained on the Synthetic VisDrone dataset, as will be detailed in Section 4.4.

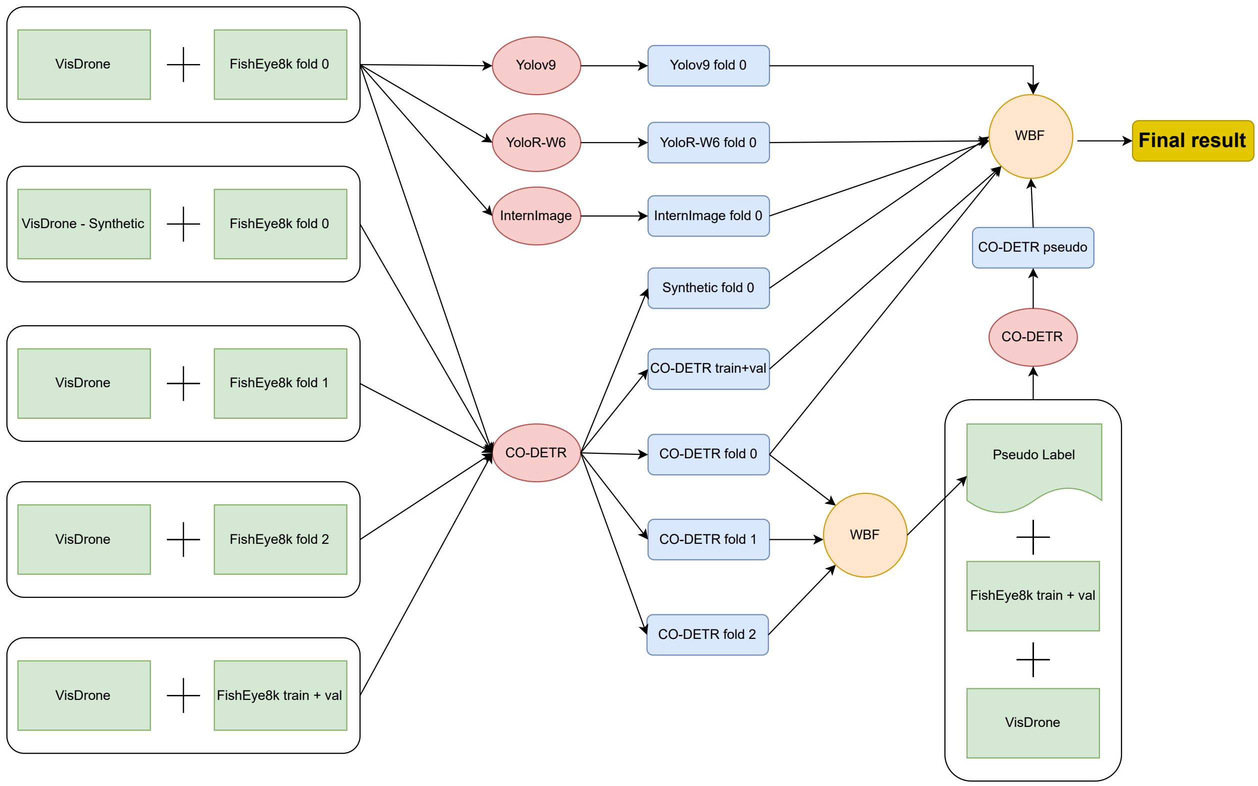


Figure 2. The proposed framework for object detection using fisheye camera images, where green blocks represent datasets, red blocks represent models, and blue blocks represent the model's output. The final prediction is obtained by employing the Weighted Boxes Fusion method to ensemble 7 models, consisting of 3 models (YOLOv9-e, YOLOR-W6, InternImage) trained with the VisDrone + Fisheye8K dataset, and 4 CO-DETR models trained with 4 different datasets: train data, train + val data, train + val + pseudo data, and synthetic data.

3.2. Model Selection

In terms of model selection, we choose the current top-ranked model in the object detection category of the COCO dataset, namely, CO-DETR [48]. Additionally, we also utilize a combination of other models such as YOLOR-W6 [38], which achieves the best performance as reported by the authors of the Fisheye8K dataset in their paper. YOLOv9-e [39], the latest object detection model in the YOLO family, is also included for experimentation to evaluate its performance on the Fisheye8K dataset. InternImage [41], a well-known model that achieves high ranks in object detection leaderboards, is also employed.

After training these models on the VisDrone + Fisheye8K dataset, we employ the WBF method [34] for ensemble modeling. Note that WBF is currently the most effective bounding box ensemble method for object detection tasks, as evidenced by its performance in various object detection competitions. In addition, we employ the pseudo-labeling method to further improve the accuracy on the test dataset. The proposed framework are illustrated in Figure 2.

4. Experiment Results and Discussion

4.1. Evaluation Metrics

Mean Average Precision Initially, the evaluation metric used for the 2024 AI City Challenge - Track 4 was the mAP, which is the mean of average precision over all classes. Because the mAP inadvertently favors strategies that lead to many false positives in detection, the evaluation metric was modified later in the competition. Consequently, most of our experiments are evaluated based on mAP formulated as

$$\text{mAP} = \frac{1}{n} \cdot \sum_{k=1}^{n} \text{AP}_k. \quad (1)$$

F1-score The primary ranking criterion was later changed to the harmonic mean of total Precision and Recall, which is the F1-score or F-measure [31]. The F1 metric serves as a balanced measure that combines Precision and Recall into a single score, offering insights into a model's effectiveness in correctly identifying instances of positive classes while minimizing false positives and false negatives.

The F1-score is calculated as follows:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2}$$

4.2. Dataset

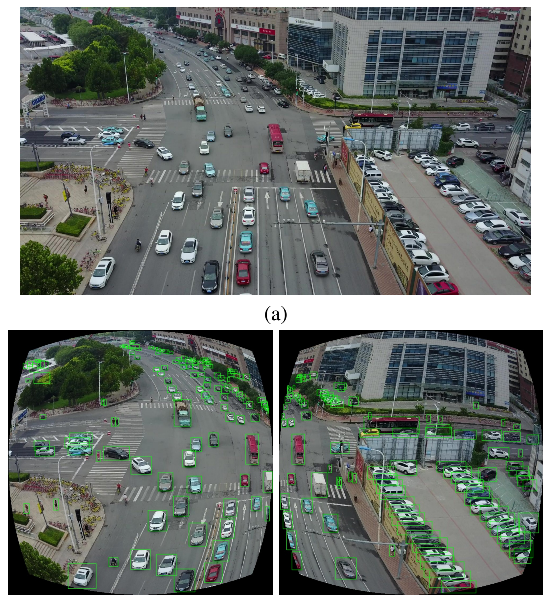
Original Dataset Track 4 of the AI City Challenge 2024 [40] is based on the Fisheye8K dataset [13]. The original dataset contains 8000 images extracted from 35 fisheye cameras in various locations across Hsinchu City, Taiwan. Images are captured at 1080x1080 and 1280x1280 resolutions. Annotations are provided for objects belonging to five categories, including Bus, Bike, Car, Pedestrian, and Truck, with a total of 157,000 bounding boxes. The dataset is divided into a training set, which consists of 5288 images, and a validation set, which has 2712 images.

Additional Dataset To increase diversity and improve the detection performance, we utilize additional datasets. According to the rules of the 2024 AI City Challenge - Track 4, using any non-public dataset for training, validation, or testing is invalid and will not be qualified for the challenge awards. Thus, we experiment with three public datasets, including the UAV dataset [7], the MIO-TCD dataset [23], and the VisDrone dataset [8]. We chose these three datasets because they bear several resemblances to the original Fisheye8K dataset, such as containing small objects, having images captured from an overhead view, and annotating similar object categories. We modify the annotations' categories and combine the datasets with the original Fisheye8K training set individually. Afterward, we fine-tune the CO-DETR [48] model for 16 epochs on each combined dataset and evaluated the results with the $mAP_{0.5\text{-}0.95}$ metric on the Fisheye8K validation set. As shown in Table 1, enhanced performance was only witnessed when training on the combined dataset composed from the Fisheye8K train set and the VisDrone train set. Hence, we select the VisDrone dataset as the additional dataset for our further experiments.

Training data	$mAP_{0.5\text{-}0.95}$
Fisheye8K only	47.00
MIO-TCD + Fisheye8K	44.50
UAV + Fisheye8K	43.50
VisDrone + Fisheye8K	**49.05**

Table 1. Performance comparison of CO-DETR model trained with different datasets on validation set.

Data Augmentation To effectively improve the performance of our model on the Fisheye8K dataset, it is necessary to generate fisheye images from the pre-selected VisDrone dataset. We expect that training on additional synthetic fisheye images makes the models more robust to radial distortion, thus enhancing their generalization ability on fisheye images. There are multiple methods for applying the fisheye effect on ordinary images [32, 35]. In our work, we utilize the formula implemented by the iFish tool [25] due to its efficiency and simplicity. When generating synthetic data, the original image is split into two square images to minimize the dark area around fisheye images. Afterward, the images are transformed and cropped to remove the surrounding dark area. To transform the bounding box's coordinates, we convert the coordinates of each vertex independently, and then we calculate the coordinates of the new top-left and bottom-right corners by taking the minimum and maximum of the newly calculated x-coordinates and y-coordinates, respectively. The process of converting an ordinary image to two fisheye images is demonstrated in Figure 3. For convenience, we term the new dataset as Synthetic VisDrone.



(a) (b) (c)

Figure 3. Example of the synthetic data generation process. Figure 3a depicts the original image. Figure 3b and Figure 3c illustrate the fisheye images generated from the left and right halves.

4.3. Implementation Details

CO-DETR In our study, we employ the CO-DETR model with the Swin-L backbone architecture, which was pre-trained with the COCO [18] and the Objects365 [33] datasets. The checkpoints of this model are publicly available on the mmdetection GitHub repository [24]. The training is conducted over 16 epochs with a learning rate of 1e-5. Throughout the training process, the image size is randomly resized from 480 to 2048. During inference, the image size is resized using a scale of (1920, 2048). We utilize the CO-DETR model trained on two datasets: VisDrone + Fish-

Model	Pretraining data	Data used for finetuning	$mAP_{0.5\text{-}0.95}$
CO-DETR	**COCO + Objects365**	**VisDrone + Fisheye8K fold 0**	**49.05**
CO-DETR	COCO + Objects365	Synthetic VisDrone + Fisheye8K fold 0	45.78
InternImage	ImageNet22k	VisDrone + Fisheye8K fold 0	41.11
YOLOv9-e	None	VisDrone + Fisheye8K fold 0	43.89
YOLOR-W6	COCO	VisDrone + Fisheye8K fold 0	43.47

Table 2. Performance comparison of different object detection models on the Fisheye8K validation set.

eye8K and VisDrone Synthetic + Fisheye8K. The model's performance is summarized in Table 2.

InternImage We use a COCO-pretrained InternImage-L model [41] and fine-tune it on the dataset composed of the Fisheye8K training set and the VisDrone training set for 50 epochs using the AdamW optimizer [22]. The learning rate is set to 1e-4.

YOLOv9 We train the YOLOv9-e [39] on the dataset composed of the Fisheye8K training set and the VisDrone training set. Since the COCO-pretrained model provided by [39] is specifically tailored for images of size 640x640, we train the YOLOv9-e on input size 1280x1280 from scratch for 250 epochs using the stochastic gradient descent (SGD) optimizer [15] with the learning rate of 0.01.

YOLOR We fine-tune the COCO-pretrained YOLOR-W6 model [38] on the dataset composed of the Fisheye8K training set and the VisDrone training set for 250 epochs using the Adam optimizer [16] with the learning rate of 0.01 and the input size of 1280x1280.

We run all experiments on one DGX node with 8 NVIDIA A100-80GB GPU. The result of each model on the Fisheye8K validation set is shown in Table 2. Evidently, when training with the dataset composed of the Fisheye8K training set and the VisDrone training set, the CO-DETR achieves superior performance compared to the remaining models. Hence, we select it as the main model for our further experiments. Regarding the YOLO models, it is shown via Table 2 that the YOLOv9-e demonstrates great potential, achieving better mAP compared to YOLOR-W6 and InternImage without using pretrained checkpoints. Notably, when utilizing the Synthetic VisDrone dataset instead of the original VisDrone dataset in the training process, the mAP of the CO-DETR model decreases. However, as illustrated in Figure 4, using the Synthetic VisDrone dataset makes the CO-DETR model more robust to radial distortion, thus enhancing its accuracy when objects are located towards the periphery of the image. As a result, ensembling the predictions of the two models will increase the overall performance, as will be demonstrated in Table 9 in Section 4.4.

(a) Original VisDrone

(b) Synthetic VisDrone

Figure 4. Predictions made by the CO-DETR model. Figure 4a and Figure 4b illustrate the bounding boxes detected for the Bike objects by the CO-DETR model trained on the original VisDrone dataset and the synthetic VisDrone dataset, respectively.

4.4. Training Strategy and Performance Analysis

Fine-tuning Strategy We conduct several experiments to evaluate the effectiveness of different fine-tuning strategies. The backbone architecture used for these experiments is CO-DETR. The baseline model, which is pretrained on the COCO dataset and the Objects365 dataset, then fine-tuned on the Fisheye8K dataset, achieves a mAP of 47% on the validation set. In a second experiment, the same pretrained model is fine-tuned on the VisDrone dataset, resulting in an mAP of 30.8%. Subsequently, we pretrain a model on the VisDrone dataset, then fine-tune it on the Fisheye8K dataset, yielding a mAP of 47.8%. Utilizing VisDrone pretrained model instead of COCO + Objects365 results in a 0.8% mAP increase. Finally, we finetune the same model previously pretrained on the COCO + Objects365 dataset, using a combination of the VisDrone and Fisheye8K datasets, resulting in a noteworthy enhancement of 2.05% mAP, bringing it to 49.05%. The results of these experiments are shown in Table 3. In subsequent steps, we adopt the best approach of combining the VisDrone and the Fisheye8K datasets together to fine-tune the CO-DETR model pretrained on the COCO and Objects365 datasets.

Pretraining data	**Finetuning data**	mAP
COCO-Objects365	Fisheye8K	47.00
COCO-Objects365	VisDrone	30.80
VisDrone	Fisheye8K	47.80
COCO-Objects365	**VisDrone + Fisheye8K**	**49.05**

Table 3. Performance comparison of CO-DETR model trained with different training strategies on the validation set.

K-fold Split We partition the Fisheye8K dataset into 3 folds, with fold 0 distributed according to the organizers' default distribution. The folds are divided by camera IDs, ensuring a 70-30 ratio of object quantities between the training and validation sets. Videos selected for the validation set in one fold are excluded from the validation sets of other folds. Based on these criteria, for fold 1, we select videos 1, 2, 4, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, and 18 for the training set, and videos 3, 5, and 11 for the validation set. For fold 2, we choose videos 1, 2, 3, 4, 5, 7, 8, 10, 11, 15, and 17 for the training set, and videos 6, 9, 12, 13, 14, 16, and 18 for the validation set. The number of objects per class and their specific ratios are described in Table 4, 5, 6. Each fold is used to train a CO-DETR model with configuration settings outlined in Section 4.3. The mAP results of the models are presented in Table 7.

Class	**Train set**		**Val set**	
Bus	2052	68.8%	930	31.2%
Bike	62068	70.2%	26305	29.8%
Car	36473	72.1%	14124	27.9%
Pedes	9111	77.6%	2632	22.4%
Truck	2115	63.8%	1202	36.2%

Table 4. Fold 0 data split from the Fisheye8K dataset.

Pseudo Labeling Our proposed pseudo-label process involves three steps as illustrated in Figure 5. Particularly, in step 1, we generate pseudo-labels by ensembling the results of the 3-fold models presented above. Then, in step 2, we combine the training and validation data from fold 0 to form new training data and validate it on the pseudo-labels we created. Finally, we combine pseudo-labels with the training and validation data to form new training data. In this step, there is no validation data, and we select the model

Class	**Train set**		**Val set**	
Bus	2193	73.5%	789	26.5%
Bike	59181	67.0%	29192	33.0%
Car	33912	67.0%	16685	33.0%
Pedes	9379	79.9%	2364	20.1%
Truck	2942	88.7%	375	11.3%

Table 5. Fold 1 data split from the Fisheye8K dataset.

Class	**Train set**		**Val set**	
Bus	2329	78.1%	653	21.9%
Bike	66235	74.9%	22138	25.1%
Car	37972	75.0%	12625	25.0%
Pedes	8111	69.1%	3632	30.9%
Truck	2284	68.9%	1033	31.1%

Table 6. Fold 2 data split from the Fisheye8K dataset.

Data	$mAP_{0.5\text{-}0.95}$
VisDrone + Fisheye8K fold 0	56.23
VisDrone + Fisheye8K fold 1	55.84
VisDrone + Fisheye8K fold 2	54.51

Table 7. K-fold mAP performance of CO-DETR on the test set.

obtained from the last epoch. Table 8 demonstrates a significant performance improvement achieved by CO-DETR trained with the proposed pseudo-data compared with those without using the pseudo-data. In other words, this table indicates that the performance is constantly improved across three steps (see Figure 5). This is due to the fact that the proposed pseudo-data helps the model better familiarize and recognize patterns in the test data.

Training data	$mAP_{0.5\text{-}0.95}$
Visdrone + Fisheye8K fold 0	56.23
Visdrone + Fisheye8K train + val	58.40
Visdrone + Fisheye8K train+val+pseudo	**61.02**

Table 8. Performance of CO-DETR model trained with our proposed pseudo-label method on the test set.

Model Ensembling We employ the WBF method [34] to ensemble multiple models, using an IOU threshold of 0.75 and a skip bounding box threshold of 0.15. As illustrated in Figure 2 of Section 3, seven models are chosen for the ensemble: CO-DETR trained on train+val+pseudo data and CO-DETR trained on train+val data from Section 4.4, as well as YOLOv9, YOLOvR-w6, InternImage, CO-DETR Synthetic, and CO-DETR fold 0 from Table 2. When ensembling the models, CO-DETR trained on train+val+pseudo data is assigned with the highest weight due to having the highest mAP, followed by the remaining

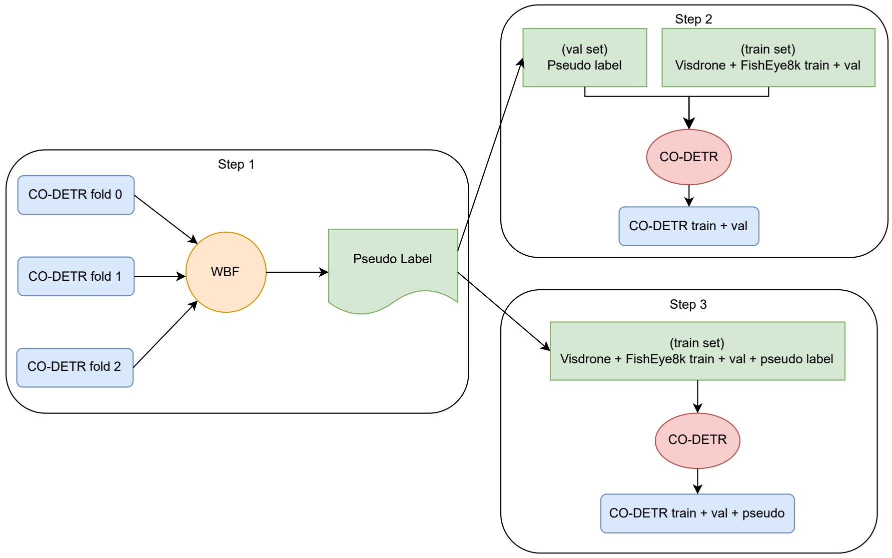


Figure 5. The proposed pseudo-labeling process.

models.

When testing on the validation set, we achieve the best performance using a confidence score threshold ranging from 0.3 to 0.4 for each class. For night-time camera footage, this threshold ranges from 0.2 to 0.3. We apply these thresholds to the final ensemble results. The best F1-score achieved by our proposed approach is 64.06% on the public leaderboard. As such, ensembling the models helps increase the accuracy by approximately 1.5% over the baseline + pseudo scheme, as shown in Table 9.

Model	F1 score
Baseline (CO-DETR only)	57.02
Baseline + pseudo (CO-DETR only)	62.46
Synthetic + pseudo + ensemble (Ours)	**64.06**

Table 9. Final F1 score of our proposed method on the test set in comparison with the baselines, where the baseline refers to the CO-DETR model trained on the VisDrone + Fisheye8K fold 0 data, baseline + pseudo stands for the CO-DETR model trained on the VisDrone + Fisheye8K data along with pseudo labels, and synthetic + pseudo + ensemble represents our final solution by ensembling 7 models as well as exploiting the proposed synthetic data and pseudo-data for training, as shown in Figure 2 of Sec. 3.

5. Conclusions

In this paper, we proposed a robust object detection method for fisheye camera images, which wisely combines the advantages of advanced techniques, such as, data augmentation, pseudo-labeling and model ensembling. Particularly, for data augmentation, we focused on finding datasets most similar to the Fisheye8K dataset. The VisDrone dataset has been chosen, as it is empirically proven to significantly improve the performance compared to others. We then developed an efficient data augmentation applied to VisDrone for generating synthetic data supplemented the model in detecting objects at the far distance and at the edges of the frame. We further enriched training data by proposing the pseudo-labeling process. Furthermore, we utilized the state-of-the-art CO-DETR object detection models to notably enhance the detection accuracy. Finally, ensembling it with other models such as YOLOv9, InternImage, and YOLOvR-w6 further improved the performance. As a result, we achieved the 1st rank in the competition with a F1-score of 64.06% on the leaderboard, as seen via Table 10.

Rank	Team Name	F1 score
1	**VNPT AI**	**64.06**
2	NetsPresso	61.96
3	SKKU-AutoLab	61.94
4	UIT-AICLUB	60.77
5	SKKU-NDSU	59.65

Table 10. Final leaderboard of Track 4.

6. Acknowledgment

The work was sponsored by Vietnam Posts and Telecommunications Group (VNPT). We would like to thank Mr. Dien Hy Ngo, Deputy General Director of the Group, for his constant encouragement and support to the research team. We also express our gratitude to the AI Lab department of VNPT AI for providing the DGX A100 infrastructure for model training. Additionally, we extend our thanks to Dr. Hung T. Le of VNPT AI for his assistance in reviewing this paper.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[2] WN Bond. Lxxxix. a wide angle lens for cloud recording. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 44(263):999–1001, 1922. 1

[3] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu M Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1844–1861, 2019. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[5] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018. 3

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 3

[7] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018. 3, 5

[8] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Lin, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 2, 3, 5

[9] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 3

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra 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. 2

[13] 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, et al. Fisheye8k: a benchmark and dataset for fisheye camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5304–5312, 2023. 1, 3, 5

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[15] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952. 6

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 6

[17] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022. 3

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander 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. 3

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6

[23] Zhiming Luo, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018. 3, 5

[24] MMDetection Contributors. OpenMMLab Detection Toolbox and Benchmark, 2018. 5

[25] Gil Mor. ifish tool, 2021. 5

[26] Hazem Rashed, Eslam Mohamed, Ganesh Sistu, Varun Ravi Kumar, Ciaran Eising, Ahmad El-Sallab, and Senthil Yogamani. Fisheyeyolo: Object detection on fisheye cameras for autonomous driving. In *Proceedings of the Machine Learning for Autonomous Driving NeurIPS 2020 Virtual Workshop, Virtual*, 2020. 3

[27] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[29] 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*, pages 779–788, 2016. 3

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[31] Cornelius J. Van Rijsbergen. *Information Retrieval*. 1979. 4

[32] Kaustubh Sadekar. Omnicv tool, 2020. 5

[33] 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. 5

[34] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107: 104117, 2021. 2, 4, 7

[35] Synthesis-AI-Dev. Fisheye-distortion tool, 2020. 5

[36] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 3

[37] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 3

[38] CHIEN-YAO WANG, I-HAU YEH, and HONG-YUAN MARK LIAO. You only learn one representation: Unified network for multiple tasks. *Journal of Information Science & Engineering*, 39(3), 2023. 2, 4, 6

[39] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024. 2, 3, 4, 6

[40] 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, 3, 5

[41] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023. 2, 3, 4, 6

[42] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193:102907, 2020. 3

[43] Robert W Wood. Xxiii. fish-eye views, and vision under water. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 12(68):159–162, 1906. 1

[44] Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uricár, Stefan Milz, Martin Simon, Karl Amende, et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9308–9318, 2019. 3

[45] Dawen Yu and Shunping Ji. Grid based spherical cnn for object detection from panoramic images. *Sensors*, 19(11): 2622, 2019. 3

[46] Xingchen Zhang, Yuxiang Feng, Panagiotis Angeloudis, and Yiannis Demiris. Monocular visual traffic surveillance: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14148–14165, 2022. 1

[47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 3

[48] 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. 2, 3, 4, 5