FE-Det: An Effective Traffic Object Detection Framework for Fish-Eye Cameras

Xingshuang Luo[1], Zhe Cui[1,2]*, Fei Su[1,2]
[1]Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, China
{luoxingshuang,cuizhe,sufei}@bupt.edu.cn

Abstract

In the realm of intelligent traffic systems, fisheye cameras have emerged as a pivotal tool, distinguished by their expansive field of view which significantly enhances the surveillance of complex street networks and intersections. However, the inherent distortion characteristics of fisheye lenses, various illumination, tiny objects and confusion of vehicle classes pose significant challenges to conventional image processing and object detection techniques. To address these challenges, we propose an advanced object detection framework named ***FE-Det*** *specifically designed for fisheye cameras in traffic monitoring systems. This framework integrates detection models optimized for day and night scene variability. Additionally, it incorporates innovative post-processing operations which brings detection enhancement, including a Vehicles Classifier Module for precise vehicle identification, a Static Objects Processing Module for more accurate detection of stationary objects and a Confidence Score Refinement Module to adjust confidence scores for improving the detection of peripheral objects. Experimental evidence substantiates that our framework exhibits a 1.4% improvement in distinguishing between day and night scenes compared to traditional models. Moreover, the application of the proposed post-processing method results in an additional enhancement of 4.1%.*

1. Introduction

Fisheye cameras have recently attracted widespread attention across industries. In traditional methodologies for the acquisition of environmental information, reliance has predominantly been placed upon narrow-angle pinhole cameras. However, comprehensive environmental perception is a necessity for autonomous vehicular technologies. In contrast, fisheye camera, characterized by its expansive field of view and extensive perspective, has emerged as a pivotal tool within the domain of autonomous driving [18]. Moreover, within the sphere of intelligent traffic systems, the application of fisheye cameras provides an unrivaled panoramic overview, surpassing the capabilities of conventional narrow field of view cameras in capturing broad environmental expanses. This attribute facilitates a significant diminution in the requisite number of cameras for the surveillance of extensive street networks and complex intersections, thereby augmenting the efficiency and comprehensiveness of traffic surveillance endeavors [8].

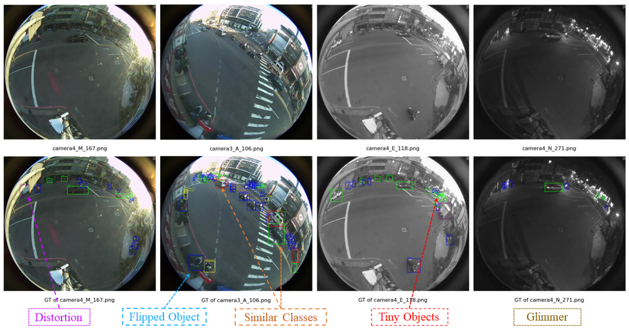


Figure 1. Hard samples in FishEye8K dataset. Red, blue, green, yellow, and purple rectangular boxes represent Bus, Bike, Car, Pedestrian, and Truck, respectively.

Nonetheless, the intrinsic properties of fisheye optics introduce geometric distortion to the captured images, posing substantive challenges to the established frameworks of image processing and object detection. Current object detection methods in fisheye images are grouped into two categories: distortion correction-based and original fisheye image-based [14]. In the distortion correction-based methods, a common practice is to use a fourth-order polynomial model or a unified camera model to correct distortions in images [3]. However, camera parameters are often unknown, and undistorted images come with artifacts from resampling distortion, especially at the periphery, reduced field of view, and non-rectangular images caused by invalid pixels [17].

However, conducting detection directly on the original

[*]Corresponding author

fisheye images of street scenes presents numerous challenges. For instance, fisheye cameras introduce target rotation that is dependent on the scene location, complicating the differentiation of vehicle categories (Car, Truck, and Bus) due to shape distortions caused by the lens. Moreover, there is a significant disparity in target features between daytime and nighttime scenes, complicating detection. Small objects at the image periphery are also difficult to detect, and street scenes frequently contain numerous densely packed stationary vehicles. Fig. 1 shows some hard examples in FishEye8K dataset.

Thus, to enhance the performance of current general-purpose object detection models on original fisheye images, we have developed An Effective Traffic Object Detection Framework for Fisheye Cameras, called FE-Det. This framework, by integrating advanced detection models and separately addressing day and night scenes, coupled with meticulous and innovative post-processing operations, achieves precise object detection on original fisheye images. Specifically, we experiment on several excellent object detection models, implement a comprehensive suite of data preprocessing operations to address issues of rotational distortion, and develop a specialized classification branch for vehicular identification. Additionally, we adjust confidence scores of small objects located at the periphery of images to enhance detection accuracy. A static objects processing module is also introduced, utilizing the Structural Similarity Index (SSIM) to assess the similarity of image regions, thereby facilitating refined handling of static objects [25].

In summary, the main contributions of this paper are as follows:

1) We propose a novel detection framework tailored for fisheye images in complex street scenes. It conducts segregated processing of day and night images, integrates Co-DETR [27], YOLOv8 [11] and InternImage [24], and enhances the detection accuracy through sophisticated post-processing techniques.

2) We have designed a creative and plug-and-play post-processing pipeline, comprising the Vehicles Classifier Module, Static Objects Processing Module, and Confidence Score Refinement Module.

3) The comprehensive experiments show the efficiency of the framework. Finally, our model is ranked 6th in 2024 AI CITY CHALLENGE Track 4 [23].

2. Related Work

2.1. General Object Detection

Object detection, as one of the foundational challenges in the field of computer vision, aims to classify instances of objects and locate their positions with bounding boxes within images.

YOLOv8. In the latest evolution of object detection models, YOLOv8 emerges as a preeminent solution, showcasing unparalleled capabilities for simultaneous detection. This iteration inherits the architectural paradigm of its predecessor, YOLOv5 [12], encompassing a backbone, head, and neck, while introducing novel architectural advancements, upgraded convolutional layers within the backbone, and a more sophisticated detection head. These enhancements position YOLOv8 as a prime candidate for real-time object detection tasks. The model utilizes the Darknet-53 backbone network, which surpasses YOLOv7 [22] in both speed and accuracy. YOLOv8 employs an anchor-free detection head to predict bounding boxes, leveraging an expanded feature map and an optimized convolutional network to surpass previous versions in efficiency and accuracy. Furthermore, YOLOv8 integrates feature pyramid networks to adeptly recognize objects of various sizes [1]. Moreover, YOLOv8 incorporates support for cutting-edge computer vision algorithms, including image classification and instance segmentation with high precision.

Co-DETR. Co-DETR enhances the proficiency of detection transformers through the utilization of several parallel auxiliary heads. This innovative training strategy simplifies the enhancement of the encoder's capability for end-to-end detection learning. It achieves this by directing some auxiliary heads to perform simultaneous one-to-many label assignments, such as ATSS, FCOS and Faster RCNN [10]. Furthermore, Co-DETR introduces an optimization by executing specialized positive queries. This is accomplished by extracting positive coordinates from the auxiliary heads, thereby improving the decoder's training efficiency and its ability to accurately identify positive samples. Upon the completion of training, these auxiliary heads are discarded. This design choice ensures that the Co-DETR method does not incur additional variables or computational overhead compared to its predecessors, while also eliminating the necessity for manual intervention in non-maximum suppression (NMS) [13].

InternImage. InternImage emerges as a foundational model predicated on the large-scale CNN framework, echoing the parameter and data scaling strategies reminiscent of Vision Transformers. At the core of InternImage design lies the implementation of deformable convolutions, a pivotal feature enabling the model to encapsulate more nuanced contextual details in object representations. Distinguishing itself from conventional CNN architectures, InternImage integrates adaptive spatial aggregation mechanisms. These mechanisms are fine-tuned based on the specifics of the input and the task at hand, thereby mitigating the rigid inductive biases typically associated with traditional CNN models [4]. Cascade Mask R-CNN [5] based on InternImage has excellent detection performance.

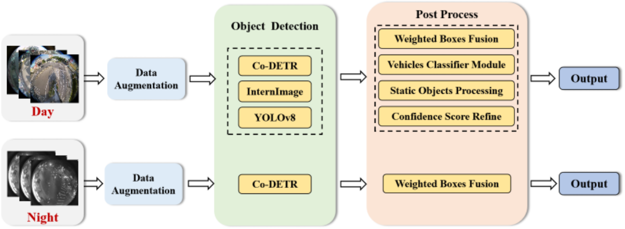


Figure 2. Overall pipeline of FE-Det. The top row shows the processing flow for the daytime scene, and the bottom row shows the processing flow for the nighttime scene.

2.2. Fisheye Object Detection

Recently fisheye cameras, owing to their large field of view (LFOV), have attracted diverse attention. Current fisheye image object detection methods can be mainly divided into two categories: distortion correction-based methods and methods based on original LFOV images. Distortion correction-based methods usually consist of two stages: image rectification and general object detection. In [20], fisheye images are transformed using Mercator projection to minimize the effects of pedestrian shape variances, followed by the application of the Viola-Jones detector for pedestrian detection. [2] firstly integrates deep learning for the detection of multi-class objects in fisheye images, and confirms the viability of approaches based on the original LFOV images. FisheyeMODNet [26] conducts end-to-end training of the network utilizing temporally sequential images that encapsulate both semantics and motion simultaneously. FisheyeDet [14] integrates distortion feature representation learning and precise bounding box refinement directly into the detection process. This innovative approach substantially enhances the generalization ability of object detection in fisheye images. FisheyeYOLO [17] designs novel forms, including curved boxes and adaptive step polygons, for fisheye image object representations. It adapts YOLOv3 [19] model to output different representations.

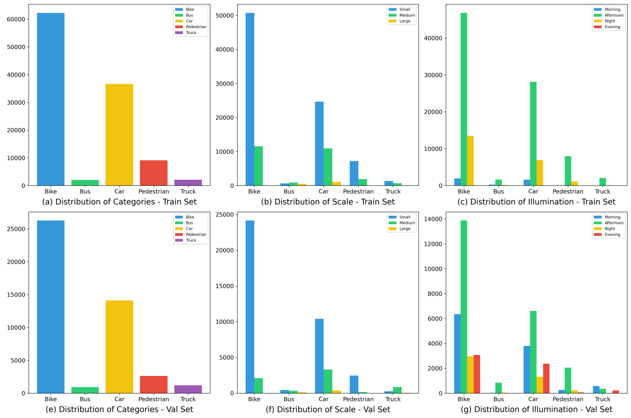
3. Methodology

In this section, we illustrate FE-Det to detect fisheye image objects. Fig. 2 shows the overall pipeline of FE-Det. We aim to maximize the detection performance in fisheye images by our enhancing strategies in three main sections: data process, ensemble detection model, and post-process.

3.1. Data Processing

3.1.1 Data Overview

2024 AI CITY CHALLENGE Track 4 is based on the FishEye8K benchmark dataset [8]. The FishEye8K benchmark dataset, presented at CVPRW23, is comprised of 5288 images for training and 2712 images for validation. The images have resolutions of 1080×1080 and 1280×1280, with a combined total of 157K annotated bounding boxes across 5 road object classes, including Bus, Bike, Car, Pedestrian and Truck. The test dataset FishEye1Keval consists of 1000 images sourced from 11 camera videos that were not employed in creating the FishEye8K dataset but similar to those in the training dataset.

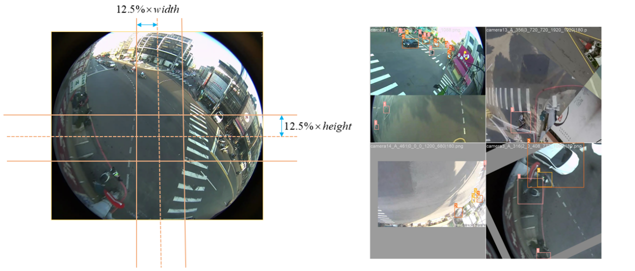


(a) Distribution of Categories - Train Set
(b) Distribution of Scale - Train Set
(c) Distribution of Illumination - Train Set
(e) Distribution of Categories - Val Set
(f) Distribution of Scale - Val Set
(g) Distribution of Illumination - Val Set

Figure 3. The distributions of objects in the FishEye8K dataset.

Fig. 1 displays visualizations of 5 road object classes observed in several streets, captured by different cameras at varying time periods. The top row displays the original images, while the bottom row represents the Ground Truth. In Fig. 1, it is evident that the fisheye view introduces shape distortion in all categories, with significant variations between distinct categories, such as pedestrians and vehicles. Moreover, among all the types of vehicles, it is particularly challenging to differentiate between cars, buses, and trucks based on their visual characteristics. Additionally, it is also observed that there is a large difference in image labeling between night and day.

Fig. 3 shows a statistical graph illustrating the quantity and size of each category, as well as the frequency of its occurrence across various time periods. Fig. 3(a) and Fig. 3(e) explicitly show that the FishEye8K dataset has an unequal distribution of categories. Fig. 3(b) and Fig. 3(f) demonstrate the presence of numerous small objects in the dataset and a significant diversity in the scales of different categories. The dimensions of bicycles and pedestrians are predominantly categorized as small to medium, with a notable absence of larger sizes. In contrast, the dimensions of buses demonstrate a more evenly distributed size spectrum. As can be seen in Fig. 3(c) and Fig. 3(g), there are also significant differences in the occurrence preferences of different categories across time. For instance, buses and trucks predominantly manifest during the day and are absent during the night, while bicycles and cars are present both during the day and at night. However, overall, the quantity of all identifiable objects in the night environment is significantly



(a) Cropping

(b) A batch of dataset following the application of data augmentation techniques.

Figure 4. Examples of image preprocessing. (a) shows the cropping scheme. (b) is a batch of images after data augmentation.

lower compared to daytime.

3.1.2 Data Augmentation

Data augmentation represents a pivotal approach within the realm of computer vision, aimed at augmenting the size and heterogeneity of training datasets. This methodology holds particular significance for endeavors related to object detection, such as the current task, where the challenges include a multitude of diminutive target objects, an expansive field of vision attributable to fisheye lenses, and suboptimal illumination conditions. To address the above difficulties, the following data enhancement methods are used in this study.

Cropping. For each image, we crop it into 4 parts, with 25% of the length or width overlapping between each part. Fig. 4(a) shows the cropping scheme.

Rotation. The fisheye view introduces both forward and reverse objects into the image. To ensure an equal distribution of forward and reverse objects, each cropped image is rotated by 90°, 180°, and 270° respectively. This adjustment aids the model in acquiring a more comprehensive understanding of the target's appearance characteristics.

Scaling. As previously stated, there is a significant disparity in the size of various targets, and scaling enables the model to more effectively learn about targets with varying dimensions.

Color manipulation and histogram equalization. Color manipulation and histogram equalization are employed to optimize the image's brightness, contrast, and saturation, hence enhancing the model's ability to recognize objects under varying lighting circumstances.

Mosaic. Mosaic stitches together multiple images into a single training example, which improves model robustness by exposing it to a wider variety of object scales, positions, and contexts in one image.

Fig. 4(b) shows a batch of images after data augmentation.

3.2. Ensemble Detection Model

The study has experimented on three object detection models: YOLOv8, Co-DETR, and InternImage. Each model possesses distinct attributes and performs well at detecting different classes of objects. The appearance characteristics of objects, such as color, vary significantly between daytime and nighttime scenes, and the standards for annotation are also not uniform. Therefore, we implement distinct training and inference processes for images taken during the day and at night.

For daytime images, since the complex variability and low resolution of traffic scenes, our proposed framework model ensemble with different models to improve performance. The object detection approach used in this paper is based on YOLOv8, Co-DETR, and InternImage. We fetch the bounding boxes of the detected objects in each image and the corresponding confidence using the detection models:

$$B_i = \{(b_{ij}, c_{ij}, s_{ij}) | i \in \nu, j \in N\} \tag{1}$$

where $b_{ij} = (\mathbf{x}_1, \mathbf{y}_1, \mathbf{w}, \mathbf{h})$ is the corresponding bounding box information, c_{ij} is the class id, s_{ij} is the confidence score, ν is the number of images and N is the number of objects in image i . We perform Weighted Boxes Fusion (WBF) to filter detection boxes that overlap the same objects. Accordingly, the final prediction extracted from the three individual models by using WBF is generally formulated as follows:

$$Z_i = \{wbf(B_{E1,i}, B_{E2,i}, B_{E3,i}) \mid i \in \nu\} \tag{2}$$

where Z represents the final prediction. $E1$, $E2$, $E3$ are YOLOv8x, Co-DETR and InternImage fine-tuned on FishEye8K dataset [21].

For images captured at night, due to the significantly fewer number of targets compared to daytime images and the virtual absence of large-sized objects like trucks and buses, we opted to fine-tune using only Co-DETR. However, during inference, we employ Test Time Augmentation (TTA), making it necessary to employ WBF for result integration.

3.3. Post Process

Weighted Boxes Fusion. WBF is an advanced technique in object detection that addresses the challenge of integrating multiple bounding boxes predicted by different models or the same model with varying configurations. After the three models output detection results, we perform WBF on them. The fusion process is calibrated by assigning variable weights to the detection outcomes from different models, because of their respective proficiency in recognizing certain categories. For instance, when detecting Bus and Truck, the InternImage model's results are given higher weights because it performs better in these categories. On the other hand, when identifying Bike and Pedestrian, the Co-DETR model's detections are prioritized due to its superior performance in these classes.

Vehicles Classifier Module. By analyzing the confusion matrix of the validation set predictions, we observed that the three vehicle categories exhibit high confusion rates. Additionally, there is a tendency to incorrectly identify the background as one of these three categories. To address this issue, we employ YOLOv8s-cls to train a classification network specifically for vehicles. We select the targets with high prediction scores but belonging to false positives as negative samples. Additionally, we extract the regions corresponding to these three categories from the training set as positive samples. Finally, we randomly partitioned all samples into training and validation sets with a ratio of 8:2.

Static Objects Processing. The static object definition is contingent upon two factors: positional stability and image region similarity. An object qualifies as static if it retains a nearly identical location in at least ten out of twenty sequential frames, corroborated by significant similarity in the relevant image area. To discern static pedestrians, which are atypical in urban scenes, the method combines static indicators with confidence scores to exclude non-dynamic entities such as utility poles. For static vehicles, intermittent detections are interpolated to maintain continuous identification. We adopt SSIM as the metric of image similarity between two images. The Structural Similarity Indexis a perceptual metric that quantifies the similarity between two images. SSIM is designed to improve upon traditional metrics like Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) by taking into account the perceptual properties of the human visual system which includes changes in structural information, luminance, and contrast. The SSIM index is a decimal value that ranges between -1 and 1. A value of 1 indicates that the two images being compared are identical. The measure between two images x and y of common size $N \times N$ is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3}$$

where μ_x and μ_y are the average pixel values of images x and y, respectively.

$$\mu_x = \sum_{i=1}^{N} w_i x_i \tag{4}$$

σ_x and σ_y are the variances of x and y, respectively.

$$\sigma_x = (\sum_{i=1}^{N} w_i (x_i - \mu_x)^2)^{\frac{1}{2}} \tag{5}$$

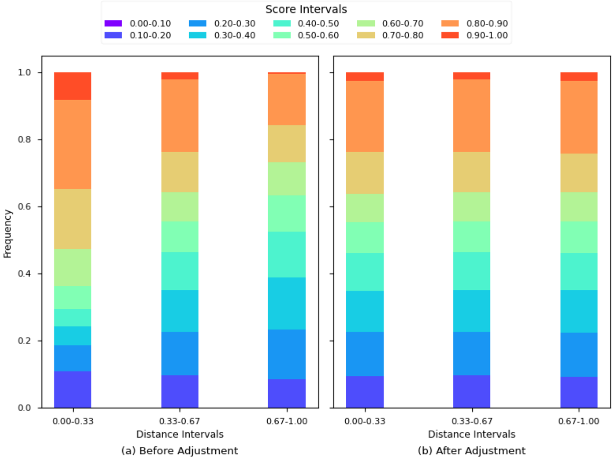


Figure 5. The frequency distribution of the confidence scores before and after adjustment. (a) shows the original distribution of confidence scores across the specified distance intervals before adjustment. (b) illustrates the frequency distribution of the confidence scores following the application of refinement techniques.

σ_{xy} is the covariance of x and y.

$$\sigma_{xy} = \sum_{i=1}^{N} w_i (x_i - \mu_x)(y_i - \mu_y) \tag{6}$$

$C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are two variables to stabilize the division with weak denominator, with L being the dynamic range of the pixel-values. $k_1 = 0.01$ and $k_2 = 0.03$ are the default parameters.

Confidence Score Refine. When conducting target detection using a fisheye camera, the distortion caused by the camera lens affects the confidence of detecting targets in different areas of the image. As a result, there are significant variations in the performance of target detection across different regions. Upon completion of the detection process, we observe that the scores decrease as the distance from the center increases from Fig. 5(a). Additionally, we notice a significant number of targets located in the surrounding edge region. Consequently, it is essential to statistically assess the confidence scores of different areas and employ correction techniques such histogram matching to modify the distribution. This method does not require modifying the initial detection model and can function as a standalone post-processing procedure to enhance the confidence distribution by examining the detection outputs.

Fig. 6 shows the flowchart of Confidence Score Refine Module. Firstly, we divide the image into three distinct regions: the central region, the intermediate annular region, and the peripheral region. For each demarcated region, the confidence scores associated with all detected objects are aggregated independently. Subsequently, an analysis of the distribution characteristics of these confidence scores

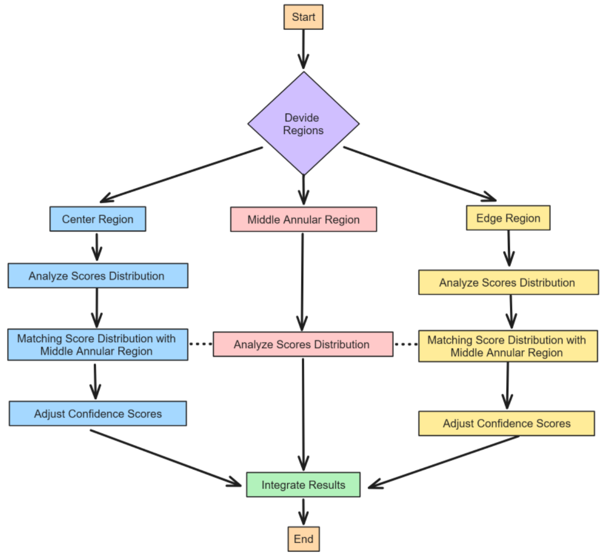

Start
Devide Regions
Center Region
Middle Annular Region
Edge Region
Analyze Scores Distribution
Analyze Scores Distribution
Matching Score Distribution with Middle Annular Region
Analyze Scores Distribution
Matching Score Distribution with Middle Annular Region
Adjust Confidence Scores
Adjust Confidence Scores
Integrate Results
End

Figure 6. The flowchart of Confidence Score Refine Module.

is conducted for each region, encompassing the calculation of their histograms [9]. The histogram of the intermediate annular region is designated as the reference distribution. Utilizing histogram matching techniques, the distributions of confidence scores in the remaining regions are adjusted to align with this reference. This adjustment of confidence scores within each region is predicated on the outcomes of the distribution matching process. Finally, the integration of these calibrated confidence scores into the final detection results.

4. Experiments

4.1. Implementation Details

We have selected Co-DETA, InternImage, and YOLOv8x as the detection networks, and YOLOv8s-cls as the network for classifying vehicles. We trained and test these models on two NVIDIA GeForce RTX 3090 24GB graphics cards.

Training phase. Since there are differences in object features and labeling standards across daytime and nighttime scenarios in the dataset, we apply different detection strategies for each. For images in daytime scenes, three models are used for training. We employ the ImageNet22K pre-trained Swin-L vision converter as the backbone for Co-DETR. The detector was pretrained using the Objects365 and MS COCO datasets. And we apply the InternImage-XL network based on the Cascade Mask R-CNN framework using DCNv3 [6] as the core operator. The detector has been pre-trained on MS COCO. Additionally, we employ the YOLOv8x model for object detection and the YOLOv8s-cls as the classification network in the training phase. The size of the input images for all detection networks is 1600×1600. The batchsize for Co-DETR, InternImage and YOLOv8 is 2, 4, 8 respectively. In contrast, the size of the input images for the classification network is 256×256 and the batchsize is 256. For the images in the night scene, only Co-DETR is used for training since trucks and buses are extremely rare. The input image size and batchsize are the same as above.

Test phase. In the inference, the TTA method is employed. We input four versions of an image into the detection networks: the original image at a resolution of 1600 × 1600 and flipped them 180°; as well as flipped versions at 90° and 270° with a resolution of 4000 × 4000. The ultimate prediction is determined by averaging the predictions generated from the augmented versions of the test data. Furthermore, we conduct a WBF process on the predictions obtained from three models. Particularly, the weights of models are not the same for different categories when fusing, because different models are good at detecting different classes.

4.2. Metrics of Evaluation

The challenge ranking is based on F1 Score [7]. It represents the harmonic mean of Precision and Recall, offering a comprehensive measure of a model's accuracy in predicting positive instances. F1 Score is the balance between precision and recall, the value ranging between 0 to 1. Higher F1 Score means a better balance on precision and recall [15].

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

where precision finds the percentage of correct predictions over false positive and true positive, which measure how accurate the prediction results are.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{8}$$

Recall defines how well the algorithm finds all the positive cases.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{9}$$

4.3. Experiments Results

Tab. 1 presents comparative results of object detection models evaluated on the validation dataset. YOLOv8 showcases strong performance in detecting Bus, Car, and Truck with AP scores of 0.693, 0.677, and 0.636 respectively. It indicates a relatively high accuracy for larger vehicle types. However, its performance on Bike and Pedestrian is lower, at 0.504 and 0.44, which could suggest a need for improved feature extraction for smaller objects. The Co-DETR models, segmented into Daytime and Night, display a mixed performance. Co-DETR-Daytime excels in detecting Pedestrian with an AP of 0.742, outperforming the other

Model	**AP_Bus**	**AP_Bike**	**AP_Car**	**AP_Pedestrian**	**AP_Truck**	**AP_0.5-0.95**	**AR_0.5-0.95**
YOLOv8x	0.693	0.504	0.677	0.440	0.636	0.590	0.698
Co-DETR-Daytime	0.535	0.685	0.474	**0.742**	0.609	0.609	0.681
Co-DETR-Night	0.486	**0.731**	0.432	0.435	0.510	0.521	0.664
InternImage	**0.770**	0.555	**0.691**	0.464	**0.788**	**0.654**	**0.702**

Table 1. Experimental results on validation dataset.

Model	**AP_0.5-0.95**	**F1 Score**
YOLOv8x	0.3102	0.4024
Co-DETR	**0.4759**	0.4743
InternImage	0.4756	0.3415
Ensemble Model	0.4089	**0.4909**

Table 2. Experimental results on test dataset.

TTA	**Night Network**	**Vehicles Classifier Module**	**Static Objects Processing Module**	**Confidence Score Refine**	**F1 Score**
					0.4909
✓					0.5329
✓		✓			0.5645
✓		✓	✓		0.5713
✓		✓	✓	✓	0.5799
✓	✓	✓	✓		0.5853
✓	✓	✓	✓	✓	**0.5883**

Table 3. Ablation results on test dataset.

models in this category, while it shows moderate performance in other categories. Co-DETR-Night, on the other hand, has its highest AP score for Bike at 0.731, suggesting an effective adaptation to low-light conditions for this category. Nonetheless, its efficacy is reduced for other categories under nighttime conditions, as seen in the other AP scores. The InternImage model shows a promising performance across all categories, with the highest AP scores for Bus at 0.77, Car at 0.691, and Truck at 0.788. Its performance on Bike and Pedestrian is also competitive at 0.555 and 0.464, respectively. This suggests a well-rounded capability in object detection tasks across varying object scales and types.

In conclusion, each model shows specific strengths in certain categories. the InternImage model consistently performs at a high level for vehicle types and pedestrians, while Co-DETR performs for small object detection such as pedestrians and Bike. This analysis shows that combining the individual models can provide a powerful solution for a variety of object detection scenarios.

Tab. 2 shows the AP_0.5-0.95 and F1 scores on the test set for YOLOv8x, Co-DETR, InternImage, and the Ensemble Model. YOLOv8x shows lower performance in both metrics compared to the other models, with an AP_0.5-0.95 of 0.3102 and a F1 Score of 0.4024. This might suggest that while it is capable of detecting objects, its robustness is relatively lower. Co-DETR displays significantly higher performance, with an AP_0.5-0.95 of 0.4759, indicating better precision across varying levels of IoU thresholds. It also has a high F1 Score of 0.4743, suggesting a good balance between precision and recall. InternImage has an AP_0.5-0.95 similar to Co-DETR at 0.4756, which denotes it has almost equivalent precision across IoU thresholds as Co-DETR. However, its F1 Score is substantially lower at 0.3415, indicating that either precision or recall, or both, are not as balanced as Co-DETR. Ensemble Model shows a mixed performance with a lower AP_0.5-0.95 of 0.4089 compared to Co-DETR and InternImage but has the highest F1 Score of 0.4909 among all models. In fact, after applying the Ensemble Model, many false positives can be filtered out. However, it also filters out some truth positives with lower confidence scores. Consequently, the AP_0.5-0.95 decreases. Nevertheless, this approach leads to a more balanced precision and recall rate, resulting in more reasonable detection outcomes. This suggests that the ensemble method, which typically combines the strengths of multiple detection models, has achieved a better balance of precision and recall, leading to a higher overall F1 Score.

Tab. 3 shows the ablation results on test dataset. Our ablation study systematically investigates the impact of each component on the overall performance of our system. Initially, we observe a baseline score of 0.4909, indicating the performance without the integration of any of the proposed modules.

TTA. With the addition of TTA, the score is increased to 0.5329, which can improve the performance and robustness of the model during the testing phase by applying data augmentation and integrating the prediction results.

Night Network Integration. Nighttime network integration refers to training a specific set of Co-DETR weights designed specifically for dark night scenes. By fusing the prediction results from the night network, the score rises to 0.5645, which indicates that using different prediction weights for images at different time improves the accuracy of detection.

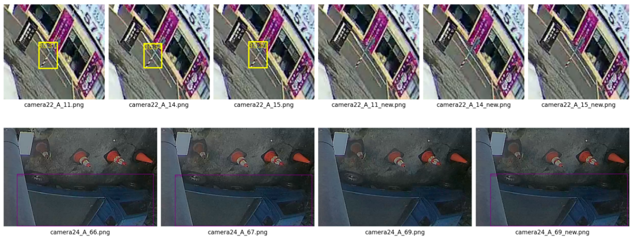


Figure 7. Visualization of the effect of the Static Objects Processing module.

Vehicles Classifier Module. In our self-curated classification dataset on vehicles, the YOLOv8s-cls achieves a top-1 accuracy of 98.5%. As analyzed above, trucks and buses are rarely present at night. Therefore, considering both inference time and classification accuracy, we will only utilize the Vehicles Classifier Module in daytime scenes and refrain from using it during nighttime scenes. Tab. 3 shows that when we introduce the Vehicles Classifier Module, designed to distinguish between different vehicle types and reduce confusion, there is a noticeable improvement to 0.5713. This suggests that the module effectively reduces misclassification among the vehicle categories and between vehicles and background classes.

Static Objects Processing. The integration of the Static Objects Processing Module, which discerns stationary entities, refines the detection algorithm by excluding inert false positives and incorporating overlooked false negatives, thereby marginally elevating the accuracy metric to 0.5799. Fig. 7 delineates the module's efficacy. The initial triplet of frames in the upper tier erroneously categorizes utility poles as pedestrians, evidenced by their diminished confidence scores, leading to their subsequent exclusion post-processing. Conversely, the trailing triad illustrates the module's rectifications. In the second row, a truck located at camera24_A_69.png was neglected, despite its identification in antecedent frames and considerable visual congruence with the preceding imagery. After applying the module, the truck is consequently integrated into the detection output of camera24_A_69.png.

Our proposed method has been submitted to the AI City Challenge 2024 Track4 for evaluation. As shown in Tab. 4, our method surpassed many methods with a score of 0.5883 and ranked sixth out of 52 teams from all over the world. Fig. 8 shows the visualization of the final submission results, from which it can be observed that our detection results are quite accurate.

5. Conclusion

In this paper, we analyze some problems that need to be solved urgently in the fisheye detection issue, e.g., tiny objects, shape distortion and confusion of similar classes. Aiming at these, we propose an advanced object detection framework named FE-Det based on Co-DETR, YOLOv8 and InternImage. The whole framework is mainly enhanced from object detection models and post-processing. Extensive experiments demonstrate the effectiveness of our method. In the future, we would conduct experiments other fisheye datasets [16] to verify the scalability of our method.

Rank	Team ID	Team Name	Score
1	9	VNPT AI	0.6406
2	40	NetsPresso	0.6196
3	5	SKKU-AutoLab	0.6194
4	63	UIT_ AICLUB	0.6077
5	15	SKKU-NDSU	0.5965
6 (ours)	**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 4. Top 10 Leaderboard of Track4 in the AI City Challenge 2023.

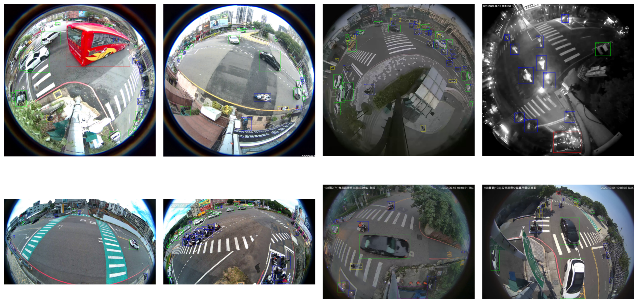


Figure 8. Visualization of the final submission. Red, blue, green, yellow, and purple rectangular boxes represent Bus, Bike, Car, Pedestrian, and Truck, respectively.

6. Acknowledgements

This work is supported by the Chinese National Natural Science Foundation under Grants 62076033.

References

[1] Armstrong Aboah, Bin Wang, Ulas Bagci, and Yaw Adu-Gyamfi. Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5350–5358, Vancouver, BC, Canada, 2023. IEEE. 2

[2] Iljoo Baek, Albert Davies, Geng Yan, and Ragunathan Raj Rajkumar. Real-time Detection, Tracking, and Classification of Moving and Stationary Objects using Multiple Fish-

eye Images. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 447–452, 2018. 3

[3] J. P. Barreto. Unifying Image Plane Liftings for Central Catadioptric and Dioptric Cameras. In *Imaging Beyond the Pinhole Camera*, pages 21–38. Springer Netherlands, 2006. 1

[4] Ahmed Ben Saad, Gabriele Facciolo, and Axel Davy. On the Importance of Large Objects in CNN Based Object Detection Algorithms. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 533–542, 2024. 2

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection, 2017. 2

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. 6

[7] Tran Hiep Dinh, Cong Hieu Le, and Quang Ha. UAV Imaging: Correlation Between Contrast and F1-Score for Vision-Based Crack Detection. In *2024 IEEE/SICE International Symposium on System Integration (SII)*, pages 657–662, 2024. 6

[8] Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Erkhembayar Ganbold, Jun-Wei Hsieh, and Ming-Ching Chang. FishEye8K: A Benchmark and Dataset for Fisheye Camera Object Detection, 2023. 1, 3

[9] Robert A. Hummel. Histogram modification techniques. *Computer Graphics and Image Processing*, 4(3):209–224, 1975. 6

[10] Shreya Jain, Samta Gajbhiye, Achala Jain, Shrikant Tiwari, and Kanchan Naithani. A Quarter Century Journey: Evolution of Object Detection Methods. In *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–8, Bhilai, India, 2024. IEEE. 2

[11] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2023. 2

[12] Hyun-Ki Jung and Gi-Sang Choi. Improved YOLOv5: Efficient object detection using drone images under various conditions. *Applied Sciences*, 12(14):7255, 2022. 2

[13] Benjamin Kiefer, Lojze Zust, Matej Kristan, Janez Pers, and Matija Tersek. 2nd Workshop on Maritime Computer Vision (MaCVi) 2024: Challenge Results. 2

[14] Tangwei Li, Guanjun Tong, Hongying Tang, Baoqing Li, and Bo Chen. FisheyeDet: A Self-Study and Contour-Based Object Detector in Fisheye Images. *IEEE Access*, 8:71739–71751, 2020. 1, 3

[15] Martinus Grady Naftali, Jason Sebastian Sulistyawan, and Kelvin Julian. Comparison of Object Detection Algorithms for Street-level Objects, 2022. 6

[16] Saravanabalagi Ramachandran, Ganesh Sistu, Varun Ravi Kumar, John McDonald, and Senthil Yogamani. Woodscape Fisheye Object Detection for Autonomous Driving – CVPR 2022 OmniCV Workshop Challenge, 2022. 8

[17] Hazem Rashed, Eslam Mohamed, Ganesh Sistu, Varun Ravi Kumar, Ciarán Eising, Ahmad El-Sallab, and Senthil Yogamani. FisheyeYOLO: Object Detection on Fisheye Cameras for Autonomous Driving. 1, 3

[18] Hazem Rashed, Eslam Mohamed, Ganesh Sistu, Varun Ravi Kumar, Ciaran Eising, Ahmad El-Sallab, and Senthil Yogamani. Generalized Object Detection on Fisheye Cameras for Autonomous Driving: Dataset, Representations and Baseline. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2271–2279, Waikoloa, HI, USA, 2021. IEEE. 1

[19] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107: 104117, 2021. 3

[20] Jae Kyu Suhr and Ho Gi Jung. Rearview Camera-Based Backover Warning System Exploiting a Combination of Pose-Specific Pedestrian Recognitions. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1122–1129, 2018. 3

[21] Duong Nguyen-Ngoc Tran, Long Hoang Pham, Hyung-Joon Jeon, Huy-Hung Nguyen, Hyung-Min Jeon, Tai Huu-Phuong Tran, and Jae Wook Jeon. Robust Automatic Motorcycle Helmet Violation Detection for an Intelligent Transportation System. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5341–5349, Vancouver, BC, Canada, 2023. IEEE. 4

[22] 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, 2022. 2

[23] 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

[24] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, and Zhiqi Li. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions, 2023. 2

[25] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 2

[26] Marie Yahiaoui, Hazem Rashed, Letizia Mariotti, Ganesh Sistu, and Ian Clancy. FisheyeMODNet: Moving Object detection on Surround-view Cameras for Autonomous Driving, 2019. 3

[27] Zhuofan Zong, Guanglu Song, and Yu Liu. DETRs with Collaborative Hybrid Assignments Training, 2023. 2