Enhancing Road Object Detection in Fisheye Cameras: An Effective Framework Integrating SAHI and Hybrid Inference

Bao Tran Gia [1,2], Tuong Bui Cong Khanh [1,2], Hien Ho Trong [1,2]
Thuyen Tran Doan [1,2], Tien Do [1,2], Duy-Dinh Le [1,2], Thanh Duc Ngo [1,2]

[1] University of Information Technology, VNU-HCM, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
{22520121, 22521624, 22520414}@gm.uit.edu.vn
{thuyentd, tiendv, duyld, thanhnd}@uit.edu.vn

Abstract

Fisheye cameras are extensively employed in surveillance systems because they provide a broad viewing angle, enhancing visibility. The reception of an image from a wide perspective can result in distortion, posing challenges for recognition systems, mainly when dealing with moving objects, as observed in traffic systems. This work presents an effective framework comprising multiple modules to address the issue of small objects and rapidly changing viewing perspectives in fisheye camera data. First, we use Slicing Aided Hyper Inference (SAHI), an algorithm that uses generic slicing-aided inference to deal with small objects. Second, we integrate the outcomes of CNN (YOLO) and state-of-the-art Transformer (Co-DERT) detection methods to utilize the respective strengths of each strategy for handling data limitations. This approach has demonstrated promising performance, achieving an F1 score of 0.6077 and achieving the 4^{th} in Track 4 of the AI City Challenge 2024.

1. Introduction

As cities grow, the demand for efficient traffic surveillance and management systems becomes increasingly urgent. Intelligent Transportation Systems (ITS) are crucial in enhancing road safety, optimizing traffic flow, and ensuring overall urban mobility. Road object detection stands out as a crucial problem for study and innovation among the various difficulties that ITS encounters. Fisheye cameras, known for their broader coverage, offer distinct advantages in ITS by reducing the need for numerous individual cameras. This broad coverage allows for more comprehensive traffic monitoring and management. However, a significant drawback of fisheye cameras is the introduction of distortions, which create difficulties when detecting vehicles and pedestrians in traffic. Addressing this issue can significantly contribute to traffic management and congestion control, making it the main focus of Track 4 [13] in the AI City Challenge 2024. In this track, participants are tasked with harnessing the capabilities of machine learning/deep learning technology to develop robust and precise object detection models. These models are expected to accurately determine the location and classify road objects such as buses, bikes, cars, pedestrians, and trucks. Furthermore, these models are expected to be able to compensate for the distortions introduced by fisheye cameras under various conditions, i.e., daylight, nighttime, varying camera angles, or different image resolutions.

Object detection methodologies have traditionally been developed with a focus on perspective cameras. However, their effectiveness is compromised when applied to images captured by fisheye cameras, primarily due to the significant distortion these cameras introduce [6]. Furthermore, the strategic placement of security cameras at higher vantage points with diverse angles significantly impacts surveillance. While it inadvertently leads to reduced video resolution, it also simultaneously introduces variations in the proportions and sizes of the objects captured and viewing perspective. An additional challenge arises from the fact that images can be captured day and night. These factors can further complicate the task of object detection, necessitating the development of methodologies that are robust to varying lighting conditions. To handle object detection, transformer-based models have recently outperformed CNN-based [8], [9], [12]. Transformer models [17] have demonstrated superior performance in various benchmarks. However, CNN-based models continue to perform well when detecting small objects or handling data limitations [4].

In this study, we present an effective approach for road

object detection in fisheye cameras. During the training phase, our methodology leverages a data augmentation technique known as *random scaling* to address the viewing perspective challenge in camera images, a critical aspect of object detection. By randomly scaling images in the training dataset, we equip our model with various perspectives, enhancing its ability to generalize across different scales. To handle small objects near the image periphery due to fisheye camera distortion, we introduce an additional loss function that places greater emphasis on these objects. Additionally, we introduce a technique that partitions the original image into sub-images, conducts inference on each, and then combines their predictions. Finally, we employ an advanced ensemble method called Weighted Boxes Fusion (WBF) [10], which combines the predictions from each different strategy. Finally, we employ an advanced ensemble method called Weighted Boxes Fusion (WBF) [10], which combines the predictions from each different strategy. This algorithm comes with a merging strategy to use the confidence scores of all proposed bounding boxes to construct the average boxes. As a result, the ensemble results exhibit greater precision than those derived from the individual models due to their ability to optimize the strengths of each model while simultaneously reducing its weaknesses. This leads to a notable enhancement in the overall quality of our system. Finally, the ensemble predictions are further refined through a post-processing strategy.

Our proposed framework has demonstrated its effectiveness by securing a commendable position within the top four contenders in Track 4. This achievement is underscored by an impressive **0.6077** F1 score, which is considerably higher than the **0.5965** F1 score of the top 5 teams, highlighting the robust performance of our model.

2. Related Works

In this section, we provide the reader the landscape encompassing fisheye images and their associated datasets, object detection methodologies, ensemble techniques, and the latest strategies for small object detection.

2.1. Fisheye Camera Datasets

Fisheye images are a distinct category of wide-angle photographs produced using a fisheye lens that yields a hemispherical and distorted perspective with a field of view often exceeding 180 degrees. This unique feature produces a curvilinear distortion, bestowing images with a signature "fisheye" look. Below, we highlight several notable fisheye datasets such as FishEye8K and WoodScape, along with their own characteristics.

The FishEye8K [5] dataset, which forms the foundation for this challenge, comprises 8,000 annotated images of varied dimensions. This dataset is characterized by approximately 157K bounding boxes, presenting one of these five categories: 'buses', 'bikes', 'cars', 'pedestrians', and 'trucks'. Detailed distribution of image sizes is provided in Table 1. The FishEye1KEval dataset is utilized for testing purposes. Additionally, Figure 1 visually illustrates the diverse time of day captured in the images.

Table 1. The distribution of instances across various classes within the FishEye8K dataset.

ID	Class	Number of Instances
0	Bus	2,984
1	Bike	88,531
2	Car	50,749
3	Pedestrian	11,759
4	Truck	3,335

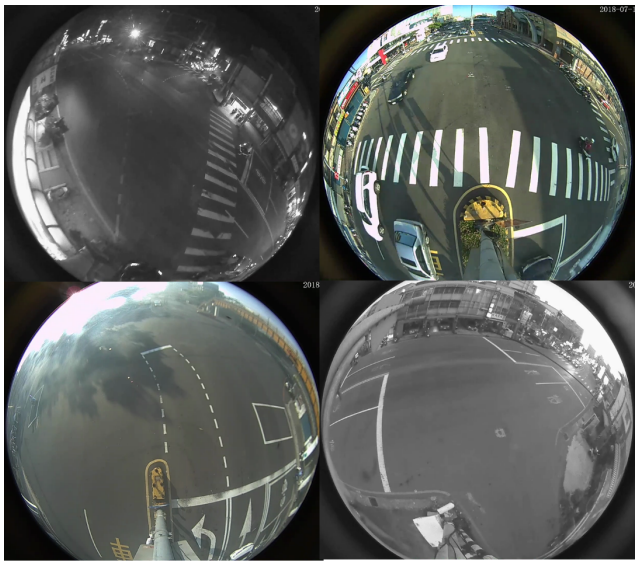

Figure 1. An illustration of the FishEye8K dataset, the dataset contains images captured during both daytime and nighttime conditions.

The WoodScape dataset [14], comprising 8,234 fisheye images, is curated for autonomous driving research with diverse camera perspectives. It facilitates training models to tackle fisheye lens distortion challenges, marking a significant advancement in this field. Researchers benefit from its utility in algorithm development and testing, enhancing autonomous driving technology.

2.2. Object detection

Object detection, a pivotal task in computer vision, entails identifying and localizing objects within digital images or videos. The primary goal of object detection algorithms is to determine whether there are any instances of semantic objects of a certain class, such as humans, cars, or animals,

in a given image or video. If these instances are present, the algorithms will localize these objects, typically providing a bounding box around the object instance.

The advent of Convolutional Neural Networks (CNNs) [8] has transformed the domain of object detection, with the emergence of numerous detectors such as Faster-RCNN [9], Cascade RCNN [3], YOLOR [12], and YOLOv6 [7]. Among these, the YOLO series has introduced a unique approach that employs a predefined set of bounding boxes, termed 'anchor boxes.' These anchor boxes are designed to encapsulate objects of diverse shapes and sizes. The model optimizes its predictions by selecting the anchor box that exhibits the highest alignment with the object's shape and size, thereby augmenting the precision of its location and size prediction. Moreover, YOLO's one-stage model design expedites training and inference processes, ensuring impressive performance.

The introduction of Transformer models [11] has significantly reshaped the landscape of computer vision, with the emergence of Co-DETR [17], an innovative training scheme that amplifies the efficiency and effectiveness of DETR-based detectors. This scheme strengthens the learning capacity of the encoder in end-to-end detectors by instructing multiple parallel auxiliary heads under the guidance of one-to-many label assignments.

2.3. Ensemble

In object detection, ensemble box algorithms have gained significant attention due to their effectiveness in improving model accuracy and robustness. Notably, Non-Maximum Suppression (NMS) is a widely used post-processing technique that eliminates less significant bounding boxes with high overlap, ensuring each object is detected only once and enhancing model precision. Beyond NMS, other ensemble box algorithms exist, such as the WBF [10] method. The WBF method iteratively updates a fused box using confidence scores, enhancing precision by effectively managing overlap between predicted boxes. This strategy leads to notable improvements in object detection model accuracy.

2.4. Slicing Aided Hyper Inference

Slicing Aided Hyper Inference (SAHI) [1] is a simple but effective framework that can be plugged into any object detection model. This methods get the idea from viewing the original image through multiple viewing perspective. Inspired by examining the original image from multiple perspectives, SAHI divides the image into overlapping patches, each undergoing independent inference procedures. This approach enhances model resilience and prediction accuracy by detecting patterns overlooked when viewing the image as a whole through a multi-dimensional inference strategy.

3. Methodologies

3.1. System Overview

The pipeline of our proposed solution is depicted in Figure 2. Initially, the original dataset is organized into multiple splits. Subsequently, various models are trained on these datasets. These models are then subjected to inference using SAHI techniques. The predictions are merged using an ensemble algorithm such as WBF. Finally, a post-processing strategy is applied to the final predictions to filter out unsatisfied predictions.

3.2. Object Detectors

During our experiments, we observed a significant class imbalance in the dataset. This phenomenon could potentially skew our models' performance, as they may overfit the overrepresented classes and vice versa. Addressing this issue is crucial to ensuring the robustness and generalizability of our object detection models across diverse scenarios.

Our approach was to leverage both anchor-based models and DETR-based models. Anchor-based models like YOLOR and YOLOv6L6 detect objects using predefined anchor boxes at various scales and aspect ratios. Furthermore, these models employ Varifocal loss [15], addressing class imbalance in training to enhance performance by adequately representing all classes during learning. In contrast, DETR-based models utilize a Multi-scale Adapter to construct a feature pyramid, which makes Co-DETR robust for detecting objects at various scales. Additionally, by eliminating the NMS post-processing step, DETR-based inference speed is significantly reduced.

3.3. Distance-Aware Loss (DAL)

Figure 3 presents the distribution of road objects in the FishEye8K dataset at varying normalized distances from the image's center point. A notable observation from this figure is the prevalence of objects situated at a considerable distance from the camera's center. This pattern suggests a potential enhancement to our model's default loss function. Specifically, it indicates the need for an increased sensitivity towards objects farther from the camera's center.

To address this issue, we propose modifying our model's IoU loss function. Our solution involves splitting the camera image into grid boxes, each assigned pre-calculated attention values to our loss function, as illustrated in Figure 4. The Generalized Intersection over Union (GIoU) loss is a popular choice for object detection tasks due to its ability to consider both the shape and position of the bounding boxes. However, in our case, we need the IoU loss function to be more sensitive to objects located farther from the camera's center. The modified the GIoU loss is as follows:

$$DAL = (1 - GIoU) \times \alpha^{v[idx]} \quad (1)$$

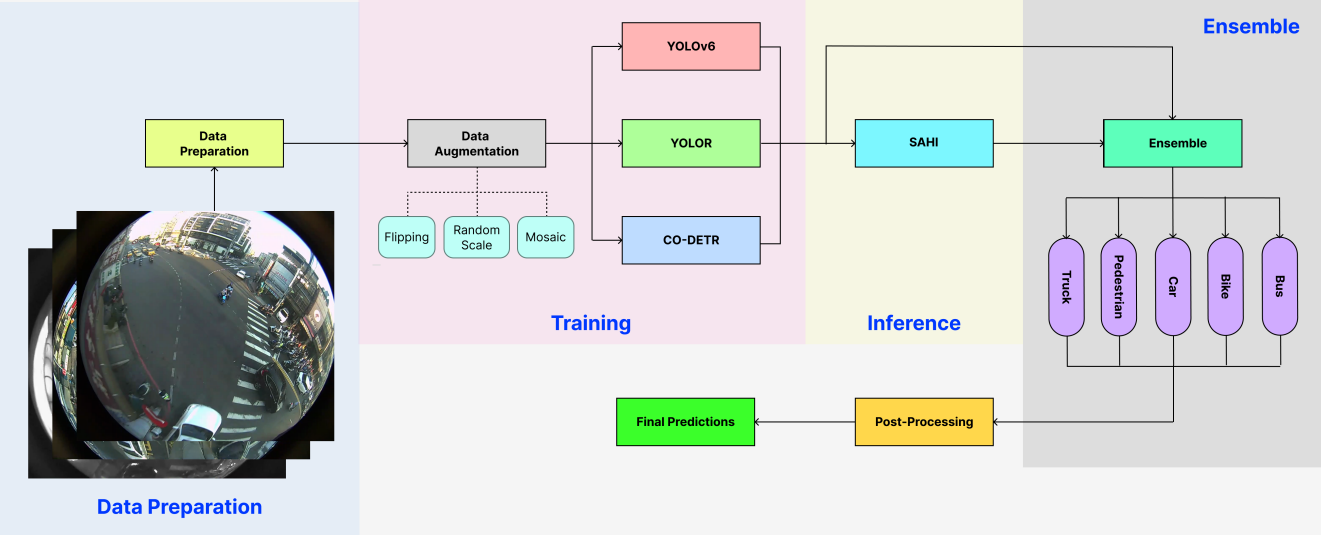

Figure 2. The overview pipeline of our system. The process begins with the data preparation block, which pre-processes the original dataset into multiple splits. Following this, the training block applies data augmentation techniques such as flipping, random scaling, and mosaic to enhance the prepared data. This augmented data is then used to train models, including YOLOv6, YOLOR, and Co-DETR. Once trained, these models undergo inference with SAHI. The ensemble block integrates the predictions from the trained models, both with and without SAHI. Finally, a post-processing strategy is applied to each prediction from the ensemble results to generate the final predictions.

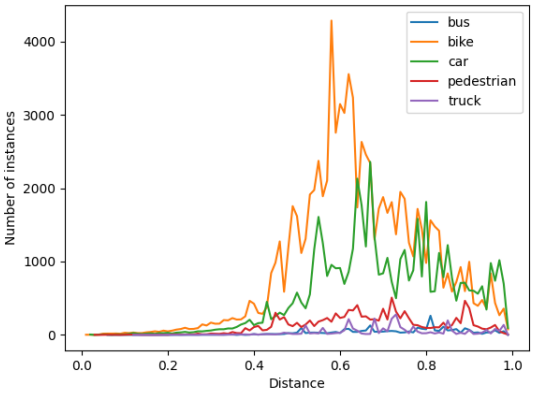

Figure 3. Distributions of the number of road object instances based on their normalized distances from the center of the image.

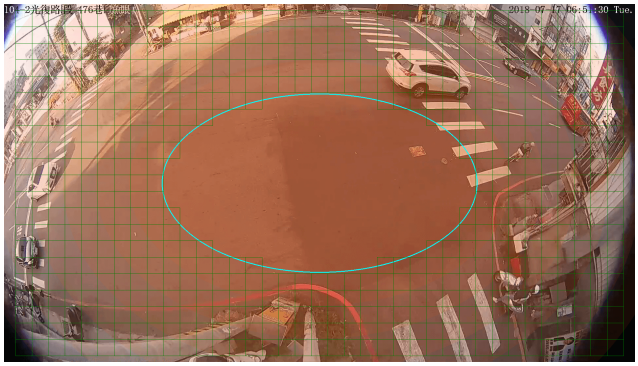

Figure 4. The illustration of our proposed class structure. Each box represents a pre-calculated attention value for our loss function. The opacity of the box is inversely proportional to the attention of our loss, meaning higher opacity indicates less attention. For every prediction made, a box encompassing its center is assigned, and the corresponding loss is added to the original loss.

The variable v represents a list of pre-computed attention values, each corresponding to a grid box in the camera image. For objects outside an ellipse defined by half the image width and height, the attention values are calculated based on the ratio of the distance from the image center to the grid box center and the distance from the image center to the image boundary. For objects within this ellipse, the attention values are set to 0. The index idx is used to fetch the corresponding attention value from v, and α is a modulating factor that balances the GIoU loss with an additional loss term.

3.4. Ensemble

In our approach, we combined the predictions from our various models using WBF. Unlike NMS, which removes redundant bounding boxes, WBF uses the confidence scores of all proposed bounding boxes to construct averaged boxes.

Figure 5 provides a comparative analysis of box aggregation for a bike object using NMS and WBF. The figure delineates the disparities in the aggregation of bounding boxes when employing these two techniques. For the bike object, there are two predictions: one capturing a person on a bike

and the other capturing the bike itself. Intuitively, the prediction capturing the bike should be retained while the other should be discarded. However, if the prediction captures the person with a higher confidence score, NMS will retain only this prediction, potentially leading to a false positive. In contrast, the WBF method can construct an "averaged" prediction that accurately represents the true positive.

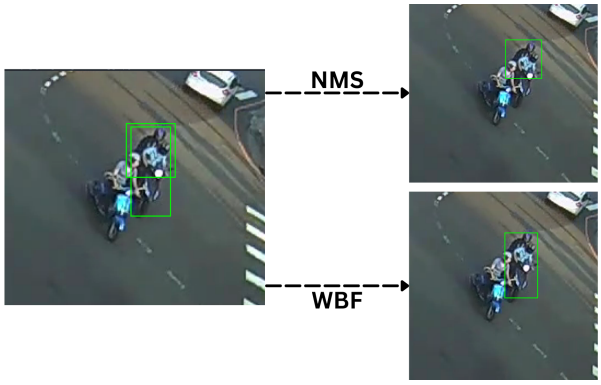

Figure 5. Comparison of box aggregation using NMS and WBF methods

3.5. Post-Processing

Small objects like pedestrians and bikes are often more challenging to detect. Factors such as occlusion, where a part of the object is hidden from view, and background clutter, where the object blends with the surrounding environment, contribute to this difficulty. These factors can reduce the distinct features the detection algorithm can use to identify these objects. Therefore, a lower confidence threshold is necessary to ensure the system is sensitive enough to detect these objects, even under challenging conditions.

On the other hand, large objects like cars, buses, and trucks are generally easier to detect due to their distinct features. However, this can lead to higher false positive rates. To mitigate this, we set a higher confidence threshold for larger objects, ensuring that only the most certain detections are considered. Thus, false positives are reduced, and prediction precision is maintained.

4. Experiments

4.1. Dataset

4.1.1 Training Datasets

In this challenge, we primarily utilized the FishEye8K dataset provided by the organizers and the WoodScape dataset. We used the WoodScape dataset for selected few pre-trained models, which allowed them to learn from a broader range of scenarios and conditions, improving their ability to generalize to unseen data.

4.1.2 Data Preparation

Initially, the dataset is partitioned using a 9:1 split ratio, where 90% of the data is allocated for training purposes, and the remaining 10% is reserved for validation. The algorithm for data sampling is shown in Algorithm 1. In this algorithm, the dataset is partitioned according to the unique identifiers of the camera and time. Specifically, for each distinct camera ID and captured time (denoted by A, M, E, N), a 9:1 split is implemented. This strategy ensures a balanced representation of each camera and time in both the training and validation sets, thereby enhancing the robustness of the model to variations across different cameras and times.

Algorithm 1: Data Sampling Algorithm

Input : Images with attribute camera ID, time (A, M, E, N)
Output: The training and validation dataset
$D \leftarrow \{\}$;
$T \leftarrow \{\}$; // Train dataset
$V \leftarrow \{\}$; // Val dataset
foreach $p \in images$ **do**
  $D[p.cam][p.time] \leftarrow D[p.cam][p.time] \cup p$
end
foreach $k \in D.keys$ **do**
  foreach $z \in [\text{"}A\text{"}, \text{"}M\text{"}, \text{"}E\text{"}, \text{"}N\text{"}]$ **do**
    $T \leftarrow (90\% \text{ of } D[k][z]) \cup T$
    $V \leftarrow (10\% \text{ of } D[k][z]) \cup V$
  end
end
return T, V

Day - Night Splitting: To further enhance the diversity of our training set, we implemented a day-night splitting strategy. This strategy involves segregating the training data based on the time of capture, allowing our models to learn and adapt to the distinct features and challenges presented by day and night conditions, such as variations in lighting and shadows.

The overview of the data we used for training and validating the performance of our models is shown in Table 2. This table provides a detailed breakdown of the number of images in each subset of our data, including the training and validation datasets and the day and night splits.

Data Augmentation: Is a powerful strategy used in machine learning to increase the robustness of the models. In this study, we employ the following augmentation techniques:

- **Flipping:** In our dataset analysis, we observed that many cameras were positioned at intersections. This is a crucial observation, as intersections are often the sites of complex traffic patterns.
- **Random Scale:** This technique, which involves training

Table 2. The overview of our dataset we used in training and testing

Dataset	M + A	E + N	Total
FishEye8K	5,841	2,159	8,000
FishEye8KEval	902	98	1,000
Train	4,679	1,728	6,407
Validation	1,162	431	1,593
Train - Day	6,068	0	6,068
Val - Day	675	0	675
Train - Night	0	2,060	2,060
Val - Night	0	227	227

at multiple scales, is designed to capture a wide range of details. Doing so can significantly improve the model's performance across different perspectives, enhancing its robustness and generalization ability.

- **Mosaic [2]:** A data augmentation strategy that combines four training images into one, providing a more diverse training sample. Therefore, it enhances its ability to generalize from the learned patterns.

4.2. Implementation Details

We conduct our experiments for the AI City Challenge's Track 4 using two NVIDIA RTX 3090 Ti 24GB graphics cards.

Co-DETR: We employed the Co-DETR model (Swin-L backbone) pre-trained on the Objects365 dataset and subsequently fine-tuned on the COCO dataset. We used this model with its default settings to train on our datasets, which comprised a *9:1 split* and a day-night dataset.

Furthermore, we leveraged this initial model for *10 epochs* of pre-training on the WoodScape dataset, followed by fine-tuning the *9:1 split* using the default setting. This enabled us to compare the model's performance when pre-trained on different datasets.

We have also substituted the Generalized Intersection over Union (GIoU) loss, which was conventionally utilized as the default loss function in the training of Co-DETR, with a loss function that we have proposed. Then, we fine-tune this model on the *9:1 split*. In this experiment, we chose modulating factor $\alpha = 2$.

YOLO series: In this study, we utilized YOLOv6-L6 and YOLOR-D6 models. We modified the default YOLO loss function to Varifocal Loss. To assess performance across image sizes, we trained the models on *1280* and *1920* dimensions for *100 epochs* and used a Non-Maximum Suppression (NMS) *threshold of 0.65* for the final prediction.

SAHI: The test dataset utilized for evaluating our model performance encompasses a variety of image sizes, predominantly large. Because we trained YOLO series models as large training image sizes, applying the SAHI strategy becomes redundant for these models. However, for the inference of the Co-DETR model, which is trained with random multi-scale augmentation, the SAHI strategy needs to be employed with a patch *width size of 1280*, maintaining the aspect ratio for the patch height and an *overlap ratio of 0.25*.

Ensemble: Our observations indicate that small objects often have low confidence scores. To address this, we conduct inference on the test dataset for each model in our chosen ensemble, setting the *confidence threshold of 0.1*. This approach prevents the premature exclusion of small objects due to their lower scores, thereby ensuring their adequate representation. Subsequently, we employ WBF with an *IoU threshold of 0.65* to combine the outputs from these models.

Post-Processing: The post-processing strategy is subsequently employed to derive the final prediction. A unique confidence threshold is utilized for each category to filter the predictions effectively. The confidence thresholds are chosen through a visualization process to determine the appropriate threshold for each category.

4.3. Experimental Results

Metrics: The evaluation metric we used to evaluate our models and used for leaderboard ranking is F1, which offers a balanced perspective by considering both precision and recall.

Evaluation Dataset: In our experimental evaluation, we employed approximately 800 images sourced from the validation subset of a *9:1 split* data partition as a testing dataset, *confidence score of less than 0.4* is excluded from the prediction evaluation process without applying a post-processing strategy.

Table 3. The evaluation result and FPS of trained models

Model	Dataset	Size	F1	FPS
Co-DETR	9:1 split	Random	0.7392	1.1
Co-DETR	day-night	Random	0.7401	
Co-DETR	WoodScape 9:1 split	Random	0.7404	
Co-DETR DAL	9:1 split	Random	0.7390	
YOLOR	9:1 split	1920	0.7306	9.8
YOLOR	9:1 split	1280	0.7335	
YOLOR	day-night	1920	0.7332	
YOLOv6-L6	9:1 split	1920	0.7364	
YOLOv6-L6	9:1 split	1280	0.6901	
YOLOv6-L6	day-night	1920	0.7370	

We noticed that the Co-DETR model's performance im-

proved when we pre-trained it using the WoodScape dataset and then fine-tuned it on a 9:1 split. This model outperformed the models that were pre-trained using COCO. Our finding suggests that using a dataset with a similar distribution to the target data can effectively enhance the model's performance.

Distance-Aware Loss: As demonstrated in Table 3, the Co-DETR-DAL model, which is trained using our proposed loss function, exhibits performance equivalent to that achieved with the original loss function. This parity can be attributed to several factors. Firstly, our proposed loss function integrates modulating factors that adjust each term's contribution to the loss function, reflecting the specific characteristics of the training data. Secondly, due to the time constraints of our experiments, extensive experimentation and fine-tuning of the proposed loss function were not feasible. Finally, in our proposed methodology, the additional losses are solely applied to the Intersection over Union (IoU) loss function, suggesting the potential for application to the box loss function. Given more time and resources, further optimization of our loss function parameters could enhance performance beyond the original loss function.

Day-Night Training: As indicated in Table 3, the day-night split training also shows potential for performance improvements. This observation suggests that the model learns distinctive features from day and night data, enhancing its ability to generalize across different lighting conditions. The day-night split training approach thus not only enriches the model's learning but provides a more comprehensive understanding of the data, leading to more accurate and reliable predictions. The comparison between the models trained on all data and trained on the day-night split is shown in Figure 6.

Figure 6. The figure illustrates the detection capabilities of two models - one trained on a comprehensive dataset (left) and the other on a day-night split dataset (right). Despite employing the same confidence threshold, the model trained on the day-night split dataset successfully identifies a car which the model trained on the full dataset fails to achieve

High-Resolution Training: High-resolution training significantly boosts YOLO series models' performance, notably for small to medium-sized objects. YOLOv6-L6 benefits notably from image size increase (*1280* to *1920*), while YOLOR shows stable performance across varied resolutions.

Inference with SAHI: We employ the SAHI technique on each trained Co-DETR model in the inference process. As shown in Table 4, the integration of the SAHI technique during the inference stage influences the performance of each Co-DETR model. Although there is a noticeable decrease in the performance of individual models when evaluating on the validation dataset, it is observed that the utilization of results from the SAHI technique can effectively enhance the overall performance.

Table 4. Comparative evaluation of trained models with and without the SAHI technique in the inference stage

Model	Dataset	F1	+ SAHI
Co-DETR	9:1 split	0.7392	0.7265
Co-DETR	day-night	0.7401	0.7294
Co-DETR-DAL	9:1 split	0.7390	0.7255
Co-DETR	WoodScape 9:1 split	0.7404	0.7277

Ensemble: As delineated in Table 5, we have selected certain models for the ensemble. Our observations indicate that all Co-DETR models consistently outperform the YOLO models, thereby leading us to incorporate all Co-DETR models into the ensemble. In the case of the YOLO series models, we have restricted our selection to those trained on the day-night dataset and those trained on a *9:1 split* with a training image size of *1920*. For model inference with SAHI, we only used Co-DETR pre-trained on WoodScape for the ensemble. The performance of some methods used for ensemble outputs is shown in Table 5. The WBF method was observed to outperform other methods.

As depicted in Table 6, the performance of ensembled various Co-DETR models with WBF is evaluated. These models include those trained on a 9:1 split, under day-night conditions and those pre-trained on WoodScape and subsequently fine-tuned on a 9:1 split. The table further illustrates

Table 5. Comparison of F1 scores for NMS, Soft-NMS, NMW, and WBF ensemble methods before and after the application of post-processing techniques.

Method	F1	+ Post-Processing
NMS	0.7439	-
Soft-NMS	0.7459	-
NMW	0.7302	0.7455
WBF	**0.7573**	**0.7583**

the performance enhancement when these models are ensembled with a Co-DETR model incorporating a distance-aware loss function and YOLO series models trained on a 9:1 split under day-night conditions.

Table 6. Comparison of baseline ensemble Co-DETR models, when combined with a Co-DETR model incorporating a distance-aware loss function and YOLO series models trained on distinct datasets.

Model	Dataset	F1
Co-DETR models	9:1 split	0.7466
+ Co-DETR-DAL	9:1 split	0.7478
+ YOLO series	9:1 split	0.7511
+ YOLO series	day-night	0.7568

Post-Processing: As depicted in Table 5, each ensemble method's performance was evaluated, highlighting the influence of post-processing strategies. Our experiments indicate that implementing post-processing techniques can enhance the effectiveness of our solution. For the Non-Maximum Weighted (NMW) [16] method, applying post-processing techniques increased the F1 score from *0.7302* to *0.7455*, representing a *1.53%* improvement. In the case of the WBF method, post-processing marks a marginal improvement of approximately *0.1%*. Even though there was only a slight increase in the validation dataset, we observed an improvement upon applying WBF with post-processing on the test set.

Comparison with other teams: Table 7 shows the performance of our solution in Track 4 of the AI City Challenge 2024. Our solution demonstrated a commendable F1 score of **0.6077**, securing the 4^{th} position amidst a competitive field of over 50 participating teams.

Table 7. In track 4 of the AI City Challenge, our proposed solution demonstrated a noteworthy performance. Among 50+ teams, our solution successfully secured the 4^{th} position.

Rank	Team	F1
1	VNPT AI	0.6406
2	NetsPresso	0.6196
3	SKKU-AutoLab	0.6194
4	**UIT_AICLUB**	**0.6077**
5	SKKU-NDSU	0.5965

5. Limitations

Our system has encountered specific challenges in the realm of object detection. Distortions originating from particular cameras have been identified as a significant hurdle. These distortions often warp the objects in the images, complicating our system's ability to identify and locate them accurately. A particular issue arises with images containing numerous tiny objects near the image's edge or objects with most of the parted overlapped by more significant objects. Our current approach tends to need to be revised to detect these objects, highlighting a critical limitation we strive to address. Focusing on system limitations is crucial in our ongoing efforts to enhance object detection capabilities. These limitations are shown in Figure 7.

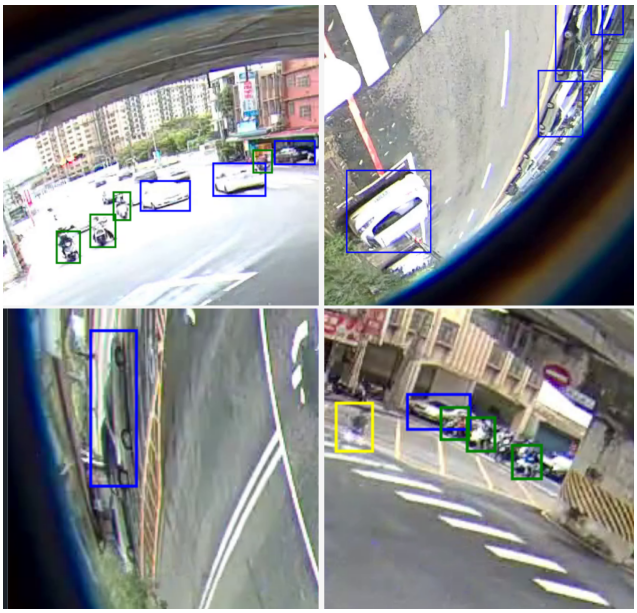

Figure 7. Illustrations of the limitations of our approach. In the first row, our system struggles to detect certain tiny objects. In the second row, we encounter difficulties in identifying objects that are partially overlapped.

6. Conclusion

In this study, we have introduced an effective framework for road object detection, explicitly tailored for fisheye cameras within the framework of Intelligent Transportation Systems (ITS). This method employs an ensemble of deep-learning models, each of which is intricately designed to overcome a specific challenge. Our approach has demonstrated robust performance, achieving a fourth position in the AI City Challenge 2024, as evidenced by an impressive **0.6077** F1 score. This work significantly advances the field of ITS and lays a solid foundation for future research in the domain of wide-angle photographic analysis and object detection.

Acknowledgment

This research is funded by University of Information Technology-Vietnam National University HoChiMinh City under grant number D1-2024-05

References

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

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. 6

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection, 2017. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 1

[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, 2023. 2

[6] Olfa Haggui, Hamza Bayd, Baptiste Magnier, and Arezki Aberkane. Human detection in moving fisheye camera using an improved yolov3 framework, 2021. 1

[7] Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3.0: A full-scale reloading, 2023. 3

[8] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015. 1, 3

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 1, 3

[10] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107: 104117, 2021. 2, 3

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 3

[12] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks, 2021. 1, 3

[13] 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. 1

[14] Senthil Yogamani, Ciaran Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uricar, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, Hazem Rashed, Sumanth Chennupati, Sanjaya Nayak, Saquib Mansoor, Xavier Perroton, and Patrick Perez. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving, 2021. 2

[15] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sünderhauf. Varifocalnet: An iou-aware dense object detector, 2021. 3

[16] Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. Cad: Scale invariant framework for real-time object detection. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 760–768, 2017. 8

[17] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training, 2023. 1, 3