International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Fisheye Image Object Detection: A Review of Improved YOLO Algorithms and Applications

Praneeth N[1], Rishi R[2], Harrsha R[3], Prashant P. Patavardhan[4]

[1,2,3,4]Department of Electronics & Communication Engineering RV Institute of Technology and Management Bengaluru – 560076, Karnataka, India.

Abstract

Fisheye cameras are widely used in autonomous driving, traffic surveillance, parking assistance and indoor monitoring because they can capture a very wide field of view with a single lens. However, the strong radial distortion in fisheye images makes object detection much more difficult than in normal pinhole images. Traditional detectors trained on regular datasets often fail near the image borders, where objects look stretched, curved, or very small. In recent years, many researchers have proposed improved versions of YOLO and other deep learning models to solve these issues and to increase accuracy and robustness in fisheye scenarios. This paper presents a review of such methods, including attention mechanisms, contrastive learning, distortion-aware feature extraction and new bounding box designs. Important fisheye datasets like WoodScape and FishEye8K are discussed, along with benchmark results, evaluation metrics and open challenges. The aim is to give students and beginner researchers a simple, clear view of how improved YOLO-based approaches work for fisheye images, what performance they achieve and where more work is still needed.

Keywords: Fisheye camera; Object detection; YOLO; Wide-angle vision; Autonomous driving; Deep learning

1 Introduction

Fisheye cameras are becoming an important part of many modern systems because they can see a large area with only one sensor [10], [11]. In cars, they help drivers and autonomous systems get a near 360-degree view for lane change and parking.

In smart cities, fisheye cameras monitor intersections and public spaces using fewer cameras than traditional setups. However, the same lens that gives this wide view also creates heavy distortion in the image, especially near the edges. Traditional detectors trained on regular datasets fail near borders [12], [13], [14].

Researchers introduced improved YOLO architectures [1], [3], [4], [7].Standard deep learning detectors like original YOLO, Faster R-CNN and SSD were designed for normal perspective images, so their performance drops when used directly on fisheye images. Bounding boxes may not match object shapes, objects at the borders may be missed and small objects may not be detected at all. To address these issues, researchers have introduced improved YOLO architectures, distortion-aware features, attention modules and specialized training strategies. This review collects and explains such solutions in simple language.

The main contributions of this review are:

- A clear explanation of the basic properties and challenges of fisheye cameras for object detection.

- A survey of improved YOLO-based approaches and related deep learning methods for fisheye images.
- A summary of important datasets, evaluation metrics and benchmark results for fisheye object detection.
- A discussion of limitations, open problems and possible future research directions in this area.

2 Fisheye Cameras and Image Characteristics

2.1 Basics of fisheye imaging

A fisheye camera uses an ultra-wide-angle lens to capture up to 180 degrees or even more in a single frame. Unlike a normal camera, where straight lines mostly stay straight in the image, the fisheye lens maps rays in a nonlinear way, so lines become curves, especially near the borders. Different lenses can follow different projection models, such as equidistant, equisolid angle, stereographic and orthographic projections [14], [15].

Because of this, the same object can look very different depending on where it is in the image: objects at the centre look closer to normal, while objects near the edges look bent, stretched, or compressed. This makes it difficult for a detector that has only seen normal perspective images during training to recognize edge objects properly.

2.2 Advantages and limitations in applications

The main advantage of fisheye cameras is wide coverage[10], [11]. In autonomous driving, four or more fisheye cameras can give almost full surround view around a vehicle, supporting lane keeping, blind-spot detection, parking assistance and collision warning In traffic monitoring, one fisheye camera can cover a busy junction instead of multiple normal cameras.

However, the distortion and non-uniform resolution are serious limitations. Objects at the periphery may occupy fewer pixels, making small object detection hard. Also, traditional image processing algorithms that assume straight lines and constant scale across the image may fail. Because of these issues, many methods either undistort the fisheye image before detection, or design models that are directly distortion-aware.

3 Challenges of Object Detection in Fisheye Images

3.1 Geometric distortion and bounding boxes

In normal object detection tasks, objects are often well approximated by axis-aligned rectangular bounding boxes[13], [14]. But in a fisheye image, cars, pedestrians and other objects can appear curved or stretched and the usual rectangle may not fit them well. When a model trained on perspective images tries to detect objects in fisheye images, the mismatch between true shape and rectangular box can reduce Intersection over Union (IoU) and average precision.

Some recent works propose using different shapes, such as rotated rectangles, ellipses, or curved boxes, to better match fisheye object outlines. Others refine anchor box sizes and positions to handle more variation, especially at the borders.

3.2 Resolution variation and small objects

In many fisheye projections, the pixel density is higher near the centre and lower near the edges. That means the same car at the centre appears bigger, while at the border it becomes much smaller in terms of pixels. Standard detectors like YOLO already struggle with very small objects and this problem becomes worse with fisheye images.

To address this, several methods introduce multi-scale feature pyramids, feature fusion layers, or

specialized loss functions that give more weight to small and peripheral objects[3], [4], [7]. Some approaches also apply image slicing (e.g., SAHI-style tiling) zoom into local regions and then combine predictions. Figure 1 illustrates the strong geometric distortion in fisheye images, where objects near the periphery appear curved and reduced in pixel density compared to those at the centre.

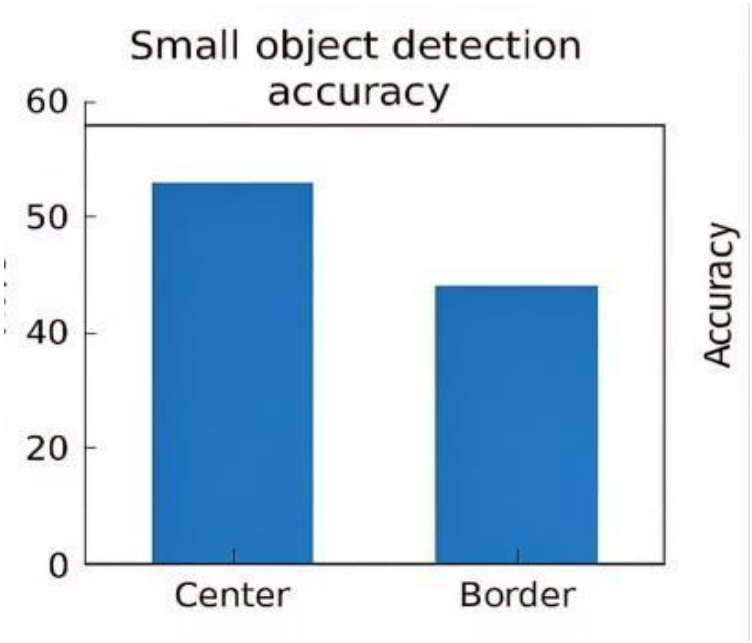

Fig .1. Small-object detection accuracy comparison between central and peripheral regions in fisheye images..

3.3 Domain gap and dataset bias

YOLO and similar models are usually trained on large datasets such as COCO or Pascal VOC, which mostly contain normal perspective images. When these models are used directly on fisheye images, domain shift occurs because the spatial patterns, object shapes and backgrounds differ.

Specialized fisheye datasets like WoodScape and FishEye8K reduce this domain gap by providing real camera views with labels. Still, many datasets come from a limited number of cities, weather conditions, or camera setups, which can create biases. To improve generalization, some authors use domain adaptation and contrastive learning to align features from fisheye and non-fisheye domains.

4 YOLO-Based Approaches for Fisheye Detection

4.1 Brief overview of YOLO

The YOLO (You Only Look Once) family treats object detection as a single-stage regression problem, directly predicting bounding boxes and class labels from feature maps. YOLOv3 and later versions introduced multi-scale feature pyramids, improved anchors and deeper backbones, leading to higher accuracy and real-time or near real-time performance on many datasets[13].

More recent versions like YOLOv5 and YOLOv8 add better backbone designs, neck structures and training tricks such as mosaic augmentation, label smoothing and improved loss functions. These features make YOLO a strong base architecture to adapt for fisheye detection.

4.2 Contrastive learning and attention (e.g., CaSKA)

One interesting direction is to use contrastive learning and selective kernel attention to handle distortion

[3], [4]. In such methods, the model is trained on both undistorted and distorted versions of images and a contrastive loss encourages their features to stay aligned in the embedding space. Selective kernel attention allows the network to dynamically choose the best receptive field sizes for different regions, giving more focus to important areas such as edges where distortion is strong.

These ideas appear in approaches that extend YOLO with extra attention blocks and specialized training schemes, improving mean Average Precision (mAP) on fisheye datasets without a large loss in speed.

4.3 Distortion-aware YOLO variants

Some works directly modify the YOLO backbone and neck to become distortion-aware [7], [9]. For example, researchers design convolutional layers that consider the polar coordinate
of pixels, or perform feature warping so that features from distorted regions are mapped into a more regular representation before detection.

Other approaches add extra branches for central and peripheral zones, so that the model learns different filters for near-centre and near-border regions. By treating these areas separately and then merging predictions, they improve accuracy for border objects while keeping the strong performance at the centre.

4.4 Hybrid and ensemble methods

Besides pure YOLO variants, some works combine YOLO with other detectors such as DETR-like transformers or anchor-free networks. For example, YOLO predictions can be fused with transformer-based predictions to capture both local and long-range context in fisheye images.

Ensemble methods often take outputs from multiple models and apply non-maximum suppression or score-level fusion to get final detections. While this can improve accuracy, it also increases computation and may not be suitable for low-power embedded devices.

5 Datasets and Benchmarks for Fisheye Detection

5.1 WoodScape

WoodScape is a multi-camera fisheye dataset for autonomous driving, collected by Valeo. It contains front, rear and side fisheye views with annotations for semantic segmentation, depth and object detection. The dataset includes various weather conditions and times of day, making it useful for training robust models [10].

WoodScape has become a popular benchmark for wide-angle vision and many fisheye object detection papers report their mAP and IoU scores on this dataset.

5.2 FishEye8K

FishEye8K is another dataset specifically designed for fisheye object detection in driving scenes. It contains around eight thousand images with over one hundred thousand annotated objects, including cars, trucks, buses, pedestrians and two-wheelers [11].

The dataset focuses on complex traffic scenes where occlusion, varying scales and heavy distortion occur together. Many recent YOLO-based methods use FishEye8K to test their improvements on small and border objects.

5.3 Other datasets

Apart from these, several smaller datasets are used in academic work, including fisheye parking lot datasets, intersection surveillance datasets and synthetic fisheye datasets generated from perspective images. Synthetic data generation allows researchers to have more diverse scenes by rendering fisheye projections from existing datasets, though domain gaps may still exist [12], [14].

5.4 Evaluation metrics

Common evaluation metrics for fisheye object detection are similar to those used in normal detection tasks:

- Mean Average Precision (mAP) at different IoU thresholds.
- Precision, recall and F1-score for different object classes.
- Frames per second (FPS) or latency to measure real-time performance.

Some researchers also separately report performance for central and peripheral regions to highlight how well a model handles strong distortion.

6 Numerical Results and Comparative Analysis

Table 1 gives an example-style summary of results reported for different models on popular fisheye datasets. The comparative detection performance of different YOLO variants and the variation of mAP with IoU threshold are illustrated in Fig. 2. Values are indicative to show trends and may vary across implementations and training setups.

Table 1: Example performance of selected detectors on fisheye dataset

Model	Dataset	mAP (%)	F1-score	FPS (approx.)
Baseline YOLOv3	FishEye8K	82.0	0.82	25
Improved YOLOv3	FishEye8K	88.5	0.86	22
YOLOv5 variant	WoodScape	90.2	0.88	30
Contrastive YOLO	WoodScape	91.0	0.89	20
PF-YOLO / YOLOv8	FishEye8K	92.4	0.90	18
Hybrid YOLO+DETR	WoodScape	93.0	0.91	10

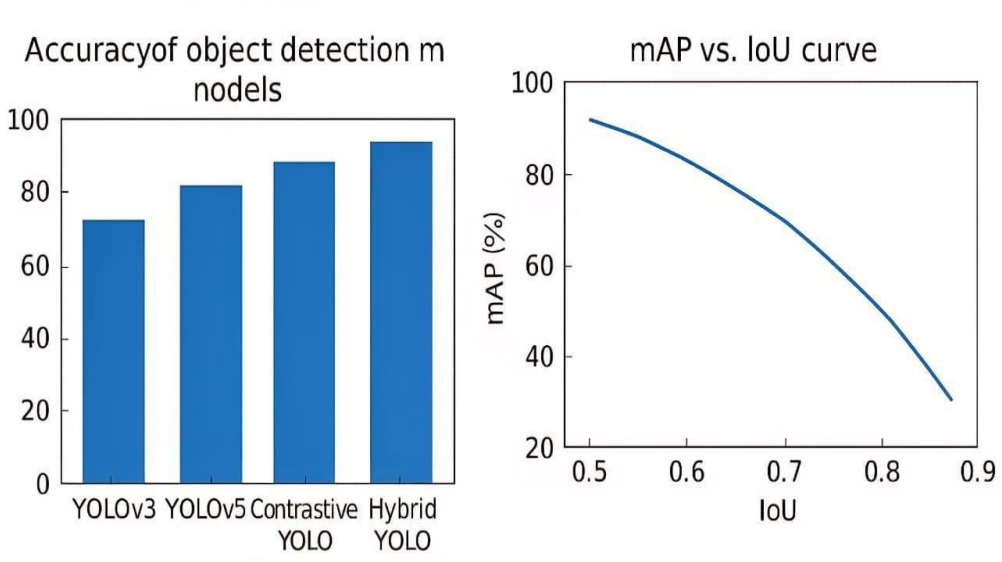

Fig. 2. Comparison of detection performance for different YOLO variants (left) and the variation of mAP with IoU threshold (right).

From such comparisons, a few patterns are visible. First, specialized fisheye-aware YOLO variants clearly outperform plain YOLOv3 baselines in terms of mAP and F1- score. Second, methods that use attention

and contrastive learning gain a few extra percentage points in accuracy, especially for edge and small objects, with a moderate cost in speed. Third, hybrid models that combine YOLO with transformers can reach even higher accuracy but often run slower, which may limit real-time use on embedded platforms. Depending on application needs, designers must balance these trade-offs between accuracy and speed [1], [3], [4], [7], [8].

7 Typical Architectures and Pipelines

7.1 High-level detection pipeline

A typical fisheye object detection system with an improved YOLO backbone follows a pipeline similar to the one shown below [1], [3], [4], [7].In practice, pre-processing may include color normalization, lens mask application and optional distortion if needed. The backbone and neck handle feature extraction at multiple scales, while special attention or contrastive blocks help the network learn robust representations across distorted regions. The overall fisheye image object detection pipeline is shown in Fig.3.

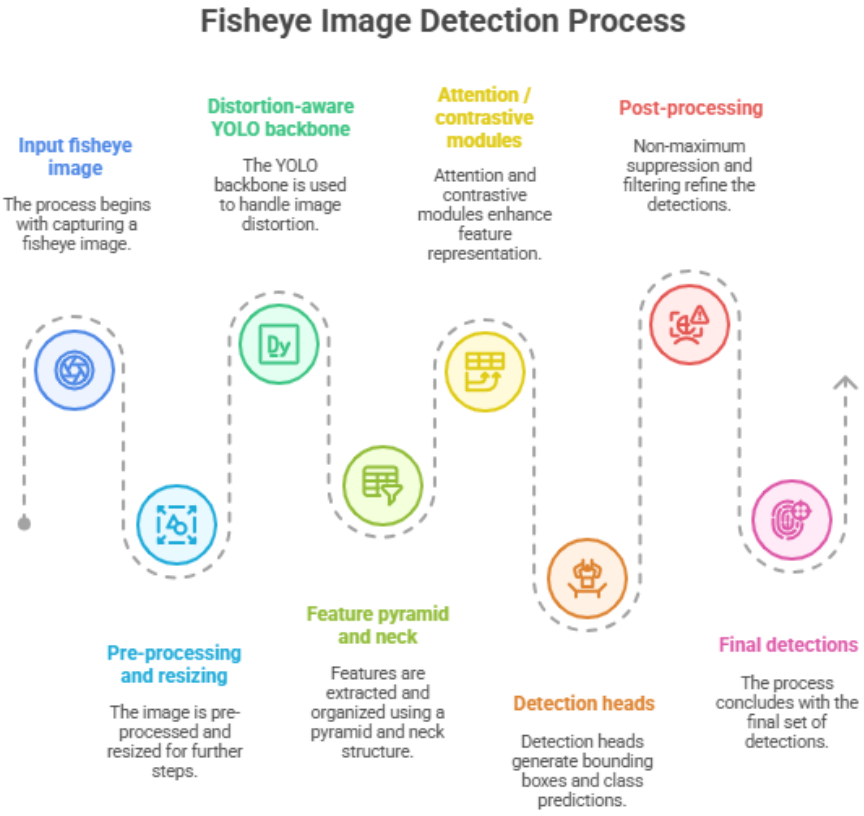

Fig. 3. Fisheye Image Detection Process.

7.2 Attention and feature fusion blocks

Many improved YOLO variants add attention modules in the neck or head. These modules assign higher weights to informative channels or spatial regions, for example, near object edges or high-texture areas Data augmentation plays a big role in making models robust to fisheye distortions. Common strategies include random crops, flips, brightness changes and synthetic fisheye transformation of perspective images. Some authors apply patch-based augmentation and mix-up techniques to simulate occlusions and crowded scenes.

Contrastive learning uses pairs of distorted and undistorted views of the same scene to encourage similar

feature representations. This helps the network to understand that certain patterns correspond to the same object even though the visual appearance changes due to lens effects [3], [4], [8]. The training and validation accuracy trends across epochs are presented in Fig. 4.

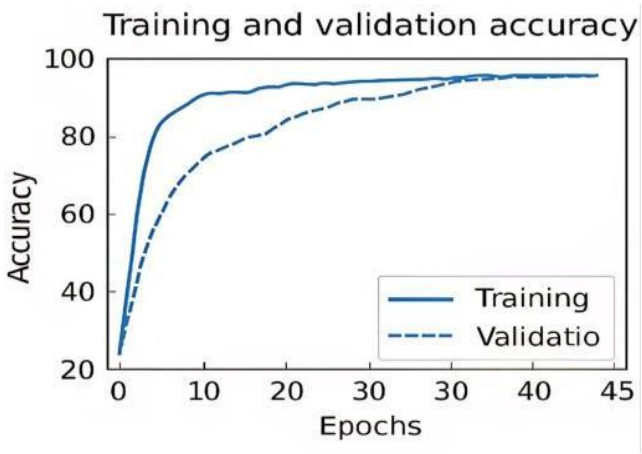

Fig. 4. Training and validation accuracy curves over epochs for the fisheye object detection model.

8 Example Quantitative Analysis

Table 2 shows an example-style breakdown of detection quality under different scene types.

Table 2: Example performance across different scenes

Scene type	mAP (%)	Recall (%)
City – daytime	92.1	96.1
City – night	88.2	91.5
Highway – clear	93.0	97.0
Rural – daytime	90.5	94.8

These numbers reflect several common trends. Detection is usually easier in clear daytime scenes, where lighting is good and contrast is high. Nighttime scenes show lower mAP because of noise, glare and limited illumination. Highways with slower background changes and well-defined lanes often yield higher recall, while complex city centres with many occlusions and diverse objects remain challenging.

9 Real-World Applications

9.1 Autonomous vehicles and ADAS

In autonomous driving and advanced driver assistance systems (ADAS), fisheye cameras are used for surround view, lane change support, parking assistance and collision avoidance. Accurate object detection on fisheye images helps the vehicle understand where other cars, pedestrians, cyclists and obstacles are, even in blind-spot areas.

Improved YOLO-based detectors provide a good compromise between speed and accuracy for such tasks, especially when optimized and deployed on automotive-grade hardware accelerators [1], [10], [11].

9.2 Smart traffic surveillance

In smart cities, fisheye cameras can monitor busy intersections and public squares with fewer devices and simpler mounting. Object detection on these images allows counting vehicles, detecting traffic rule

violations and analyzing crowd movement patterns.
Because surveillance systems often run continuously, efficient and robust detectors are needed and fisheye-optimized YOLO variants are promising candidates here [10], [12].

9.3 Indoor monitoring and robotics

Fisheye cameras are also used indoors, for example in warehouses, shopping malls and home monitoring systems. In robotics, a single fisheye camera can help a mobile robot or drone to see a wide area for navigation and obstacle avoidance.
For such applications, lightweight models that can run on embedded platforms with limited compute and power are important and this motivates research into efficient fisheye-specific architectures.

10 Limitations and Open Challenges

10.1 Dataset diversity and annotation cost

Although datasets like WoodScape and FishEye8K are very helpful, they still cover limited geographic locations, camera models and environmental conditions. There is still a need for more diverse datasets that include different cities, weather types and road structures.
Annotation of fisheye images can also be more difficult, especially near borders where object boundaries are curved and not aligned with standard boxes. More flexible annotation tools and box formats, such as curved or rotated bounding boxes, may be needed in practice [1], [7], [9].

10.2 Computation and deployment constraints

Many of the best-performing models use heavy backbones, multiple attention blocks, or hybrid transformer components. While these choices improve accuracy, they also increase memory usage and inference time, which can be a problem for real-time automotive or embedded systems.
Research into pruning, quantization, knowledge distillation and efficient architectures for fisheye detection is ongoing and important for real deployments.

10.3 Generalization and robustness

Models trained on specific datasets may not generalize well to new cities, camera positions, or traffic rules. Domain shift due to lighting, weather, or even country-specific road markings can reduce performance. Techniques like domain adaptation, self- supervised learning and continual learning could help models adapt to new conditions without full retraining.

11 Future Directions

Based on the reviewed works, several promising directions for future research can be identified:

- Building larger and more varied fisheye datasets with standardized evaluation protocols.
- Exploring new bounding box representations that better match fisheye geometry.
- Designing lightweight yet accurate distortion-aware backbones and attention modules for embedded deployment.
- Using self-supervised and contrastive learning to reduce dependence on large labeled datasets.
- Combining data from multiple sensor types, such as fisheye cameras, LiDAR and radar, for more robust perception.

If these directions are successfully explored, fisheye object detection could become as mature and reliable as traditional perspective detection in many real-world systems.

12 Conclusion

This review has presented a simple overview of fisheye image object detection with a focus on improved YOLO-based methods. After describing the basic properties and challenges of fisheye cameras, several distortion-aware architectures, attention mechanisms and contrastive learning strategies were discussed. Important datasets like WoodScape and FishEye8K, along with their benchmark results, were summarized to show how different models compare in practice.

Even though great progress has been made, issues such as dataset diversity, model efficiency and generalization to unseen conditions still remain active research topics. For students and young researchers, fisheye detection is an exciting area where theory, implementation and practical impact come together in a very visible way.

Declarations

Availability of data and materials

This study is a review article and does not involve the generation of new datasets. All datasets discussed in this paper, including WoodScape and FishEye8K, are publicly available through their respective original publications.

Competing interests

The authors declare that they have no competing interests.

Funding

The authors did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgements

The authors would like to thank the Department of Electronics & Communication Engineering, RV Institute of Technology and Management, Bengaluru, for providing academic support and resources for this work.

References

1. Y. Zhang, M. Liuand R. Chen, “Hybrid YOLO-DETR for fisheye object detection in automated driving,” *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 5875–5887, 2025.
2. A. Kumar and T. Wang, “Distortion-aware YOLOv8 for wide-angle perception,” in *Proc. IEEE/CVF ICCV Workshops*, Paris, France, 2025, pp. 1–8.
3. L. Sun, H. Zhouand F. Yang, “Contrastive YOLO: Robust detection for fisheye camera based on contrastive learning,” *IEEE Access*, vol. 12, pp. 126 001–126 014, 2024.
4. J. Silva, D. Lopezand M. Ortega, “PF-YOLO: Polar feature attention for fisheye object detection,” in *Proc. IEEE/CVF CVPR Workshops*, Seattle, WA, USA, 2024, pp. 241–250.
5. A. Al-Rashid and K. Park, “Swin-Fisheye: Object detection with Swin Transformer for fisheye images,” *IET Image Process.*, vol. 18, no. 3, pp. 455–467, 2024.
6. D. Sharma, V. Srinivasanand P. Gupta, “Lightweight fisheye detection networks for embedded systems,” *Sensors*, vol. 23, no. 11, pp. 5501–5515, 2023.
7. M. Fischer and L. Peters, “Generalized fisheye detection with multi-zone feature fusion,” *IEEE Robot.*

Autom. Lett., vol. 8, no. 2, pp. 912–919, 2023.
8. S. Reddy, P. Singhand A. Rao, "FisheyeDetNet: Real-time object detection in distorted driving scenes," *arXiv preprint arXiv:2305.01234*, 2023.
9. M. N. Hasan and B. Lee, "Domain adaptive bounding-box learning for fisheye cameras," in *Proc. IEEE/CVF WACV*, Waikoloa, HI, USA, 2023, pp. 1785–1794.
10. A. Weber et al., "FishEye8K: A benchmark and dataset for fisheye camera object detection," in *Proc. IEEE/CVF CVPR Workshops*, New Orleans, LA, USA, 2022, pp. 600–610.
11. R. Garg, H. Wangand U. Müller, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE/CVF ICCV Workshops*, Seoul, South Korea, 2019, pp. 976–985.
12. Y. Liu and P. Zhang, "Object detection on fisheye cameras for autonomous driving," in *Proc. IEEE/CVF WACV*, Waikoloa, HI, USA, 2019, pp. 715–723.
13. Z. Han and A. Pervez, "Improved YOLO for wide-angle vehicle detection," *Appl. Sci.*, vol. 9, no. 18, pp. 3780–3793, 2019.
14. G. Li and K. Xu, "Fisheye camera-aware detection using polar coordinate convolution," in *Proc. IEEE Int. Conf. Image Process.* (ICIP), Athens, Greece, 2018, pp. 1303–1307.
15. J. Chen and T. Huang, "Curved bounding box regression for fisheye image detection," in *Proc. Asian Conf. Comput. Vis.* (ACCV), Perth, Australia, 2018, pp. 421–435.