Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061690

ARTICLE

Robust Detection for Fisheye Camera Based on Contrastive Learning

Junzhe Zhang[1], **Lei Tang**[1,*] **and Xin Zhou**[2]

[1]School of Information Engineering, Chang'an University, Xi'an, 710064, China
[2]College of Automation, Northwestern Polytechnical University, Xi'an, 710129, China

*Corresponding Author: Lei Tang. Email: tanglei24@chd.edu.cn

Received: 30 November 2024; Accepted: 19 February 2025; Published: 16 April 2025

ABSTRACT: Fisheye cameras offer a significantly larger field of view compared to conventional cameras, making them valuable tools in the field of computer vision. However, their unique optical characteristics often lead to image distortions, which pose challenges for object detection tasks. To address this issue, we propose Yolo-CaSKA (Yolo with Contrastive Learning and Selective Kernel Attention), a novel training method that enhances object detection on fisheye camera images. The standard image and the corresponding distorted fisheye image pairs are used as positive samples, and the rest of the image pairs are used as negative samples, which are guided by contrastive learning to help the distorted images find the feature vectors of the corresponding normal images, to improve the detection accuracy. Additionally, we incorporate the Selective Kernel (SK) attention module to focus on regions prone to false detections, such as image edges and blind spots. Finally, the mAP_{50} on the augmented KITTI dataset is improved by 5.5% over the original Yolov8, while the mAP_{50} on the WoodScape dataset is improved by 2.6% compared to OmniDet. The results demonstrate the performance of our proposed model for object detection on fisheye images.

KEYWORDS: Fisheye; contrastive learning; Yolov8; attention

1 Introduction

With the acceleration of urbanization and the continuous advancement of transportation technology, traffic management has become increasingly complex and demanding. To enhance traffic operational efficiency, ensure safety, and reduce accident rates, effective monitoring systems are essential in modern traffic management. These systems play a crucial role in improving traffic flow, assessing road and infrastructure conditions, enforcing traffic regulations through continuous surveillance, aiding in accident investigation and liability determination, as well as providing security monitoring to prevent and detect criminal activities. Traditionally, traffic monitoring relies heavily on pinhole cameras; however, these cameras have significant limitations. They provide minimal coverage areas with numerous blind spots, resulting in poor monitoring outcomes. In response to these challenges, fisheye cameras have emerged as a viable solution, effectively addressing the shortcomings of traditional pinhole cameras. Fisheye cameras offer an omnidirectional field of view and can capture comprehensive visual information, making them particularly advantageous for applications in intelligent surveillance [1–3], unmanned aerial vehicles [4,5], virtual reality [6,7], autonomous driving [8–10], and robotics [11–13]. However, the use of fisheye cameras presents unique challenges. Due to their spherical view characteristics, objects closer to the lens appear less distorted, while those farther away exhibit significant distortion. This inherent property results in the same object appearing with

different shapes depending on its position within the image. Such geometric distortions introduce substantial difficulties in accurately detecting and identifying objects in fisheye camera images.

In the early 20th century, Wood (1908) [14] developed the first fisheye camera and coined the term "Fisheye". Subsequently, Bond introduced the first fisheye lens using a hemispherical design, marking the beginning of optical advancements in fisheye lenses. With the breakthroughs in Convolutional Neural Networks (CNNs) for computer vision tasks, increasing numbers of researchers have shifted their focus to CNN applications. However, while CNNs excel at processing flat images, their application in fisheye cameras presents challenges. Specifically, deformed objects in fisheye imagery make feature extraction more difficult, often resulting in false or missed detections. To address this issue, fisheye images are commonly converted into planar representations using various projection methods, such as perspective projection [15] and equirectangular projection [16]. Among these methods, perspective projection involves projecting the fisheye image onto a section of the sphere's surface. While this approach offers high accuracy and relatively lightweight computational demands, it requires precise algorithmic implementation to avoid distortion, particularly when selecting the appropriate angular parameters. In contrast, equirectangular projection maps spherical coordinates into a flat rectangle but fails to maintain the continuity and periodicity inherent in spherical coordinates. This results in edge distortions and non-linear image deformation, especially near the poles of the projected image due to uneven distribution of spatial resolution.

Recent studies have incorporated fisheye images directly into CNN architectures. Researchers often integrate feature enhancement modules within CNN frameworks to improve feature extraction from distorted objects [17]. Some approaches have explored the integration of multiple state-of-the-art models to facilitate hybrid inference [18], while others have utilized synthetic fisheye images to augment training datasets [19]. However, existing works have not fully exploited the potential of comparing synthetic fisheye images with their corresponding original images within a contrastive learning framework.

Contrastive learning is a training framework that learns by comparing input samples [20–23]. Its objective is to maximize the similarity between "positive pairs" (samples from the same category) and simultaneously minimize the similarity between "negative pairs" (samples from different categories) in the embedding space. For instance, two views of the same image can be treated as a positive pair, whereas two views from different images are considered negative pairs [24]. This learning strategy is based on the principle of instance discrimination, where each image is treated as an individual category and the model aims to distinguish it from all others. Since contrastive learning typically requires paired samples from the same image to be classified into the same category, data augmentation becomes essential for generating diverse yet complementary views [25]. To our knowledge, this approach has not been systematically explored or applied to object detection tasks in fisheye camera imagery.

To address the challenge posed by fisheye camera distortion, we propose an innovative solution in this study. Our approach leverages contrastive learning by utilizing standard images alongside their corresponding synthetic fisheye images, generated via a specialized algorithm. This method employs corresponding image pairs as positive samples and others as negative samples to enhance object detection accuracy for fisheye cameras through contrastive learning techniques. We adopt Yolov8, a leading-edge object detection model, as our base architecture and introduce an improved methodology to achieve superior performance in detecting objects within fisheye camera data. The primary contributions of this research are outlined below:

1. We integrate contrastive learning into fisheye camera object detection for the first time, capitalizing on the consistent features between synthetic fisheye images and their corresponding standard counterparts. This novel application significantly advances existing methods in addressing fisheye distortion.

2. To mitigate common issues such as edge and corner misdetections, we incorporate the SK attention module into our model. This module effectively boosts the model's ability to perceive local features, thereby improving detection accuracy.

3. Experiments on the enhanced KITTI dataset and the open fisheye image dataset demonstrate the effectiveness of our proposed method.

2 Literature Review

2.1 Object Detection

Object detection is a critical task in image processing and has found wide applications in domains such as traffic monitoring. Early approaches to object detection primarily relied on manual feature extraction; however, this method reached performance saturation around 2010 [26]. With the advent of deep learning, this bottleneck has been overcome, leading to rapid advancements in deep learning-based object detection algorithms. These algorithms generally fall into two categories: two-stage detectors and one-stage detectors. Two-stage detectors frame the detection process as a "coarse-to-fine" approach, while one-stage detectors treat it as a single-step process [27].

Two-stage detectors primarily rely on CNN-based object detection algorithms. R-CNN [28], as a landmark algorithm in this category, first selectively extracts features from candidate frames and then predicts and identifies objects using a linear SVM [29] classifier. However, this approach results in relatively slow detection speeds. To address this inefficiency, SPPNet [30] introduces a Spatial Pyramid Pooling (SPP) layer, enabling CNNs to produce fixed-length representations while reducing convolutional computations. This innovation makes SPPNet over 20 times faster than R-CNN without compromising detection accuracy. Despite the improved efficiency, SPPNet's requirement for feature extraction and the generation of 2000 region proposals during training significantly increases processing time. To further tackle the speed limitations of R-CNN, researchers developed Fast R-CNN [31], building upon the foundations of both R-CNN and SPPNet. While still containing some computational redundancies in later stages, Fast R-CNN achieves more than 200 times the speed of its predecessor. Subsequent advancements have led to the development of even faster algorithms, including Faster-RCNN [32], RFCN [33], and Light Head RFCN [34]. Beyond addressing computational inefficiencies for enhanced speed, Lin et al. proposed FPN [35], which was integrated into R-CNN to enable multi-scale feature fusion and improve detection accuracy.

Two-stage detectors, which follow a coarse-to-detailed process, offer strong guarantees in terms of recall and accuracy but are rarely adopted in engineering applications due to their computational complexity and slower processing speeds. In contrast, one-stage detectors have gained popularity because they can detect all objects in a single step of reasoning. These detectors are particularly well-suited for mobile devices due to their real-time capabilities and ease of deployment. This approach significantly improved detection speed, though Yolo [36] initially lagged behind SSD [37] and Faster R-CNN in terms of accuracy. Subsequently, Yolov4 [38], Yolov5 [39], and Yolov7 [40] were introduced, each successive version improving both speed and accuracy (from 5 to 160 frames per second) through innovations like dynamic label assignment and model reparameterization. Most recently, Yolov8 [41], developed by Glenn Jocher, has emerged as a new paradigm in object detection. Similar to Yolov5, Yolov8 replaces the backbone with a C2f structure that provides richer gradient information and employs an anchor-free design combined with a Decoupled-head architecture for the detection head. This approach achieves convolutional decoupling of detection and classification, further enhancing accuracy and efficiency in object detection. Furthermore, compared to Transformer-based [42] detection algorithms like DETR [43], Yolov8 demonstrates lower resource consumption and superior deployability, underscoring its strong engineering applicability.

2.2 Fisheye Camera Object Detection

Fisheye cameras offer a wider field of view than standard cameras at a lower cost, making them increasingly popular among scholars. However, these lenses require computer vision techniques to address lens distortion, which has historically been managed through de-distorting fisheye images or features, such as the SIFT algorithm [44], to approximate the appearance of an average human being from a distorted image. While this approach simplifies feature classification, it inevitably introduces approximation errors that degrade performance. Later advancements introduced CNN-based fisheye algorithms. Su et al. [45] developed SPHCONV (spherical convolution), which adjusts rectangular projections to shape filters based on spherical tangent plane projections, simulating the filter response generated across all projections of a spherical image. Although this method achieves high accuracy, it is computationally expensive. SphereNet [46] then integrated CNN into the sphere tangent plane and effectively reversed image distortion by adjusting the convolution kernel sampling grid position. However, increasing the number of network layers in this approach reduces overall accuracy. SpherePHD [47] further addressed this challenge by increasing the number of spherical facets, projecting sphere information onto an icosahedron for more uniform spatial resolution distribution. This significantly improved boundary distortion and discontinuity issues. Nonetheless, the increased computational cost associated with additional spherical facets remained a trade-off. To mitigate this, Chiang et al. [48] generated multiple perspective views from a fisheye image and applied an existing detector to analyze these synthetic images, enhancing detection accuracy through multi-angle examination of the fisheye image. However, this method proved ineffective for detecting small objects. In response, Kim et al. [49] extended the region of small objects using scalable spherical projection, thereby improving the model's detection accuracy for small objects.

However, while their accuracy is improving, CNN-based object detection algorithms still face challenges with high computational costs and inefficiency. With the continuous iterative updates of Yolo, this fast single-stage detector has become widely adopted for object detection in fisheye images. Chen et al. [50] introduced a cascaded feature pyramid network (CFPN) model to preserve the spatial information of small objects at the network's end. This model was integrated with the Yolov3 framework for detecting small objects in fisheye images of traffic streams. On the other hand, Zhou et al. [51] enhanced the speed and accuracy of object detection in fisheye images by leveraging a faster and more advanced Yolov7 architecture, which incorporates modulated deformable convolution and Swin Transformer [52] blocks.

3 Research Methodology

This section provides a detailed description of our proposed method, with the overall process illustrated in Fig. 1. First, we generated two synthetic fisheye images with varying distortion levels for each image in the original KITTI [53] dataset. The annotations for these synthetic fisheye images were produced based on their corresponding distortion parameters. Next, the original images were blended with the synthetic fisheye images to create a new dataset for model training. The design of our model is described below through its main modules and processes: (1) The fusion backbone module based on the attention mechanism is designed to extract different types of features, including multi-scale features for recognizing objects of different sizes as well as classification features reflecting the overall semantics of the image. (2) In the dual-task detection header module, the multi-scale features are efficiently aggregated into high-level representations for predicting the location and class of the object. In contrast, the classification features are encoded as embedding vectors. (3) In the loss function section, we compute the regression loss and the classification loss of the prediction box, and the comparison loss based on the embedding vectors of the fisheye image and its original image. The purpose of this design is to ensure that the feature map of the fisheye image is as similar

as possible to the original image, thus enhancing the detection of the fisheye image. Specific methods for generating artificial fisheye images and their annotations are detailed in Section 4.1.

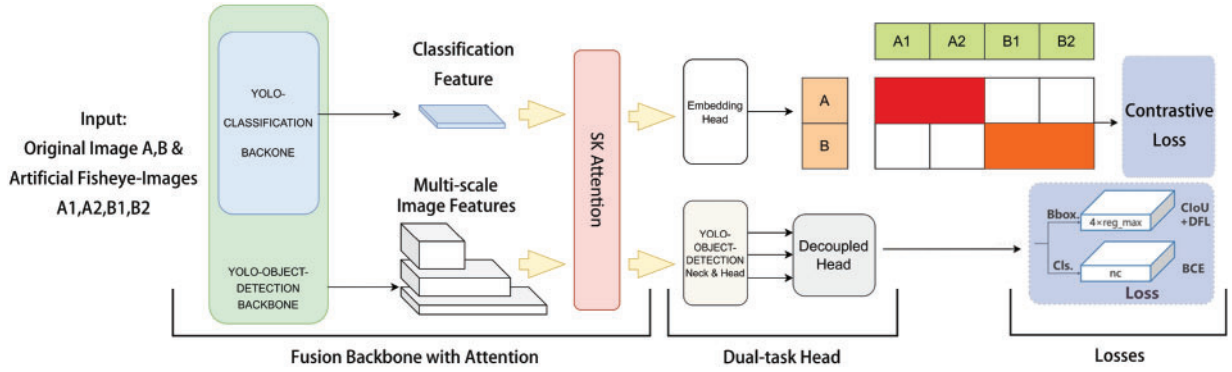

Figure 1: Framework diagram of proposed method

3.1 General Description of Methodology

This paper adopts the backbone network of Yolov8-l as its foundation. Yolov8-l is a high-performance parametric architecture derived from the Yolov8 detection model, which employs convolutional operations to extract features at three scales from input images of dimension 640 × 640 × 3. For consistent handling of non-square images, we follow the methodology established by Yolo series models. Specifically, the model adjusts the longer side of the image to match the required input dimensions while automatically resizing the shorter side proportionally. The image is then padded with gray pixels to achieve a square format. We made several key modifications to the backbone network, as illustrated in Fig. 2:

(1) The first nine blocks of Yolov8's detection network also serve as the backbone for the classification branch. To enable simultaneous feature extraction for both tasks, we incorporated an additional feature output before the SPPF (Spatial Pyramid Pooling Fast) layer. This design allows the fusion backbone to output three multi-scale feature maps for object detection and one feature map for image classification concurrently. By integrating these processes, the network achieves more efficient feature extraction.

(2) Each of the above-mentioned feature maps is processed through an SK (Selective Kernel) attention layer [54]. These layers generate new multiscale features F_1, F_2, and F_3, as well as classification-specific features F_{cls}. The SK attention mechanism automatically adjusts its kernel size based on the input characteristics, enabling the network to capture features at different scales more effectively. By introducing SK attention, we enhance the focus on regions of interest within the feature maps, which significantly improves the model's sensitivity to critical features.

These modifications collectively enhance the fusion backbone's adaptability and accuracy in extracting and fusing multiple features across different tasks.

The Dual-task Head module consists of a Detection Head and an Embedding Head, designed to effectively process the multiple feature information extracted in the previous step. The Detection Head is based on the Yolov8-l architecture, as shown in Fig. 3, and is responsible for efficiently aggregating multi-scale features F_1, F_2, and F_3 into a high-level representation for accurate prediction of object location and category. To achieve this, the detection head incorporates both a Feature Pyramid Network (FPN) and a Path Aggregation Network (PANet). The FPN enhances hierarchical feature fusion through top-down paths, enabling effective complementation of features across different scales. This design improves object detection capability at each scale, allowing the model to better handle both small and large object detection tasks.

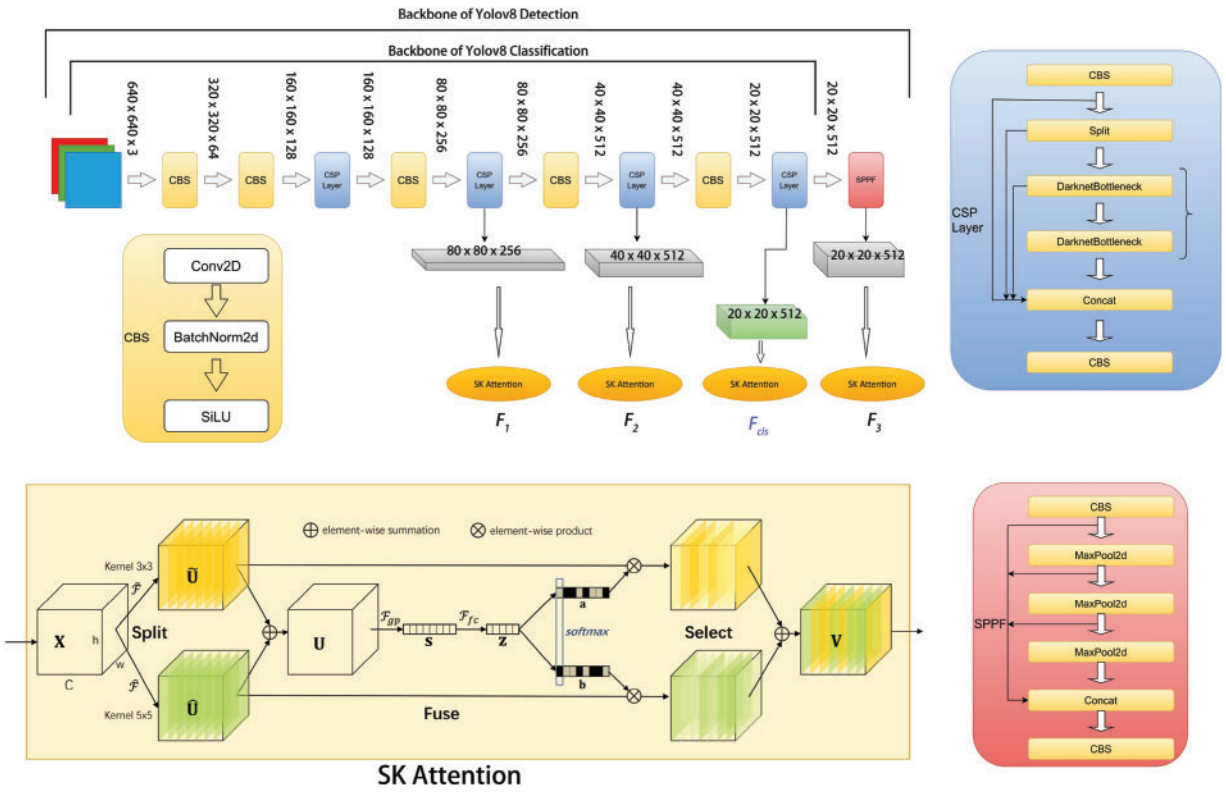

Figure 2: Integrating attention based feature fusion backbone

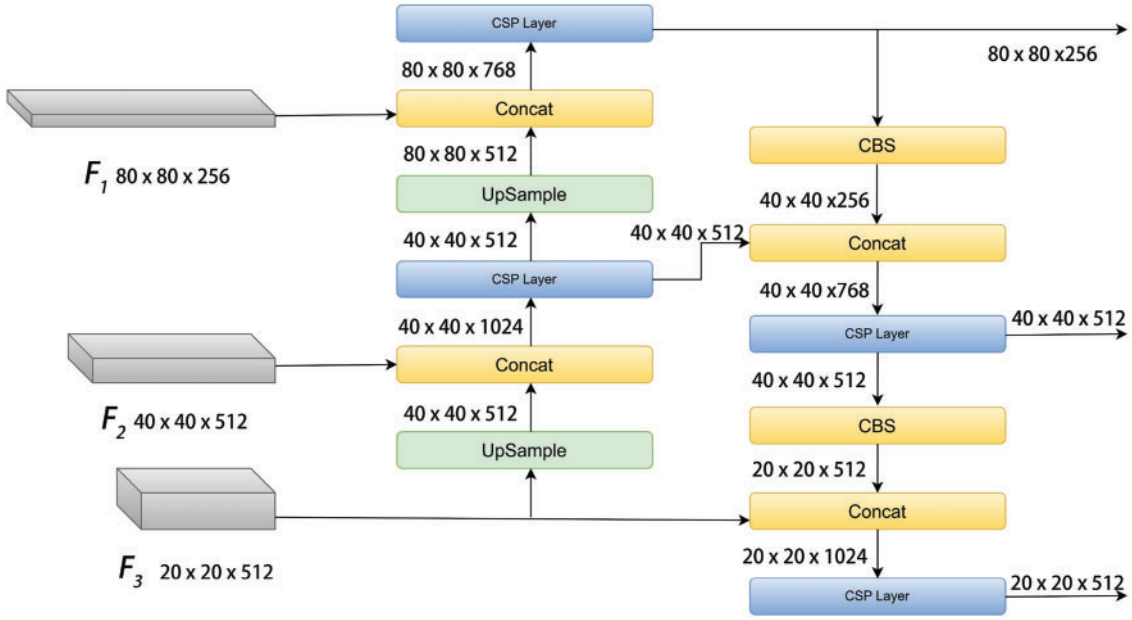

Figure 3: Detector head

Meanwhile, PANet improves feature representation and promotes efficient information reuse through a bottom-up pathway. Although this design slightly increases computational costs, it significantly enhances

multi-scale information transfer efficiency, thereby strengthening the model's detection capabilities in complex scenarios. The PANet architecture effectively integrates low-level semantic information with high-level semantic features, enabling the model to fully utilize information across different levels. Furthermore, by combining the structures of FPN and PANet, the CSP (Cross Stage Partial) module further enhances feature map integration at multiple scales. This module boosts the expressive power of the feature maps by facilitating the fusion of shallow and deep features, ensuring that the model can achieve accurate and efficient detection even when dealing with objects of varying sizes and complexity.

The classification features F_{cls} are encoded as embedding vectors using the Embedding Head, as illustrated in Fig. 4. F_{cls} undergo convolution operations. The resulting feature representations are then processed through mean pooling and a linear layer to produce a compact vector $F_e \in \mathbb{R}^{1\times1024}$, which serves as input for the subsequent contrastive loss calculation step.

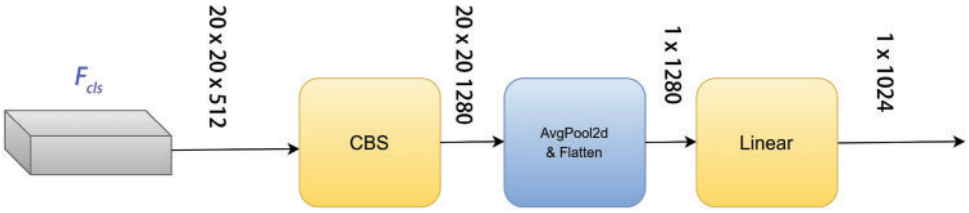

Figure 4: Embedded head

3.2 Loss Function

The losses employed during model training include CIoU (Complete Intersection over Union), BCE (Binary CrossEntropy), and contrastive loss.

As an extension of the traditional IoU (Intersection over Union) metric, CIoU is specifically designed to optimize the regression task for the object bounding box. In Yolov8, the implementation of CIoU loss incorporates not only the overlap between predicted and actual boxes but also integrates the compatibility of centroid distance and aspect ratio between bounding boxes. The CIoU loss function can be expressed as:

$$CIoU = IoU - \frac{d_c}{\max(w_h)} - \alpha \cdot v \tag{1}$$

Here d_c represents the distance between the centroid of the predicted box and the actual box, w_h is the aspect ratio of the actual box, α denotes the weight factor controlling the aspect ratio loss, and v measures the shape difference between the actual box and the predicted box. By incorporating these factors, the CIoU loss provides enhanced contextual information to the model, thereby accelerating convergence and improving detection accuracy. This loss function places a strong emphasis on the regression quality of the overall frame during training, effectively reducing uncertainty in bounding box predictions.

BCE loss is a vital function used to deal with binary classification tasks and is widely used in the object presence discrimination part of object detection. The BCE loss is mainly used to evaluate the model's ability to discriminate between objects in the object frame and the background. The basic form can be expressed as:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2}$$

Here N is the number of samples. y_i is the true label of the ith sample (0 for background, 1 for object). $\hat{y}_i$ is the predicted probability of the model for the ith sample.

In addition to the above two loss functions commonly used in object detection models, we also use Contrastive Loss to minimise the difference between the synthetic fisheye image features and the original image features. Taking Fig. 5 as an example, each batch reads 6 images, including the original images A, and B and the distorted synthetic fisheye images A_1, A_2, B_1, B_2. The output of the Embedding Head is F_e^A, F_e^B, F_e^{A1}, F_e^{A2}, F_e^{B1}, F_e^{B2}.

(a) Undistorted standard image

(b) $k_1 = -0.2$

(c) $k_1 = -0.25$

Figure 5: Schematic diagram from normal image to distorted image

After obtaining the above 6 embedding feature vectors, construct the following sample pairs:

$$Pairs=(d(F_e^A,F_e^{A1}),d(F_e^A,F_e^{A2}),d(F_e^A,F_e^{B1}),d(F_e^A,F_e^{B2}),d(F_e^B,F_e^{B1}),d(F_e^B,F_e^{B2}),d(F_e^B,F_e^{A1}),d(F_e^B,F_e^{A2})) \tag{3}$$

Here $d(,)$ is the Euclidean distance between the two vectors. Specifically, each batch constructs eight sample pairs, denoted as $P_i \in Pairs(i = 1, 2, ..., 8)$, representing the distances of these sample pairs. When P_i corresponds to a positive sample pair, it is assigned a label of 0; conversely, if P_i reflects a negative sample pair, it is assigned a label of 1. The resulting labels for the sample pairs are as follows:

$$Labels = (0, 0, 1, 1, 0, 0, 1, 1) \tag{4}$$

Then the Contrastive Loss can be expressed as:

$$Loss_{Contrastive} = 1/8 * (1 - Labels) * Pairs^2 + Labels * (\max(0, m - Pairs)^2) \tag{5}$$

Here m serves as a hyperparameter designed to constrain the minimum distance between negative sample pairs. Specifically, when P_i represents a positive sample pair, the associated label $Labels_i$ is set to 0, leading to a loss function defined as P_i^2. In this case, a larger distance between positive sample pairs results in an increased loss value, while a smaller distance yields a reduced loss value. Conversely, when P_i denotes a negative sample pair, the label $Labels_i$ is assigned a value of 1, and the corresponding loss function is expressed as $\max(0, m - P_i)^2$. Here, if the distance between negative sample pairs exceeds m, the loss value becomes 0; conversely, if the distance is less than m, the loss increases as the distance decreases. Overall, the goal of the contrastive loss is to minimize the distance between positive sample pairs while ensuring that the distance between negative sample pairs exceeds m.

4 Experiment

4.1 Data Preparation

Currently, publicly available fisheye image datasets remain limited to WoodScape [55] and Fisheye8K [56]. To address the challenge of insufficient data volume in these datasets for effective model training, researchers have turned to synthetic fisheye images as a common solution. Notably, Broks et al. [57] have conducted extensive research on methods for augmenting synthetic fisheye images for convolutional neural network (CNN) object detection, incorporating polynomial fisheye models into their approaches. In their study, Duong et al. [19] leveraged the iFish tool [58], which is also based on a polynomial fisheye model, to develop the Synthetic VisDrone dataset, achieving promising training outcomes. Following this approach, our study utilizes a polynomial fisheye model to augment the KITTI dataset, thereby improving its suitability for model training purposes.

The KITTI dataset was co-founded by Karlsruhe Institute of Technology (KIT), Germany, and Toyota Technological Institute at Chicago (TTI-C), USA, in 2012, and is one of the most commonly used computer vision algorithm evaluation datasets for autonomous driving scenarios internationally. Kitti contains 7481 perspective camera images and their annotations, including urban, rural, and highway scenes, with up to 15 vehicles and 30 pedestrians in each image. To train and validate the performance of the proposed model, we expanded the KITTI dataset using synthetic fisheye images, and the expanded dataset has 22,443 images.

It is important to note that the KITTI dataset includes the following annotated object categories: “Pedestrian”, “Car”, “Truck”, “Cyclist”, “Van”, “Tram”, “Person set”, and “Don’t care”. However, subsequent publicly available autonomous driving scene datasets, such as the vehicle section of COCO [59], do not include the categories “Tram”, “Person set”, and “Don’t care”. Furthermore, in COCO, the category “Van” has been absorbed into the broader “Car” label rather than retained as a separate class. Similarly, within the fisheye image dataset WoodScape [55], all vehicles are grouped under a single label without further subdivisions. To align our dataset with these more recent conventions, we have removed the annotations for

"Tram", "Person set", and "Don't care" from KITTI and replaced the "Van" category with the merged "Car" label. Consequently, the final dataset includes the categories: "Pedestrian", "Car", "Truck", and "Cyclist".

4.2 Synthetic Fish-Eye Image Augmentation

In this paper, polynomial fisheye model is used to generate synthetic fisheye images.

$$x_u = x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \tag{6}$$

$$y_u = y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \tag{7}$$

In the above equation, the coordinates x_d, y_d, x_u, y_u represent normalized pixel coordinates within the image centroid coordinate system, where $x_d, y_d, x_u, y_u \in [-1, 1]$. Here, x_d and y_d correspond to the original pixel coordinates, while x_u and y_u denote the distorted pixel coordinates. The parameter $k_i < 0$ signifies the distortion coefficient, and r represents the distance from the original pixel to the image center. As the absolute value of k_i increases, the image exhibits more pronounced fisheye distortion, as illustrated in Fig. 6.

Figure 6: Schematic diagram of distorted image object detection box

In this study, we set k_2 and k_3 to 0 and varied k_1 to different values in order to generate synthetic fisheye images. Specifically, Fig. 5a presents the undistorted standard image, while Fig. 5b corresponds to an image with $k_1 = -0.2$, and Fig. 5c corresponds to an image with $k_1 = -0.25$. Consequently, in our augmented KITTI dataset, which consists of 22,443 images, one-third are undistorted images, one-third have a distortion coefficient of $k_1 = -0.2$, and the remaining third have a distortion coefficient of $k_1 = -0.25$.

4.3 Annotations for Synthetic Fish-Eye Image

We use the same formula to generate the labelled box for the synthetic fisheye image. All pixel points are taken from the original bounding box and mapped onto the new image using fisheye enhancement. As shown in Fig. 6, the original bounding box (green point) is now curved. The maximum and minimum values of the pixel coordinates of the green point are saved. When the minimum and maximum x and y values are found, two points will be created, one with pixel coordinates (x_{min}, y_{min}) and the other with pixel coordinates (x_{max}, y_{max}), to produce the labelled box of the artificial fisheye image (red box).

4.4 Implementation Details

Our network was built on PyTorch 1.18.0, using Adam as the optimizer to train the model. The initial learning rate was set to 0.01, the weight decay was 1×10^{-4}, and the batch was set to 6. The entire training process took roughly 50 h. All experiments were implemented on a server with 16 GB RAM and NVIDIA 4090 GPUs.

4.5 Comparison with General Methods

We experimentally compare with other general algorithms for object detection on the augmented KITTI dataset. Specifically, within our augmented KITTI dataset comprising 22,443 images, one-third are undistorted images, one-third correspond to a distortion coefficient of $k_1 = -0.2$, and the remaining third correspond to a distortion coefficient of $k_1 = -0.25$. This arrangement ensures that our training, validation, and test sets include images with varying degrees of distortion. We employ this training methodology to assess the robustness of our approach across images with different distortion levels, thereby highlighting its versatility in various scenarios. To illustrate the effectiveness of our method, we select some classic, high-performance general algorithms from the field of object detection for comparative analysis of robustness.

Ge proposed YoloX [60], which uses an anchorless frame design with decoupled headers and a leading label assignment strategy. Zhang et al. proposed DINO [61], which uses a hybrid query approach for anchor initialization and a forward scheme for frame prediction. Chen et al. proposed YoloF [62] by presenting two key components, namely Dilated Encoder and Uniform Matching, which bring considerable improvements. As shown in Table 1, our method is 57.7%, 16.7%, and 0.54% higher than DIno, YoloX, and YoloF, respectively, proving the validity of our method.

Table 1: Comparison of the proposed method with general methods, with the best method in **bold**

Model	mAP_{50} (%)
Dino [61]	58.6
YoloX [60]	79.2
YoloF [62]	91.9
Ours	**92.4**

4.6 Ablation Study

4.6.1 Yolov8's Performance

The experiments employed Yolov8 as the base model on the augmented KITTI dataset. The training ran for 100 generations with an initial learning rate of 0.01, achieving an mAP_{50} score of 86.9%. While the detection accuracy is high, there remains potential for improvement.

4.6.2 Performance of Contrastive Learning

Adding contrastive learning enabled the model to better understand the similarity between fisheye image features and their corresponding standard image features. After training, when a fisheye image was inputted, the model could more effectively identify the corresponding standard image features, thereby enhancing detection accuracy. As shown in the table, incorporating contrastive learning improved model accuracy by 3.7%. These results demonstrate that our proposed contrastive learning method successfully boosts object detection performance.

4.6.3 Performance of SK Attention

Fisheye images often contain numerous edges and blind spots, which can challenge detection accuracy. To address this, we introduced the SK attention block to improve performance in these regions. As shown in Table 2, the integration of SK attention effectively enhanced the model's accuracy. The experiment confirms its effectiveness in overcoming these challenges.

Table 2: Ablation study, with the best method in **bold**

Model	mAP_{50} (%)
Yolov8	86.9
Yolov8 + Contractive Learning	89.1
Yolov8 + SK-Attention	90.1
Ours	**92.4**

4.6.4 Performance of Combining

Our approach combines both contrastive learning and SK attention mechanisms. First, we utilized the SK attention block to enhance the quality of feature maps generated by the backbone network. Subsequently, contrastive learning was employed to enable the model to better understand the similarities between fisheye image features and their corresponding standard image features. As illustrated in Table 2, applying these two techniques concurrently yielded a positive synergistic effect, significantly improving overall detection performance.

4.7 Comparison with Methods Designed for Fisheye Image

To further demonstrate the effectiveness of our method on real fisheye images, we conducted additional testing of our model on the WoodScape dataset [55]. This dataset comprises 8234 labeled fisheye images and annotates five classes: pedestrians, vehicles, bicycles, traffic lights, and traffic signs.

In this subsection, we initialized our model with the weights obtained from training on the augmented KITTI dataset and subsequently fine-tuned the model on WoodScape. Specifically, we randomly selected a subset of 6500 labeled fisheye images from the dataset to serve as the training data for fine-tuning, while the remaining 1734 fisheye images were set aside for testing. The test results are presented in Table 3.

Table 3: Comparison of the proposed method with methods designed for fisheye image, with the best method in bold

Model	mAP_{50} (%)
OmniDet	69.3
TFEM	68.5
Ours	**71.9**

In comparison to the OmniDet [10] method proposed by the WoodScape team and the novel fisheye image object detection approach (Transformer-based Feature Enhancement Module, TFEM) introduced by Cao et al. [17], our model demonstrates a noticeable improvement in performance. These findings underscore the capability of our method to effectively adapt to and enhance object detection tasks within fisheye image contexts.

5 Conclusion

To enhance object detection performance on fisheye images, we present a novel training framework based on contrastive learning. Our approach utilizes Yolov8 as the foundation and incorporates the SK Attention module to address detection challenges in blind areas and edge regions. Compared to existing object detection networks, our model demonstrates improved robustness and achieves superior detection results for fisheye images. The proposed method holds significant potential for real-world applications in

computer vision domains such as autonomous driving and robotics, where fisheye cameras are commonly employed. In future research, we aim to further refine the model by exploring additional real-world scenarios to enhance its real-time performance and applicability.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Lei Tang and Xin Zhou; formula derivation, data collection, analysis and interpretation of results, draft manuscript preparation: Junzhe Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics Approval: This study did not involve any human or animal subjects, and therefore, ethical approval was not required.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Yang L, Hu G, Song Y, Li G, Xie L. Intelligent video analysis: a pedestrian trajectory extraction method for the whole indoor space without blind areas. Comput Vis Image Underst. 2020;196(3):102968. doi:10.1016/j.cviu.2020.102968.
2. Jakab D, Deegan BM, Sharma S, Grua EM, Horgan J, Ward E, et al. Surround-view fisheye optics in computer vision and simulation: survey and challenges. IEEE Trans Intell Transp Syst. 2024;25(9):10542–63.
3. Luo X, Cui Z, Su F. Fe-Det: an effective traffic object detection framework for fish-eye cameras. In: Conference on Computer Vision and Pattern Recognition; 2024; Seattle, WA, USA. p. 7091–9.
4. Luo C, Yu L, Yan J, Li Z, Ren P, Bai X, et al. Autonomous detection of damage to multiple steel surfaces from 360 panoramas using deep neural networks. Comput Aided Civ Infrastruct Eng. 2021;36(12):1585–99. doi:10.1111/mice.12686.
5. Barmpoutis P, Stathaki T, Dimitropoulos K, Grammalidis N. Early fire detection based on aerial 360-degree sensors, deep convolution neural networks and exploitation of fire dynamic textures. Remote Sens. 2020;12(19):3177. doi:10.3390/rs12193177.
6. Bertel T, Yuan M, Lindroos R, Richardt C. OmniPhotos: casual 360° VR photography. ACM Trans Graph. 2020;39(6):1–12. doi:10.1145/3414685.3417770.
7. Munt A. Cinematic virtual reality: towards an optics of eco-screenwriting. In: Screenwriting for virtual reality. Cham, Switzerland: Springer; 2024. p. 53–72.
8. Cui Z, Heng L, Yeo YC, Geiger A, Pollefeys M, Sattler T. Real-time dense mapping for self-driving vehicles using fisheye cameras. In: 2019 International Conference on Robotics and Automation (ICRA); 2019; Montreal, QC, Canada: IEEE. p. 6087–93.
9. Häne C, Heng L, Lee GH, Fraundorfer F, Furgale P, Sattler T, et al. 3D visual perception for self-driving cars using a multi-camera system: calibration, mapping, localization, and obstacle detection. Image Vis Comput. 2017;68:14–27. doi:10.1016/j.imavis.2017.07.003.
10. Kumar VR, Yogamani S, Rashed H, Sitsu G, Witt C, Leang I, et al. OmniDet: surround view cameras based multi-task visual perception network for autonomous driving. IEEE Robot Autom Lett. 2021;6(2):2830–7. doi:10.1109/LRA.2021.3062324.
11. Billings G, Johnson-Roberson M. Adaptation of a ROI based object pose estimation network to monocular fisheye images. IEEE Robot Autom Lett. 2020;5(3):4241–8. doi:10.1109/LRA.2020.2994036.

12. Gao W, Wang K, Ding W, Gao F, Qin T, Shen S. Autonomous aerial robot using dual-fisheye cameras. J Field Robot. 2020;37(4):497–514. doi:10.1002/rob.21946.
13. Roxas M, Oishi T. Variational fisheye stereo. IEEE Robot Autom Lett. 2020;5(2):1303–10. doi:10.1109/LRA.2020.2967657.
14. Wood RW. Fish-eye views, and vision under water. The London, Edinburgh, Dublin Philos Mag J Sci. 1906;12(68):159–62. doi:10.1080/14786440609463529.
15. Sun Y. Analysis for center deviation of circular target under perspective projection. Eng Comput. 2019;36(7):2403–13. doi:10.1108/EC-09-2018-0431.
16. Yang ST, Wang FE, Peng CH, Wonka P, Sun M, Chu HK. DuLa-Net: a dual-projection network for estimating room layouts from a single RGB panorama. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019; Long Beach, CA, USA. p. 3363–72.
17. Cao H, Li Y, Liu Y, Li X, Chen G, Knoll A. Lightweight fisheye object detection network with transformer-based feature enhancement for autonomous driving. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2024; Abu Dhabi, United Arab Emirates: IEEE. p. 7399–405.
18. Gia BT, Khanh TBC, Trong HH, Doan TT, Do T, Le DD, et al. Enhancing road object detection in fisheye cameras: an effective framework integrating SAHI and hybrid inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024; Seattle, WA, USA. p. 7227–35.
19. Duong VH, Nguyen DQ, Van Luong T, Vu H, Nguyen TC. Robust data augmentation and ensemble method for object detection in fisheye camera images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024; Seattle, WA, USA. p. 7017–26.
20. Wu Z, Xiong Y, Yu SX, Lin D. Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. p. 3733–42.
21. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020; Seattle, WA, USA. p. 9729–38.
22. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning; 2020; Vienna, Austria: PMLR. p. 1597–607.
23. Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A. Unsupervised learning of visual features by contrasting cluster assignments. Adv Neural Inf Process Syst. 2020;33:9912–24.
24. Tian Y, Krishnan D, Isola P. Contrastive multiview coding. In: Computer Vision–ECCV 2020: 16th European Conference; 2020 Aug 23–28; Glasgow, UK: Springer; 2020. p. 776–94.
25. Ye M, Zhang X, Yuen PC, Chang SF. Unsupervised embedding learning via invariant and spreading instance feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019; Long Beach, CA, USA. p. 6210–9.
26. Zou Z, Chen K, Shi Z, Guo Y, Ye J. Object detection in 20 years: a survey. Proc IEEE. 2023;111(3):257–76. doi:10.1109/JPROC.2023.3238524.
27. Girshick R, Donahue J, Darrell T, Malik J. Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans Pattern Anal Mach Intell. 2015;38(1):142–58. doi:10.1109/TPAMI.2015.2437384.
28. Girshick R, Donahue J, Darrell. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014; Columbus, OH, USA. p. 580–7.
29. Suthaharan S, Suthaharan S. Support vector machine. In: Machine learning models and algorithms for big data classification: thinking with examples for effective learning. Boston, MA, USA: Springer; 2016. p. 207–35.
30. He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell. 2015;37(9):1904–16. doi:10.1109/TPAMI.2015.2389824.
31. Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision; 2015; Santiago, Chile. p. 1440–8.

32. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
33. Zhao D, Ma L, Li S, Yu D. End-to-end denoising of dark burst images using recurrent fully convolutional networks. arXiv:1904.07483. 2019.
34. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J. Light-head R-CNN: in defense of two-stage object detector; 2017. arXiv:1711.07264. 2017.
35. Lin TY, Dollr P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017; Honolulu, HI, USA. p. 2117–25.
36. Redmon J. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; Las Vegas, NV, USA. p. 779–88.
37. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Computer Vision-ECCV 2016: 14th European Conference; 2016 Oct 11–14; Amsterdam, The Netherlands. p. 21–37.
38. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934. 2020.
39. Wang Z, Jin L, Wang S, Xu H. Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. Postharvest Biol Technol. 2022;185(2):111808. doi:10.1016/j.postharvbio.2021.111808.
40. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023; Vancouver, BC, Canada. p. 7464–75.
41. Hussain M. YOLOv5, YOLOv8 and YOLOv10: the go-to detectors for real-time vision. arXiv:2407.02988. 2024.
42. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European Conference on Computer Vision; 2020; Glasgow, UK: Springer. p. 213–29.
43. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable DETR: deformable transformers for end-to-end object detection. arXiv:2010.04159. 2020.
44. Cruz-Mota J, Bogdanova I, Paquier B, Bierlaire M, Thiran JP. Scale invariant feature transform on the sphere: theory and applications. Int J Comput Vis. 2012;98(2):217–41. doi:10.1007/s11263-011-0505-4.
45. Su YC, Grauman K. Learning spherical convolution for fast features from 360 imagery. In: Advances in neural information processing systems. 2017; Long Beach, CA, USA. 30 p.
46. Coors B, Condurache AP, Geiger A. Spherenet: learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018; Munich, Germany. p. 518–33.
47. Lee Y, Jeong J, Yun J, Cho W, Yoon KJ. SpherePHD: applying CNNs on a spherical polyhedron representation of 360 degree images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019; Long Beach, CA, USA. p. 9181–9.
48. Chiang SH, Wang T, Chen YF. Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches. Image Vis Comput. 2021;105(8):104069. doi:10.1016/j.imavis.2020.104069.
49. Kim S, Park SY. Expandable spherical projection and feature concatenation methods for real-time road object detection using fisheye image. Appl Sci. 2022;12(5):2403. doi:10.3390/app12052403.
50. Chen PY, Hsieh JW, Chang MC, Gochoo M, Lin FP, Chen YS. Fisheye multiple object tracking by learning distortions without dewarping. In: 2023 IEEE International Conference on Image Processing (ICIP); 2023; Kuala Lumpur, Malaysia: IEEE. p. 1855–9.
51. Zhou J, Yang D, Song T, Ye Y, Zhang X, Song Y. Improved YOLOv7 models based on modulated deformable convolution and swin transformer for object detection in fisheye images. Image Vis Comput. 2024;144(12):104966. doi:10.1016/j.imavis.2024.104966.
52. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2021; Montreal, QC, Canada. p. 10012–22.

53. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2012; Providence, RI, USA: IEEE. p. 3354–61.
54. Li X, Wang W, Hu X, Yang J. Selective kernel networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019; Long Beach, CA, USA. p. 510–9.
55. Yogamani S, Hughes C, Horgan J, Sistu G, Varley P, O'Dea D, et al. WoodScape: a multi-task, multi-camera fisheye dataset for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019; Seoul, Republic of Korea. p. 9308–18.
56. Gochoo M, Otgonbold ME, Ganbold E, Hsieh JW, Chang MC, Chen PY, et al. FishEye8K: a benchmark and dataset for fisheye camera object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023; Vancouver, BC, Canada. p. 5305–13.
57. Broks M, Teličko J, Jakovičs A. Artificial fish-eye image augmentation approach for CNN based object detection. In: 2024 IEEE 11th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE); 2024; IEEE. p. 1–6.
58. Mor G, ifish tool. 2021 [cited 2025 Feb 18]. Available from: https://github.com/Gil-Mor/iFish.
59. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference; 2014 Sep 6–12; Zurich, Switzerland. p. 740–55.
60. Ge Z. YOLOX: exceeding YOLO series in 2021. arXiv:2107.08430. 2021.
61. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. arXiv:2203.03605. 2022.
62. Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun J. You only look one-level feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021; Nashville, TN, USA. p. 13039–48.