IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 28, 2020, accepted April 10, 2020, date of publication April 14, 2020, date of current version April 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2987868

FisheyeDet: A Self-Study and Contour-Based Object Detector in Fisheye Images

TANGWEI LI[1,2], GUANJUN TONG[1], HONGYING TANG[1], BAOQING LI[1], AND BO CHEN[1,2]

[1]Science and Technology on Microsystem Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 201800, China

[2]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Guanjun Tong (tongguanjun@mail.sim.ac.cn)

This work was supported by the National Key Research and Development Program of China.

ABSTRACT Fisheye Images have attracted increasing attention from the research community due to their large field of view (LFOV). However, the geometric transformations inherent in fisheye cameras result in unknown spatial distortion and large variations in the appearance of objects. And this fact leads to poor performance of the state-of-the-art methods in conventional two-dimensional (2D) images. To address this problem, we propose a self-study and contour-based object detector in fisheye images, named FisheyeDet. The No-prior Fisheye Representation Method is proposed to guarantee that the network adaptively extracts distortion features without prior information such as prespecified lens parameters, special calibration patterns, etc. Furthermore, in order to tightly and robustly localize objects in fisheye images, the Distortion Shape Matching strategy is proposed, which invokes the irregular quadrilateral bounding boxes based on the contour of distorted objects as the core. By combining with the ''No-prior Fisheye Representation Method'' and ''Distortion Shape Matching'', our proposed detector builds an end-to-end network. Finally, due to the lack of public fisheye datasets, we are on the first attempt to create a multi-class fisheye dataset VOC-Fisheye for object detection. Our proposed detector shows favorable generalization ability and achieves 74.87% mAP (mean average precision) on the VOC-Fisheye, outperforming the existing state-of-the-art methods.

INDEX TERMS Fisheye, object detection and recognition, large field of view (LFOV), deep learning.

I. INTRODUCTION

Recently fisheye cameras, owing to their large field of view (LFOV), have attracted diverse attention from both technical experts and the public in general. Due to providing rich visual information, they cover a wide variety of applications, ranging from generating the contents of augmented reality (AR) or virtual reality (VR) [1], improving the performance of intelligent robot vision systems [2], to reducing the complexity of perception systems [3]. As opposed to object detection under pinhole cameras, where remarkable achievements [4]–[9] have been accomplished, object detection under fisheye cameras still needs serious progress. Not enough achievements of object detection under fisheye cameras are mainly caused by the fact that the geometric transformations inherent in fisheye cameras result in unknown spatial distortion and large variations in the appearance of objects.

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

Current object detection methods in fisheye images are grouped into two categories: distortion correction-based and original LFOV image-based. In the distortion correction-based methods, image warping encompassing the correction model plays a key role in processing. The original LFOV image-based methods try to design location-based convolutional kernels [10]–[13] or adopt heuristic rules [14]–[16] to extract distortion features directly. These works have obtained satisfactory results, especially in pedestrians and vehicles [17]–[20]. However, the prevalence of these methods can be attributed to the assumption that geometric transformations are fixed and known. Meanwhile, these methods just use rectangular bounding boxes (shown in Fig.1 (a), (b), and (c)) to localize objects. Therefore, two drawbacks exist in the above methods. Firstly, building a unified fisheye distortion model is impossible since the distortion comes from many impacts like intrinsic camera parameters and various lens distortion parameters. So fisheye features extracted from corrected images or manually designed location-based convolutional kernels may be seriously affected the accuracy of

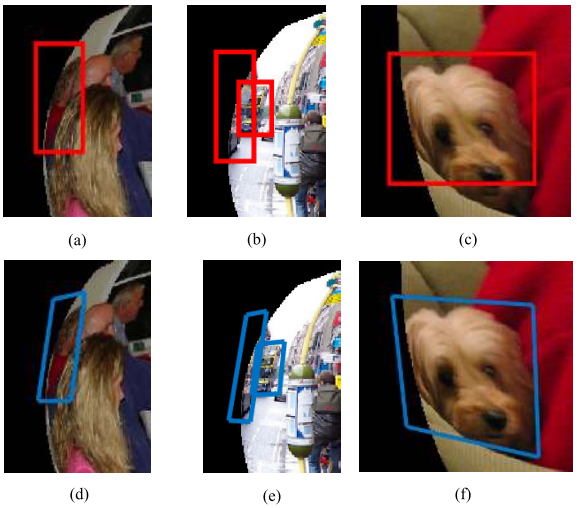


FIGURE 1. Comparison of irregular quadrilateral bounding boxes and rectangular bounding boxes for localizing objects in fisheye images. (a) (d) indicate that the rectangular bounding box brings redundant noise (black area), (b) (e) indicate that the rectangular bounding box causes unnecessary overlap, (c) (f) indicate that the rectangular bounding box cannot be exactly localized.

detection and the robustness of the model. Secondly, rectangular bounding boxes cannot provide relatively accurate locations in fisheye images. For example, as shown in Fig.1, rectangular bounding boxes [4], [21], [22] result in redundant information from background noise, unnecessary overlap, or information loss due to inaccurate annotations in fisheye images.

To address these two problems, we propose a self-study and contour-based object detector in fisheye images, called FisheyeDet, with the aim to adaptively extract valid distortion features and precisely identify predictable bounding boxes. To adaptively extract valid distortion features, we design a No-prior Fisheye Representation Method which constructs an effective fisheye feature pyramid for object detection, without prior information such as prespecified lens parameters, special calibration patterns, etc. In this regard, this method extracts valid distortion features in three steps: (1) introducing newly finer convolutional kernels which adaptively sample the feature maps at valid locations so as to fit the appearance of distorted objects precisely; (2) taking full advantage of context features by fusing high-level and low-level features; (3) improving the sub-network and aggregating the distortion features into a fisheye feature pyramid. As for precisely identifying predictable bounding boxes, we introduce a novel Distortion Shape Matching strategy which invokes irregular quadrilateral bounding boxes to precisely localize all kinds of distorted objects in fisheye images. Comparing the object locations obtained by rectangular bounding boxes and by irregular quadrilateral bounding boxes, as shown in Fig.1, we find that the latter can alleviate the influence of redundant information, unnecessary overlap, and inaccurate annotations.

To the best of our knowledge, there is no benchmark fisheye dataset for the multi-class object detection task. Thus, we are the first to create a fisheye-like version of PASCAL VOC [23] (called VOC-Fisheye). With the synthesized fisheye dataset, verification and evaluation of FisheyeDet can be completed with a virtual optical environment. Experimental results demonstrate that our method outperforms state-of-the-art methods in terms of mAP by 2.57%-18.14%.

Our contributions are summarized in four-folds:

- We propose FisheyeDet, a self-study and contour-based object detector in fisheye images. Instead of performing global parametric transformations and warping features, FisheyeDet effectively integrates distortion features representations learning and tighter bounding boxes locations refinement into the detector, which significantly improves the generalization capability of object detection in fisheye images.
- We design a No-prior Fisheye Representation Method to extract valid distortion features and construct an effective fisheye feature pyramid adaptively. In the process of learning the properties of distortion in fisheye images, manual design based on prior information is not involved, such as designing location-based convolutional kernels or correcting images.
- The designed Distortion Shape Matching utilizes irregular quadrilateral bounding boxes to tightly localize the distorted objects, which improves the accuracy of prediction.
- Both qualitative and quantitative experiments on the VOC-Fisheye dataset demonstrate the superiority of our method.

The rest of the paper is organized as follows: Section II examines previous related works. The network architecture of FisheyeDet is proposed in Section III-A. Each module of this network is detailed in Section III-B and Section III-C. And in Section III-D, the loss function of our network is described. Section IV introduces the generation of a synthesized fisheye dataset, the training strategy, and the quantitative and qualitative experiments.

II. RELATED WORK

In this section, we review the previous works related to our method from two aspects: object detection in large field of view (LFOV) and deformation modeling.

A. OBJECT DETECTION IN LARGE FIELD OF VIEW

Object detection is a fundamental problem in computer vision, and detection in LFOV is one of the branches. Current object detection methods in fisheye images are grouped into two categories: distortion correction-based and original LFOV image-based.

1) THE DISTORTION CORRECTION-BASED METHODS

The distortion correction-based methods usually include two stages: image warping and object detection. Image warping is the key to these methods. Silberstein *et al.* [24] first used Caltech Camera Calibration Toolbox to conduct the distortion

correction, then used a multiple-part based detection method to detect pedestrian, called Accelerated Feature Synthesis (AFS). Based on this AFS-based system, Levi *et al.* [19] presented an integrated system using temporal cues based on information from multiple subsequent video frames to reduce the system error rates. Bertozzi *et al.* [25] first used the equidistance mapping function to both correct the lens distortion and obtain a wide-angle view without strong aberrations, then they used a Soft-Cascade + Aggregated Channel Feature (ACF) classifier to detect vehicles and pedestrians. To maximize the usage of lost regions, Kim *et al.* [26] used directional interpolation to transform a fisheye image into an anamorphic image. Jeong *et al.* [27] utilized a distortion model to transform the distorted images to flat images, and then applied to the Histogram of Oriented Gradient (HOG) to detect cars on the flat images. Suhr *et al.* [17] transformed fisheye images via Mercator projection to reduce the impact of pedestrian shape variations, then the Viola-Jones detector was used to achieve pedestrian detection.

2) THE ORIGINAL LFOV IMAGE-BASED METHODS

Deng *et al.* [28] were the first to introduce deep learning to detect multi-class objects in fisheye images, validating the feasibility of the original LFOV image-based methods. After that, Yang *et al.* [29] compared the results of different detection algorithms that take equirectangular projection (ERP) images directly as inputs, showing that the network only produces a certain accuracy without projecting ERP images into conventional 2D images. Lee *et al.* [10] adjusted convolutional kernels and pooling operators to work in spherical coordinates so that the convolutional neural network (CNN)-based detectors could be used directly on spherical coordinates. SphereNet, proposed by Benjamin Coors *et al.* [11], could adapt the sample locations of the convolutional filters which effectively reverse distortions. In 360° images, the transformation is location-based, so Su *et al.* [12] proposed an approach to adjust the kernel shape based on its location on the sphere. To make the model easy to train and deploy, they further presented the Kernel Transformer Network (KTN). It learned a transformation that considered both spatial and cross-channel correlation.

However, some shortcomings exist in these methods: the strong dependency on handcrafted features or location-based convolutional kernels, and a remarkable loss of image quality during the geometric transformations, leading to some valid object features loss, particularly around the perimeter.

B. DEFORMATION MODELING

Deformation modeling is an extremely important research topic over a long period. Besides using object parts with a deformable configuration [30] to model objects, a lot of tremendous works have been done in designing translation-invariant features. There have been some noteworthy works including scale-invariant feature transform (SIFT) [31], oriented FAST and rotated BRIEF (ORB) [32], and deformable part-based models (DPM) [14]. Such works have a common flaw: the poor representation power caused by the handcrafted features, only applicable to limited geometric transformations. In the era of deep learning, Spatial transformer networks (STN) [33] was the first work to confirm the feasibility of learning translation-invariant features by CNNs. Scattering networks [34] and TIpooling [35] could learn different types of transformations. In addition, some researchers [36], [37] bent their effort to specific transformations. However, these works still cannot handle unknown transformations in more complex object tasks. Dai *et al.* [38] proposed Deformable ConvNets to effectively model geometric transformations in complex vision tasks since the offsets could be learned by adding deformable convolution and deformable RoIpooling modules.

III. THE PROPOSED METHODOLOGY

In this section, we elaborate on the self-study and contour-based object detector, FisheyeDet. We first briefly introduce the network architecture in Section III-A. Then we detail the No-prior Fisheye Representation Method in Section III-B. After that, we propose a novel strategy, named Distortion Shape Matching in Section III-C. Finally, we precisely describe the loss function of our method in Section III-D.

A. ARCHITECTURE

FisheyeDet is inspired by SSD [22], aiming to build an end-to-end trainable network to detect objects in fisheye images. The designed No-prior Fisheye Representation Method constructs a novel fisheye feature pyramid, yielding an end-to-end solution. Considering the characteristics of objects in fisheye images, the Distortion Shape Matching strategy is designed to tightly localize the distorted objects. Both of two novel designs are independent of the backbone architecture. Therefore, the architecture of FisheyeDet uses the same VGG-16 reduced backbone designed in SSD [22]. Then the Distortion Feature Extraction Layers are added on this backbone. In this part, the No-prior Fisheye Representation Method is used to adaptively extract valid distortion features. The Prediction Detection Layers are inserted after the Distortion Feature Extraction Layers. In this part, the Distortion Shape Matching strategy is applied to get the final outputs including precise locations and corresponding categories. The high-level overview and the detailed architecture of FisheyeDet are illustrated in Fig.2 and Fig.3, respectively. The details of each component in FisheyeDet are as follows:

1) MULTI-SCALE BACKBONE LAYERS

The architecture of FisheyeDet adopts the same VGG-16 reduced backbone designed in SSD [22]. More specifically, our network keeps the layers from conv1_1 to conv5_3 of VGG-16 architecture [39], and then converts the fc6 and fc7 (two fully-connected layers) of VGG-16 into conv6 and conv7 (convolutional layers). All of these constitute our Multi-scale Backbone Layers. After that, several modules can be added.

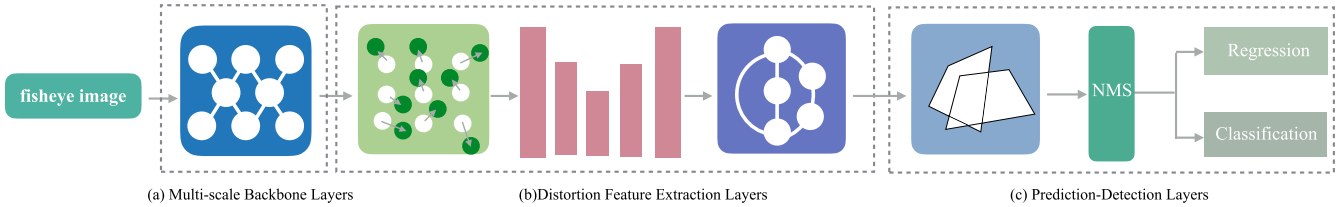


FIGURE 2. Overview of the proposed FisheyeDet, which consists of Multi-scale Backbone Layers, Distortion Feature Extraction Layers, and Prediction Detection Layers.

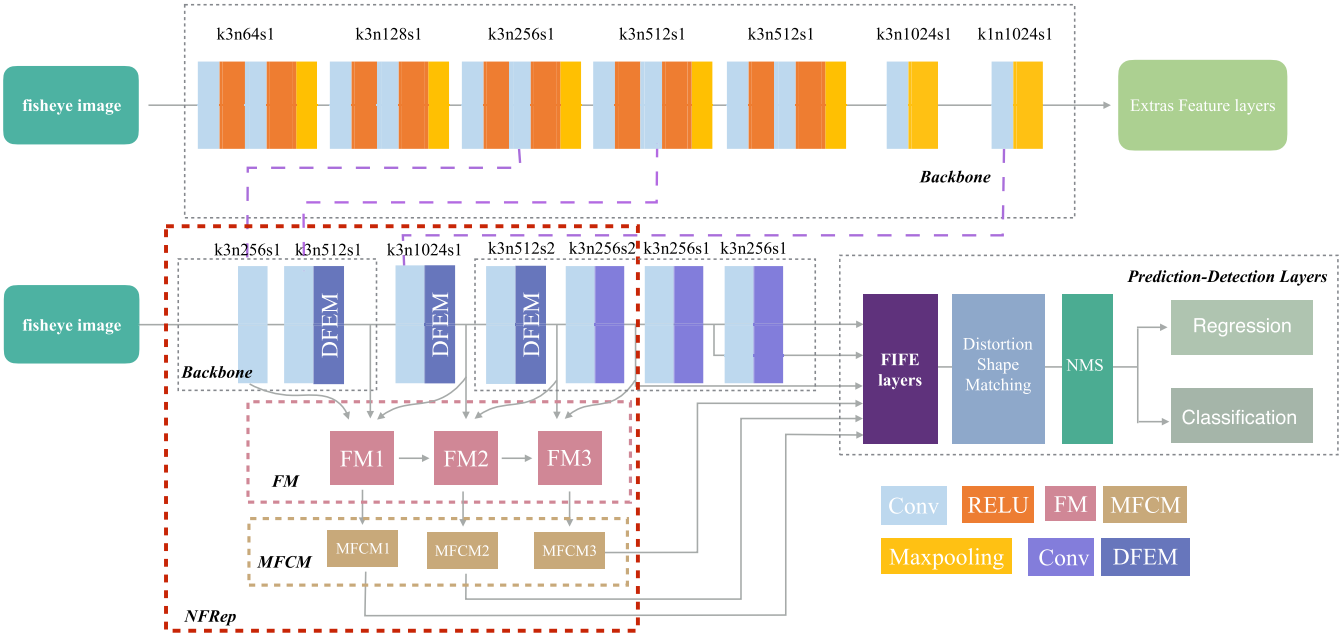


FIGURE 3. Details of the proposed FisheyeDet. It consists of Multi-scale Backbone Layers, Distortion Feature Extraction Layers, and Prediction Detection Layers. The No-prior Fisheye Representation Method (NFRep) is used in Distortion Feature Extraction Layers, which consists of Distortion Feature Extractor Module (DFEM), Fish-context Module (FM), and Multi-filter Feature Connections Module(MFCM). The outputs of Distortion Feature Extraction Layers are used as the input of the Prediction Detection Layers, denoting as feature pyramid (FIFE layers). At each location of a FIFE layer, the Distortion Shape Matching is used to output an n-dimensional vector for each anchor box. Non-maximum suppression (NMS) operation is applied during the test process to merge the outputs of the Distortion Shape Matching.

2) DISTORTION FEATURE EXTRACTION LAYERS

Our Distortion Feature Extraction Layers aim to extract fisheye features accurately. As we all know, most state-of-the-art methods [40]–[42] construct different feature pyramids to detect an object with variant sizes. In addition, the manually designed geometric transformations of fisheye images are not fit well the nature of the real distortion. Following these intuitions, we propose a self-study process, in which the properties of distortion in fisheye images can be learned by the network without introducing any design invoking the prior information. In this process, the key is that we design the No-prior Fisheye Representation Method to construct an effective fisheye feature pyramid for object detection, without prior information such as prespecified lens parameters, special calibration patterns, etc. Concretely, we design the Distortion Feature Extractor Module which replaces handcrafted convolutional kernels with deformable convolutional blocks. This newly introduced finer convolutional kernels will sample the feature maps at valid locations so as to fit the appearance of objects precisely. Then considering the importance of the context, we further design the Fish-context Module to fuse features from different levels by combining low-resolution, high-level features with high-resolution, low-level features. After that, the Multi-filter Feature Connections Module is proposed to improve the sub-network and aggregate the distortion features. The structure of each module is shown in Section III-B1, III-B2, and III-B3 for details.

3) PREDICTION DETECTION LAYERS

In this part, multiple output layers constitute our Fisheye Image Feature Extraction (FIFE) layers. And then the Distortion Shape Matching strategy allows us to tightly predict candidate object regions represented by irregular quadrilateral bounding boxes. After that, these regions undergo an efficient non-maximum suppression (NMS) process. Finally, we get detection results. In this stage, the Distortion Shape Matching strategy is proposed to remedy the dilemma of containing redundant information, unnecessary overlap, and inaccurate annotations. Simultaneously, a contour-based object localization will be obtained.

B. NO-PRIOR FISHEYE REPRESENTATION METHOD

The No-prior Fisheye Representation Method consists of three modules, i.e., Distortion Feature Extractor Module, Fish-context Module, and Multi-filter Feature Connections Module. All of the above modules make adaptively and accurately extracting distortion features possible. More details about each part are described below.

1) DISTORTION FEATURE EXTRACTOR MODULE

In real-world systems, owing to the unknown geometric transformations, deformation of the sampling grid is needed, via adaptively learning from preceding feature maps.

We employ the conv4_3 and conv7 of VGG-16 reduced backbone as parts of feature extraction layers and add several convolutional layers as the other feature extraction layers (Fig.3). The traditional convolutional unit samples the input feature maps at a fixed location, which is not sufficient for object detection in fisheye images. Herein, we utilize the deformable convolutional [38] blocks to achieve better receptive fields that cover the objects in fisheye images. The deformable convolution consists of two steps: (1) sampling over the input feature map x, with an offset Δp_n; (2) Weighted summation of sampling points, with a weighting coefficient w. For each location p_0 on the output feature map y, the formula is as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{1}$$

where R is a grid which defines the receptive field size and dilation, p_n represents each location in R. The offsets Δp_n can be learned from the preceding feature maps, via additional convolutional layers. Thus, introducing deformable convolution to construct Distortion Feature Extractor Module can learn receptive fields adaptively, especially when the geometric transformations are unknown.

2) FISH-CONTEXT MODULE

In general, low-level features are suitable for the objects with simple detailed information while high-level features are for objects with complex detailed information. Therefore, fully mining the distortion features from different levels plays a key role. U-Net [43], a U-shaped architecture, consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. In this way, the high-resolution features from the contracting path are combined with the upsampled output. By using this structure, It can propagate context information to higher-resolution layers, achieving multi-level feature fusion. Inspired by this structure, we design a tiny hourglass structure, named Fish-context Module. This module takes full advantage of context features by fusing high-level and low-level features to produce rich contextual information.

Fish-context Module fuses features from different levels by combining low-resolution, high-level features with high-resolution, low-level features. As illustrated in Fig 4, since the Fish-context Module takes three layers with different scales as input, we adopt one deconvolutional operation to expand the lower-resolution layer and one convolutional operation to reduce the higher-resolution layer before fusing features. Specifically, a convolutional layer and a learned deconvolutional layer reap the lower-level context and higher-level context respectively. In particular, a batch normalization layer and a RELU layer are added after each convolutional layer and deconvolutional layer.

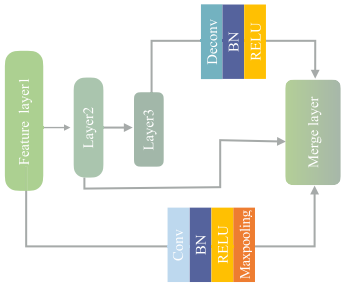


FIGURE 4. Illustration of Fish-context Module.

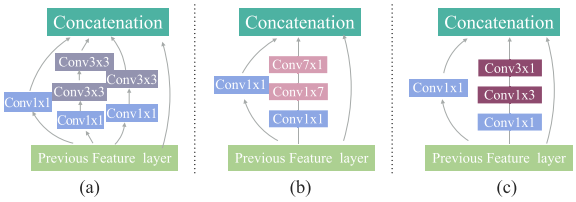


FIGURE 5. Illustration of Multi-filter Feature Connections Modules. Note that (a), (b), and (c) are the three types of Multi-filter Feature Connections Modules.

3) MULTI-FILTER FEATURE CONNECTIONS MODULE

The multi-scale CNN (MS-CNN) [44] points out that the way of improving accuracy is to improve the sub-network of each task. Moreover, recent evidence [45] reveals that the wider and deeper network which gets more accurate features is more conducive to classification and localization. In this regard, Inception network structure [46] has shown great capability. Meanwhile, it also generates multi-scale features by applying different scale convolutional kernels and concatenating all these outputs. Following these principles, we design the Multi-filter Feature Connections Module.

Multi-filter Feature Connections Module generates multi-scale features from a single-level layer by placing a group of convolutional operations with different convolutional kernels. Noticing that Multi-filter Feature Connections Module takes the features generated by Fish-context Module as input. This would allow each Multi-filter Feature Connections Module to reap the multi-level multi-scale features, namely, these features are more representative. Meanwhile, to explicitly extract the distortion features of different layers, we design three types of Multi-filter Feature Connections Modules. Fig.5 (a), (b), and (c) are the detailed structure, respectively.

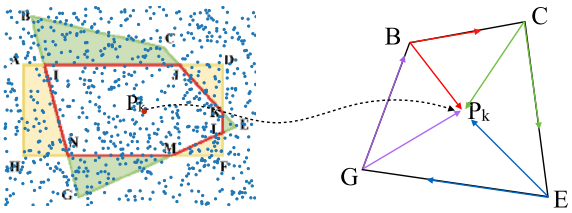


FIGURE 6. Irregular IoU computation. The left part is an example of irregular IoU computation. The right part is an example of using the Vector Cross Product to estimate the location of the point P_k.

Noticing that before 3×3 convolutions and 7×7 convolutions, a 1×1 convolution is used to compute dimension reductions. This method significantly reduces the parameter count. Besides, Inception v2 [46] pinpoints that the effect of a $1 \times n$ convolution followed by a $n \times 1$ convolution is similar to a $n \times n$ convolution. Using this trick, it can save computation cost.

C. DISTORTION SHAPE MATCHING

As described in the previous parts, the proposal of the Distortion Shape Matching strategy has innovations in localizing the objects in fisheye images. In this strategy, novel irregular quadrilateral bounding boxes are used to overcome the challenges shown in Fig.1 (a) redundant information, (b) unnecessary overlap, and (c) information loss. This strategy contains two important parts:

1) IRREGULAR INTERSECTION OVER UNION (IoU) COMPUTATION

Using an 8-points representation method is a contour-based way to describe the objects in fisheye images. However, the overlap formed by this method is a polygonal area, so the previous methods [21], [22], [47] of IoU computation between the ground truth box and every anchor box may lead to an inaccurate IoU and further ruin the proposal learning. Therefore, we put forward a simplified and statistical-based calculation method to compute the irregular IoU, as shown in Algorithm 1. Fig.6 visualizes a geometric principle.

Algorithm 1 Irregular IoU Completion

Input: $Q_1, Q_2, \ldots, Q_N$: Quadrilateral; N: Randomly generate N sample points;
Output: IoU between quadrilateral pairs: IoU

```
P_k: kth random sample point;
for each pair < Q_i, Q_j > (i<j) do
    Points Set PSet_{Q_i}, PSet_{Q_j}
    for k = 1 to N do
        Estimate the location of the point P_k
        If P_k is inside the Q_i, add P_k to PSet_{Q_i}
        If P_k is inside the Q_j, add P_k to PSet_{Q_j}
    end for
    Compute intersection I of PSet_{Q_i} and PSet_{Q_j}
    IoU_{i,j} ⇐ Num(I) / (Num(PSet_{Q_i}) + Num(PSet_{Q_j}) − Num(I))
end for
```

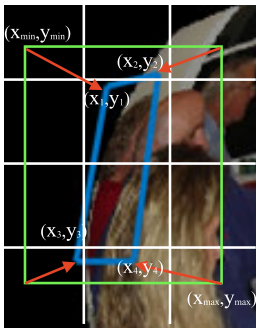


FIGURE 7. Illustration of learning of irregular quadrilateral bounding boxes. Note that the green box is the best matched anchor box, and the blue box is the ground truth box. The red arrows mean the offsets between the best matched anchor box and the ground truth box.

To compute the IoU for each pair $< Q_i, Q_j >$, the first step is to uniformly sample N points ($P_k, k \in [1, N]$). Then, we estimate the location of each sample point P_k. In this step (Lines 5-7 in Algorithm 1), considering the fact that in our paper every Q_i is a convex quadrilateral, we use the Vector Cross Product method (2) to estimate the point P_k whether it is inside an irregular quadrilateral or not:

$$\begin{aligned} BC \times BP_k > 0, \quad CE \times CP_k > 0 \\ EG \times EP_k > 0, \quad GB \times GP_k > 0 \end{aligned} \tag{2}$$

where B, C, E, and G are the four vertices of a quadrilateral, P_k is a random sample point (Shown in Fig.6), if all four formulas are established, then the point P_k is inside the quadrilateral. It's remarkable that those points inside the ground truth box will all be reserved for sharing computation. Following this way, IoU could be easily computed.

2) LEARNING OF IRREGULAR QUADRILATERAL BOUNDING BOXES

In this part, we adopt 8-points $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ to annotate the distorted ground truth box, instead of the conventional 4-points. The outputs of Distortion Shape Matching strategy include best matched anchor boxes and the offsets between the anchor boxes and the corresponding ground truth boxes.

By calculating IoU between each anchor box and ground truth box, the predictor can get the offsets from the best matched anchor box at each location (see Fig.7 for an example). More precisely, let ground truth box $gt = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, anchor box is a rectangle denoted as: $ab = (x_{min}, y_{min}, x_{max}, y_{max})$. From the given coordinates, we calculate the width and height of the anchor box respectively $w_{ab} = x_{max} - x_{min}$, $h_{ab} = y_{max} - y_{min}$. The offsets are calculated as follows:

$$\begin{aligned} \Delta x_i' = \frac{x_i - x_{min}}{w_{ab}}, \quad \Delta x_j' = \frac{x_j - x_{max}}{w_{ab}} \\ \Delta y_i' = \frac{y_i - y_{min}}{h_{ab}}, \quad \Delta y_j' = \frac{y_j - y_{max}}{h_{ab}} \end{aligned} \tag{3}$$

where $i = 1, 3$, $j = 2, 4$. This is considered as a direct regression from a rectangle bounding box to the best matched quadrilateral ground truth box.

D. LEARNING

Joint localization loss and confidence loss is the popular and powerful loss function in the object detection task [21], [22], [47]–[50], so we adopt a similar form to define the loss function. The loss function is defined as:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{4}$$

where x is the match indication matrix. c is the confidence, l is the predicted location, and g is the ground truth location, N is the number of anchor boxes that match ground truth boxes. Noticing that in the match indication matrix x, if the i-th anchor box and the j-th ground truth box match according to the Distortion Shape Matching strategy, x_{ij} is set to 1, otherwise 0. Additionally, α is set to 1.

As the image resolution has a certain loss, some positive information is lost, which leads to an imbalance between positive and negative samples during training. Herein, we introduce hard negative mining [22] and focal loss [51] to address this problem. Thus, the smooth $L1$ loss [47] is for L_{loc} and the focal loss is for L_{conf}. The training objective is to minimize this loss function (see equation (4)).

IV. EXPERIMENTS

In this section, we evaluate our method in fisheye images. To compensate for the lake of benchmark fisheye datasets for the multi-class object detection task, we are on the first attempt to address this problem by creating a fisheye-like version of PASCAL VOC, called VOC-Fisheye in Section IV-A. And in Section IV-C, we evaluate our FisheyeDet on the VOC-Fisheye dataset. Finally, extensive ablation studies are performed to validate the effectiveness of our approach in Section IV-D.

A. SYNTHESIZED DATASETS–VOC-FISHEYE DATASET

Convolutional neural networks (CNNs) are potential methods for mining distortion features of the objects in fisheye images. Exploring the reasons, one of the indispensable factors is the large-scale training datasets. Although various public datasets, e.g., PASCAL VOC [23], ImageNet [52], and COCO [53] are available for the identification of multiple objects, they aim at generic object detection, not specific for fisheye object detection.

Considering the difference between traditional images and fisheye images, when the models based on pre-trained on the generic datasets are directly applied to the fisheye object detection task, the results are often unsatisfactory. Besides, it is expensive and time-consuming to generate large-scale datasets by collecting data and corresponding annotations. Instead of using the above-mentioned method, we produce a synthesized training fisheye dataset based on an existing perspective image dataset–PASCAL VOC [23]. To the author's best knowledge, it is the first public fisheye dataset for the multi-class object detection task by transforming a known perspective image dataset. Under the following parts, we introduce a procedure for generating synthesized fisheye images.

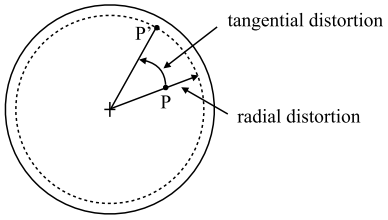


FIGURE 8. Fisheye-like Distortion. *P* represents the perspective imaging point. *P'* represents the fisheye imaging point. Radial distortion means the offset of the image position along the imaging radius direction, and the tangential distortion indicates the offset of the image pixel position along the tangential direction of the imaging point.

1) FISHEYE-LIKE DISTORTION

Camera lens distortions roughly contain two categories: tangential distortion and radial distortion [54]. For normal cameras, radial distortion can be considered negligible in many applications, but for fisheye lenses, it cannot be ignored due to adverse effects [55]. Fig.8 visualizes the fisheye-like distortion. In this paper, we only consider the radial distortion without tangential distortion since the tangential distortion is caused by the manufacturing process [56].

To visually approximate the appearance of fisheye images, we first normalize the coordinates of the original images (Fig.9(a)), which means that $(0, 0)$ represents the center and $(\pm 1, \pm 1)$ are the coordinates of four corners, respectively. Then we introduce the radial distortion [57], in which straight lines bend outward from the center of the image. Specifically, the coordinate mapping between the pixels (x, y) on an original image and corresponding pixels (x', y') in a fisheye image can be defined as:

$$(x', y') = (x\sqrt{1 - \frac{y^2}{2}}, y\sqrt{1 - \frac{x^2}{2}}) \tag{5}$$

Furthermore, we introduce a coordinate scaling factor $e^{-r^2/n}$ to control the severity of its distortion.

$$(x'', y'') = (x'e^{-r^2/n}, y'e^{-r^2/n}) \tag{6}$$

where r is the distance from coordinates (x', y') to the center.

2) ANNOTATION

After leveraging the procedure mentioned above, we get 49653 fisheye-looking training images and 14856 testing images. In order to precisely describe the object contour, we adopt irregular quadrilaterals to annotate the locations of objects. The parameters are $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, where (x_i, y_i) are the location of the four corners of the ground truth box.

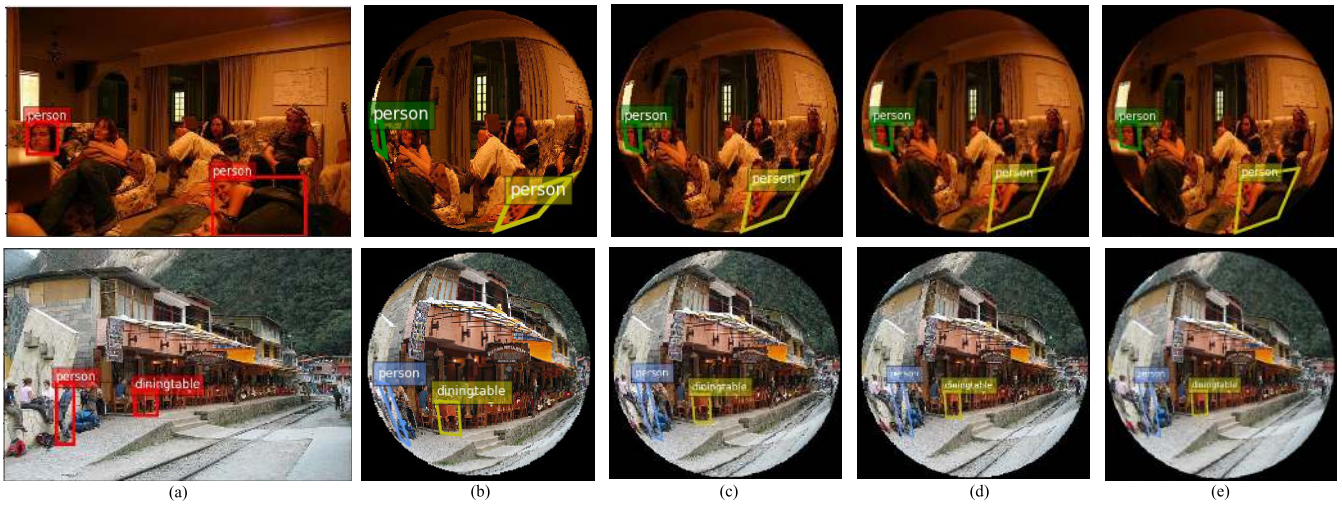


FIGURE 9. Representative samples of the VOC-Fisheye dataset. (a) original images from the PASCAL VOC [23], (b),(c),(d) and(e) fisheye images with different distortions.

TABLE 1. The feature extraction layers.

layers	conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2
Resolution	38×38	19×19	10×10	5×5	3×3	1×1

3) FURTHER DETAILS OF VOC-FISHEYE

To verify the network without prior information, we expect our dataset contains different distortions. Based on this idea, an original image from the PASCAL VOC [23] can be transformed into several of fisheye images by introducing different coordinate scaling factors. Some examples of the VOC-Fisheye dataset can be seen in Fig.9

B. IMPLEMENTATION DETAILS

The VOC-Fisheye 'trainval' dataset (the same as PASCAL VOC) is used to train our model. It consists of 20 object categories, each of which has the annotated ground truth location (8-points) and corresponding category information. During the training stage, the input images are randomly cropped and resized to 300×300. FisheyeDet is trained with stochastic gradient descent (SGD) optimizer with batch size 64. The learning rate *lr* is set to 0.001, and the nesterov momentum to 0.9 with the weight decay of 5×10^{-4}. The learning rate of every 80 epochs decreases once. In addition, the aspect ratios of anchor boxes are set to 1, 2, 3, 5, 1/2, 1/3, 1/5. Average precision (AP) is used as an evaluation protocol for each object category and mean AP (mAP) [23] is computed over all object categories.

C. COMPARISON TO THE STATE-OF-THE-ART

In this subsection, we quantitatively evaluate our FisheyeDet on the VOC-Fisheye dataset and compare it with the other seven state-of-the-art object detection methods [8], [21], [22], [47]–[50].

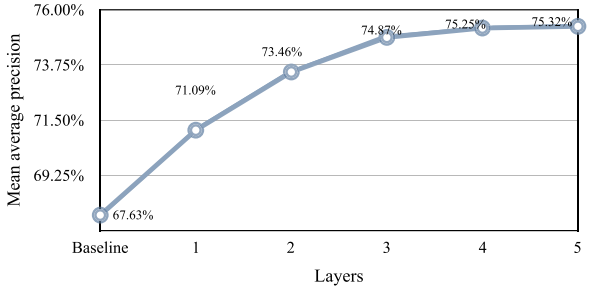


FIGURE 10. Results of using No-prior Fisheye Representation Method in different numbers of layers (of 3×3 convolutions) in the original SSD feature extraction network. With the increase of layers, we report the results on VOC-Fisheye 2007 test.

In Fig.10, we evaluate the effect of the No-prior Fisheye Representation Method based on the original SSD feature extraction network (shown in Table 1). It obviously shows that: (1) the value of mAP steadily improves with the increasing layers. (2) the growth of mAP gradually decreases from 3.46% to 0.07%, indicating that the improvement saturates when using 3 layers (conv4_3, conv7, conv8_2). So we introduce the No-prior Fisheye Representation Method to the last three layers for training and testing. Other methods, which are the representative of the current mainstream models of object detection, are re-trained and implemented according to the details provided in the corresponding papers. Results are given in Table 2. Our method (FisheyeDet) improves mAP to 74.87%, which is surpassing all exiting methods (e.g.,63.42% mAP of Faster R-CNN, 56.73% mAP of YOLO, 68.92% mAP of YOLO V3, 67.63% mAP of SSD, 70.01% mAP of DSSD, 70.82% mAP of RSSD, and 72.30% mAP of ATSS) on the VOC-Fisheye dataset.

Additionally, in order to further reinforce and confirm the superiority of our method, we also evaluate FisheyeDet on the COCO-Fisheye dataset and the AP_{bbox} is boosted by

FIGURE 11. Baseline vs the proposed FisheyeDet. The first and third lines images are based on the baseline for detecting objects and locating the objects under different distortions. The second and last line images are based on the FisheyeDet for detecting objects and locating the objects under different distortions.

TABLE 2. Comparison to the state-of-the-art methods.

Methods	Input	Train	Test	mAP
Faster R-CNN [47]	$\sim 1000\times600$	2007+2012	2007	63.42%
YOLO [21]	448	2007+2012	2007	56.73%
YOLO V3 [48]	416	2007+2012	2007	68.92%
SSD [22]	300	2007+2012	2007	67.63%
DSSD [49]	321	2007+2012	2012	70.01%
RSSD [50]	300	2007+2012	2012	70.82%
ATSS [8]	$\leq1333\times800$	2007+2012	2012	72.30%
Ours (FisheyeDet)	300	2007+2012	2007	**74.87%**

0.7%-14.4%. The descriptions of the COCO-Fisheye dataset, implementation details, and experimental results are detailed in Appendix A.

D. ABLATION STUDIES

To understand FisheyeDet better, we carry out a series of controlled experiments on the VOC-Fisheye dataset to examine the impact of different components in our method. The baseline is a simple detector based on the original SSD, with 300 × 300 input size and VGG-16 reduced backbone.

1) NO-PRIOR FISHEYE REPRESENTATION METHOD

a: DISTORTION FEATURE EXTRACTOR MODULE

Without the prior information, refining the process of fisheye distortion feature extraction is a crucial part of the object detector. And the deformable convolutional blocks have shown its capability in learning representative image features for the general object detection task [38]. Base on these viewpoints, we introduce the deformable convolutional blocks to construct Distortion Feature Extractor Module. To indicate the effect of this module, we adopt two different forms (3 × 3 convolutions and 5 × 5 convolutions) in our detector and report the results in Table 3.

As shown in Table 3, we observe that it has much better mAP (3.03%) for the module with 5 × 5 convolutions, and has less improvement in mAP (2.64%) for the module with 3 × 3 convolutions. Thus, Distortion Feature Extractor Module is effective and necessary in fisheye images without

TABLE 3. Ablation studies of FisheyeDet: effect of Distortion Feature Extractor Module. a: 3 × 3 convolutions, b: 5 × 5 convolutions. Animals: bird, cat, cow, dog, horse, sheep; Vehicles: aeroplane, bicycle, boat, bus, car, motorbike, train; Indoors: bottle, chair, diningtable, pottedplant, tvmonitor, sofa; Person: person; mAP: the mean AP of all object categories.

a	b	Animals	Vehicles	Indoors	Person	mAP
Baseline		73.66%	73.34%	55.08%	66.89%	67.63%
√		75.69%	73.72%	61.09%	68.74%	70.27%
	√	75.71%	73.90%	62.07%	69.15%	70.66%

TABLE 4. Ablation studies of FisheyeDet: effect of Fish-context Module. DFEM: Distortion Feature Extractor Module; FM: Fish-context Module; Animals: bird, cat, cow, dog, horse, sheep; Vehicles: aeroplane, bicycle, boat, bus, car, motorbike, train; Indoors: bottle, chair, diningtable, pottedplant, tvmonitor, sofa; Person: person; mAP: the mean AP of all object categories.

Method	Animals	Vehicles	Indoors	Person	mAP
Baseline	73.66%	73.34%	55.08%	66.89%	67.63%
Baseline+FM	76.27%	74.98%	56.68%	68.44%	69.55%
Baseline + DFEM + FM	78.77%	77.47%	61.45%	69.96%	72.68%

prior information. Even though the 5 × 5 convolutions can achieve better results, the training speed is very slow. This is understandable: the more perception fields obtained, the more effective features of object instances extracted. However, the offsets that need to learn (change from 3 × 3 to 5 × 5) are also correspondingly increased (see Section III-C), which consumes more time. Therefore, in the remaining experiments, we use 3 × 3 deformable convolutional blocks in the feature extraction network.

b: FISH-CONTEXT MODULE

Although Distortion Feature Extractor Module can boost detection performance, each layer in the feature pyramid contains single-level information, which is similar to the original SSD [22]. As discussed in Section III-B2, each Fish-context Module fuses features from different levels by combining high-level features with low-level features. That is, it may have additional information about objects which are larger or smaller than the objects. Table 4 presents the effect of the Fish-context Module.

By adding Fish-context Module, we can see that the gaining is improving from 67.63% to 72.68%. This significant improvement illustrates the effectiveness of joining high-level features with the low-level features, resulting in a better representation of objects in fisheye images.

c: MULTI-FILTER FEATURE CONNECTIONS MODULE

Multi-filter Feature Connections Module aims to generate multi-scale features. To reveal the influence of the Multi-filter Feature Connections Module on performance, we carry out a comparative experiment. Table 5 reports the effect of the Multi-filter Feature Connections Module.

TABLE 5. Ablation studies of FisheyeDet: effect of Multi-filter Feature Connections Module. MFCM: Multi-filter Feature Connections Module; NFRep: No-prior Fisheye Representation Method consist of Distortion Feature Extractor Module, Fish-context Module, and Multi-filter Feature Connections Module; Animals: bird, cat, cow, dog, horse, sheep; Vehicles: aeroplane, bicycle, boat, bus, car, motorbike, train; Indoors: bottle, chair, diningtable, pottedplant, tvmonitor, sofa; Person: person; mAP: the mean AP of all object categories.

Method	Animals	Vehicles	Indoors	Person	mAP
Baseline	73.66%	73.34%	55.08%	66.89%	67.63%
Baseline + MFCM	76.23%	74.55%	56.61%	68.20%	69.35%
Baseline + FM + MFCM	76.63%	76.48%	57.06%	68.87%	70.32%
Baseline + NFRep	80.76%	78.03%	62.79%	70.53%	73.90%

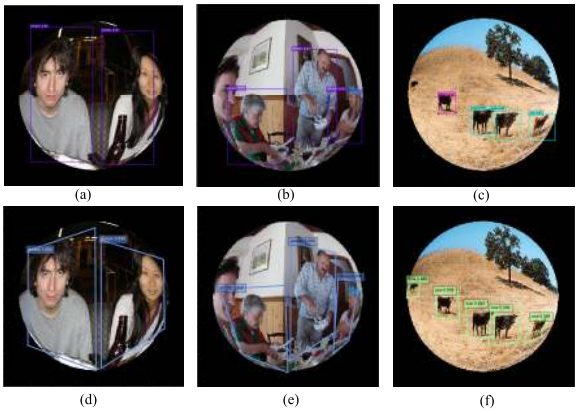


FIGURE 12. Visualization of Distortion Shape Matching. (a) (b) (c) use rectangular bounding boxes for localizing objects in fisheye images, and (d) (e) (f) use irregular quadrilateral bounding boxes for localizing objects in fisheye images. By comparing (a) (b) (c) and (d) (e) (f), the problems of redundant information, unnecessary overlaps, and inaccurate annotations are solved, respectively.

Using Multi-filter Feature Connections Module alone, the proposed module is sightly better than the original SSD. By integrating the Distortion Feature Extractor Module and Fish-context Module, the mAP rises nearly 6.27%.

2) DISTORTION SHAPE MATCHING

Using the No-prior Fisheye Representation Method can enhance the detection capability, yet it is limited by rectangular bounding boxes such as redundant information, unnecessary overlaps, and inaccurate annotations (shown in Fig.1). To solve this limitation, we introduce the Distortion Shape Matching strategy into our network.

As shown in Fig.12, Distortion Shape Matching strategy can solve the problems of redundant information, unnecessary overlaps, and incorrect annotations, respectively. And it indicates that this strategy generates a tighter object region at the prediction stage.

In order to expose the effect of the Distortion Shape Matching strategy, we do further study on this. As Table 6 shows, the architectures with this strategy are better than those

TABLE 6. Ablation studies of FisheyeDet: effect of Distortion Shape Matching (DSM). FM: Fish-context Module; MFCM: Multi-filter Feature Connections Module; NFRep: No-prior Fisheye Representation Method consist of Distortion Feature Extractor Module, Fish-context Module, and Multi-filter Feature Connections Module; Animals: bird, cat, cow, dog, horse, sheep; Vehicles: aeroplane, bicycle, boat, bus, car, motorbike, train; Indoors: bottle, chair, diningtable, pottedplant, tvmonitor, sofa; Person: person; mAP: the mean AP of all object categories.

Method	Animals	Vehicles	Indoors	Person	mAP
Baseline	73.66%	73.34%	55.08%	66.89%	67.63%
Baseline + DSM	75.90%	75.81%	60.99%	68.78%	71.04%
Baseline + FM	76.27%	74.98%	56.68%	68.44%	69.55%
Baseline + FM + DSM	78.17%	76.34%	62.21%	69.79%	72.32%
Baseline + MFCM	76.23%	74.55%	56.61%	68.20%	69.35%
Baseline + MFCM + DSM	78.03%	76.04%	62.30%	69.46%	72.19%
Baseline + NFRep	**80.76%**	78.03%	62.79%	70.53%	73.90%
Ours (FisheyeDet)	78.23%	**79.15%**	**66.91%**	**72.42%**	**74.87%**

without this strategy in mAP. Specifically, if we introduce the Distortion Shape Matching strategy into the Baseline, the mAP is significantly increased by 3.41%. Such observation remains true when applied to the other three methods including the Baseline + FM, the Baseline + MFCM, and the Baseline + NFRep. It verifies that directly predicting the compact irregular quadrilateral bounding boxes is essential. Interestingly, the Baseline + NFRep (i.e., FisheyeDet without DSM) is slightly better than our detector in Animals. By revisiting the VOC-Fisheye and experimental results, we conjecture that the degradation is probably due to the following aspect: our detector is relatively complex, which requires more samples to enhance its capability. However, some categories belonging to Animals have too few samples, leading to a slight degradation compared with the Baseline + NFRep (i.e., FisheyeDet without DSM) in AP of these categories and consequent performance degradation in Animals categories. Nevertheless, we also claim that this module can capture the distortion information in fisheye images.

Integrating the aforementioned two parts (Section IV-D1 and Section IV-D2), we can see that the mAP is boosted by 7.24% and the AP of each category increases 4.57%, 5.81%, 11.89%, 5.53% with respect to Animals, Vehicles, Indoors, Person, respectively. This sharp increase demonstrates the favorable robustness and generalization capability of proposed FisheyeDet, no matter how severe the distortion of the objects will be. Although there are some objects missed, most of the objects can be robustly recalled.

V. CONCLUSION

In this paper, we propose a self-study and contour-based object detector in fisheye images, denoted as FisheyeDet, combined with the "No-prior Fisheye Representation Method" and "Distortion Shape Matching". No-prior Fisheye Representation Method requires neither a prior lens design specifications nor a special calibration pattern to adaptively generate the sampling locations and construct an effective feature pyramid. To do this, we first construct the Distortion Feature Extractor Module which improves the ability to extract fisheye distortion features. Besides, we utilize the Fish-context Module to fuse context features. In addition, Multi-filter Feature Connections Module is introduced to aggregate the distortion features. Furthermore, we leverage distortion characteristics of fisheye to guide the bounding box, designing a novel Distortion Shape Matching strategy to precisely localize all kinds of distorted objects in fisheye images. These works are helpful in building an end-to-end network. Due to the lack of benchmark fisheye datasets for the multi-class object detection task, we are on the first attempt to put forward the VOC-Fisheye dataset. Experiments on this dataset show that FisheyeDet significantly outperforms the state-of-the-art conventional methods. The FisheyeDet allows arbitrary-distorted input, keeps good generalization ability, and shows higher accuracy.

APPENDIX

In this part, in order to further reinforce and confirm the superiority of our method, we compare the experimental results of the proposed FisheyeDet with state-of-the-art object detection methods [8], [22], [47], [49], [58] in Table 7 on the COCO-Fisheye dataset.

TABLE 7. Detection results comparisons in terms of accuracy performance.

Methods	Input	Train	Test	AP_{bbox}
Faster R-CNN [47]	$\sim 1000\times600$	trainval35k	minival	15.2
SSD [22]	300	trainval35k	minival	18.8
DSSD [49]	321	trainval35k	minival	21.7
RefineDet [58]	320	trainval35k	minival	23.3
ATSS [8]	$\leq1333\times800$	trainval35k	minival	28.9
Ours (FisheyeDet)	300	trainval35k	minival	**29.6**

DATASET

The COCO-Fisheye dataset is constructed from the COCO [53] dataset by using the fisheye production method mentioned in Section IV-A. This dataset contains 354861 fisheye-looking training images and 15000 testing images. The training images and testing images are from trainval35k and minival, respectively. In this dataset, it consists of 80 object categories.

IMPLEMENTATION DETAILS

During the training stage, the input images are randomly cropped and resized to 300 $\times$ 300. FisheyeDet is trained with stochastic gradient descent (SGD) optimizer with batch size 64. The learning rate lr is set to 0.001, and decreases at the 80-th and the 100-th epochs, respectively. The nesterov

momentum to 0.9 with the weight decay of 5×10^{-4}. In addition, the aspect ratios of anchor boxes are set to 1, 2, 3, 5, 1/2, 1/3, 1/5. For evaluation metrics, we adopt the standard mean average-precision scores averaged over multiple IoU thresholds.

EXPERIMENTAL RESULTS

As shown in Table7, our method obtains 29.6% AP_{bbox}, which is surpassing all exiting methods (e.g., 15.2% AP_{bbox} of Faster R-CNN [47], 18.8% AP_{bbox} of SSD [22], 21.7% AP_{bbox} of DSSD [49], 23.3% AP_{bbox} of RefineDet [58], and 28.9% AP_{bbox} of ATSS [8]) on the COCO-Fisheye dataset.

REFERENCES

[1] D. Schmalstieg and T. Hollerer, *Augmented Reality: Principles and Practice*. Reading, MA, USA: Addison-Wesley, 2016.
[2] D. Tseng, C. Chen, and C. Tseng, "Automatic detection and tracking in multi-fisheye cameras surveillance," *Int. J. Comput. Electr. Eng.*, vol. 9, no. 1, pp. 370–383, 2017.
[3] Á. Sáez, L. Bergasa, E. López-Guillén, E. Romera, M. Tradacete, C. Gómez-Huélamo, and J. del Egido, "Real-time semantic segmentation for fisheye urban driving images based on ERFNet," *Sensors*, vol. 19, no. 3, p. 503, 2019.
[4] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2965–2974.
[5] K. Chen, J. Li, W. Lin, J. See, J. Wang, L. Duan, Z. Chen, C. He, and J. Zou, "Towards accurate one-stage object detection with AP-loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5119–5127.
[6] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.
[7] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling, "CBNet: A novel composite backbone network architecture for object detection," 2019, *arXiv:1909.03625*. [Online]. Available: http://arxiv.org/abs/1909.03625
[8] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," 2019, *arXiv:1912.02424*. [Online]. Available: http://arxiv.org/abs/1912.02424
[9] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," 2019, *arXiv:1911.09070*. [Online]. Available: http://arxiv.org/abs/1911.09070
[10] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon, "SpherePHD: Applying CNNs on a spherical PolyHeDron representation of 360 degree images," 2018, *arXiv:1811.08196*. [Online]. Available: http://arxiv.org/abs/1811.08196
[11] B. Coors, A. Paul Condurache, and A. Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 518–533.
[12] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 529–539.
[13] Y.-C. Su and K. Grauman, "Kernel transformer networks for compact spherical convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9442–9451.
[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[15] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 237–244.
[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[17] J. K. Suhr and H. G. Jung, "Rearview camera-based backover warning system exploiting a combination of pose-specific pedestrian recognitions," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1122–1129, Apr. 2018.
[18] I. Baek, A. Davies, G. Yan, and R. R. Rajkumar, "Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 447–452.
[19] D. Levi and S. Silberstein, "Tracking and motion cues for rear-view pedestrian detection," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 664–671.
[20] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perroton, and P. Perez, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," 2019, *arXiv:1905.01489*. [Online]. Available: http://arxiv.org/abs/1905.01489
[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 21–37.
[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[24] S. Silberstein, D. Levi, V. Kogan, and R. Gazit, "Vision-based pedestrian detection for rear-view cameras," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 853–860.
[25] M. Bertozzi, L. Castangia, S. Cattani, A. Prioletti, and P. Versari, "360° detection and tracking algorithm of both pedestrian and vehicle using fisheye images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 132–137.
[26] H. Kim, J. Jung, and J. Paik, "Fisheye lens camera based surveillance system for wide field of view monitoring," *Optik*, vol. 127, no. 14, pp. 5636–5646, Jul. 2016.
[27] J.-S. Jeong, H.-T. Kim, B. Kim, and S.-B. Cho, "Wide rear vehicle recognition using a fisheye lens camera image," in *Proc. IEEE Asia–Pacific Conf. Circuits Syst. (APCCAS)*, Oct. 2016, pp. 691–693.
[28] F. Deng, X. Zhu, and J. Ren, "Object detection on panoramic images based on deep learning," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 375–380.
[29] W. Yang, Y. Qian, J.-K. Kamarainen, F. Cricri, and L. Fan, "Object detection in equirectangular panorama," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2190–2195.
[30] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 264–271.
[31] R. K. Gupta and S. D. Senturia, "Pull-in time dynamics as a measure of absolute pressure," in *Proc. IEEE 10th Annu. Int. Workshop Micro Electro Mech. Syst. Invest. Micro Struct., Sensors, Actuat., Mach. Robots*, Nagoya, Japan, Jan. 1997, pp. 290–294.
[32] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, vol. 11, no. 1, p. 2.
[33] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[34] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.
[35] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, "TI-POOLING: Transformation-invariant pooling for feature learning in convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 289–297.
[36] R. Gens and P. M. Domingos, "Deep symmetry networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2537–2545.
[37] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5028–5037.
[38] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[40] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4875–4884.

[41] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[42] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A single-shot object detector based on multi-level feature pyramid network," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 9259–9266, Jul. 2019.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Medic. Image Comput. Comput.-Assist. Intervent. (MICCAI)*. Springer, 2015, pp. 234–241.

[44] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 354–370.

[45] X. Tang, D. K. Du, Z. He, and J. Liu, "PyramidBox: A context-assisted single shot face detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 797–813.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[48] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[49] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[50] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," 2017, *arXiv:1705.09587*. [Online]. Available: http://arxiv.org/abs/1705.09587

[51] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 2980–2988.

[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[53] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Sep. 2014, pp. 740–755.

[54] C. B. Duane, "Close-range camera calibration," *Photogramm. Eng*, vol. 37, no. 8, pp. 855–866, 1971.

[55] C. Hughes, M. Glavin, E. Jones, and P. Denny, "Review of geometric distortion compensation in fish-eye cameras," in *Proc. IET Irish Signals Syst. Conf. (ISSC)*, 2008, pp. 162–167.

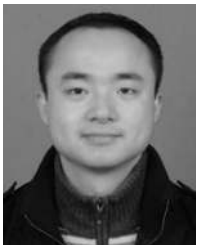
[56] C. Hughes, M. Glavin, E. Jones, and P. Denny, "Wide-angle camera technology for automotive applications: A review," *IET Intelligent Transport Systems*, vol. 3, no. 1, pp. 19–31, 2009.

[57] J. Fu, S. Ranjbar Alvar, I. V. Bajic, and R. G. Vaughan, "FDDB-360: Face detection in 360-degree fisheye images," 2019, *arXiv:1902.02777*. [Online]. Available: http://arxiv.org/abs/1902.02777

[58] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.

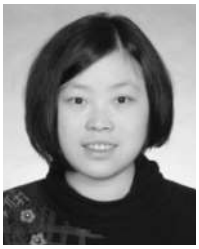
TANGWEI LI received the B.S. degree in information security from Hangzhou Dianzi University, in 2016. She is currently pursuing the M.S. degree with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and also with the University of Chinese Academy of Sciences.

Her research interests include object detection and recognition, and image processing.



GUANJUN TONG received the Ph.D. degree from the Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai, China, in 2009.

He is currently a Professor and the Master's Supervisor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. His research interests include the signal processing and the application of wireless sensor networks.



HONGYING TANG received the Ph.D. degree from Shanghai Jiao Tong University (SJTU), China, in 2015.

She is currently an Engineer with the Science and Technology on Microsystem Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. Her research interest includes unmanned aerial vehicle (UAV) communications.

BAOQING LI received the Ph.D. degree from the State Key Laboratory of Transducer Technology, Shanghai Institute of Metallurgy, Chinese Academy of Sciences, Shanghai, China, in 2000.

He is currently a Professor and the Ph.D. Supervisor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. His current research interest includes the application of wireless sensor networks.

BO CHEN received the B.S. degree from Hangzhou Dianzi University, in 2018. He is currently pursuing the M.S. degree with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and also with the University of Chinese Academy of Sciences.

His research interests include computer vision, and image processing.