引用格式：ZHANG Saiqiang, SI Shaofeng, LU Bin, et al. A Person Detection Algorithm in Fisheye Images Based on Rotated Boxes[J]. Acta Photonica Sinica, 2021, 50(10):1010003
张赛强,司绍峰,鲁斌,等.一种基于旋转框的鱼眼图像行人检测算法[J].光子学报,2021,50(10):1010003

# 一种基于旋转框的鱼眼图像行人检测算法

张赛强[1,2],司绍峰[1,2],鲁斌[1,2],李庆[2],陈本瑶[3],戎安心[3]
(1 中国科学院微电子研究所,北京 100029)
(2 中国科学院大学,北京 100049)
(3 湖州市特种设备检测研究院,浙江 湖州 313000)

**摘　要**:由于鱼眼图像存在几何畸变,导致现有的基于鱼眼图像的行人检测算法存在着检测准确率低以及后处理计算复杂度高的问题。针对上述问题,提出了一种使用旋转边界框的行人检测算法。首先,算法采用无锚框网络结构,使用中心热图预测边界框的中心点,在后处理筛选边界框时无需进行非极大值抑制,避免了旋转框之间交并比计算的引入;其次,设计具有角度和尺度自适应的高斯核函数,用于拟合畸变行人的中心分布,大幅减少了背景特征的干扰,并且平衡了具有不同成像大小的行人在边界框回归过程中的差异;最后,设计角度交并比损失,同时结合交并比损失以及旋转框参数的Ln范数损失,并通过指示函数改善角度正则项与交并比损失回归不一致的问题。在公开数据集上对算法进行了验证,实验结果表明,算法的平均mAP为51.33%,取得了目前最佳的检测结果,检测帧率达到49 fps,与带锚框的检测算法相比,提升了139%,综合性能优于现有的基于鱼眼图像的行人检测算法。

**关键词**:鱼眼图像;行人检测;无锚框;旋转高斯核;角度交并比
**中图分类号**:TP391.4　　**文献标识码**:A　　**doi**:10.3788/gzxb20215010.1010003

# A Person Detection Algorithm in Fisheye Images Based on Rotated Boxes

ZHANG Saiqiang[1,2], SI Shaofeng[1,2], LU Bin[1,2], LI Qing[2], CHEN Benyao[3], RONG Anxin[3]
(1 *Institute of Microelectronics of the Chinese Academy of Sciences, Beijing* 100029, *China*)
(2 *University of Chinese Academy of Sciences, Beijing* 100049, *China*)
(3 *Huzhou Special Equipment Inspection Center, Huzhou, Zhejiang* 313000, *China*)

**Abstract**: Due to the geometric distortion of fisheye images, the existing person detection algorithms based on fisheye images have the problems of low detection accuracy and high computational complexity in post-processing. A rotation-aware person detection algorithm was proposed to solve the problems. First, the algorithm adopted an anchor-free network structure and used heatmap to predict the center point of the bounding box, there was no need to apply non-maximum suppression on the bounding boxes during post-processing which avoids the calculation of intersection over union(IoU) between rotated bounding boxes. Then, a Gaussian kernel function with angle and scale adaptation was adopted to fit the center distribution of person with distortions, which greatly reduced the interference of background features, and balanced the difference of person with different sizes under fisheye images during the bounding boxes regression. Finally, the Angle-IoU(AIoU) was designed to combine both IoU loss and Ln-norm loss, indicator function was used to deal with inconsistent regression between IoU loss and Ln-norm of angle regular term. The proposed algorithm was verified on public datasets, experimental results show that the algorithm

**基金项目**:浙江省市场监督管理局质量技术基础建设项目(No.20200125)
**第一作者**:张赛强(1996—),男,硕士研究生,主要研究方向为鱼眼俯视视角下的目标检测。Email: zhangsaiqiang19@mails.ucas.ac.cn
**导师(通讯作者)**:李庆(1972—),男,研究员,博士,主要研究方向为人工智能、信息处理与多源信息融合。Email: liqing@ime.ac.cn
**收稿日期**:2021-06-30;**录用日期**:2021-08-18

has achieved the state-of-art performance with an average mAP of 51.33%, detection frame rate reaches 49 fps, which is 139% higher than the detection algorithm with anchor-based network structure, the comprehensive performance of the algorithm is better than other existing person detection algorithms in overhead fisheye images.

**Key words**: Fisheye images; Person detection; Anchor-free; Rotated Gaussian kernel; AIoU
**OCIS Codes**: 100.2000; 120.1880; 110.2970; 330.7326

## 0 引言

随着人工智能的发展,行人检测已经成为智能视频监控领域的一个重要研究课题[1]。相较于普通窄角相机,鱼眼相机通常被安装在应用场景顶部,往往具有更大的视场空间,同时俯视场景下行人间相互遮挡的情况也大大减少,因此在视频监控以及智能家居中被广泛应用。

鱼眼图像下传统的行人检测算法通常基于背景差方法提取变化区域,通过聚类确定行人位置[2-3],在检测结果中常常包含较多的非行人目标,为了准确地检测出行人,文献[4]引入支持向量机(Support Vector Machine,SVM)分类器对前景目标进行识别,但上述方法往往受环境和光照变化的影响[5],得到边界框的质量不佳,同时泛化能力较差。随着目标检测技术的发展,出现了深度学习目标检测算法,如YOLO(You Only Look Once)[6]、SSD(Single Shot Detector)[7]、Faster R-CNN(Faster Region Convolutional Neural Network)[8]以及CornerNet[9]等,不过这些算法针对的大多是直立姿态的行人,对鱼眼图像下的畸变行人检测效果不理想,文献[10]指出在鱼眼图像下直接使用YOLOv2进行行人检测时会丢失部分行人,存在着漏检问题。为了检测出具有畸变状态的行人,文献[1]将一张鱼眼图像进行旋转切割,得到36张子图,在子图上使用Faster R-CNN进行检测,并将子图上的检测边界框重新映射到原始鱼眼图像上,同样文献[11]使用YOLOv3在24张子图上进行检测,文献[12]则是基于一张鱼眼图像生成若干透视图,将这些透视图组合起来使用YOLO进行行人检测,尽管上述方法能够取得较好的检测效果,可是使用大量子图进行检测以及过于复杂的前后处理,导致检测效率较低。

为了解决检测效率较低的问题,有研究者直接针对原始鱼眼图像进行处理,同时在边界框中引入角度参数进行旋转框检测。文献[13]中提出对训练图像进行旋转增强,使模型具有旋转不变性,不过边界框的角度预测值由边界框中心点位置给出,准确性较差;RAPiD(Rotation-Aware People Detection)[14]中利用带锚框的网络进行角度值的回归,能够在鱼眼图像上取得出色的性能表现,相较于轴对齐边界框,该方法可以避免生成多张子图,减少前后处理上的计算消耗,但是带锚框的方法在使用非极大值抑制(Non-Maximum Suppression, NMS)筛选边界框时,引入了基于旋转框的交并比(Intersection over Union, IoU)的计算,增加了模型的推理时间,无法满足更高实时性的需求。此外,文献[15]中指出边界框参数的回归和基于IoU的评价指标之间并不存在较强的相关性,而RAPiD中仅仅依靠边界框参数损失进行回归存在着性能瓶颈,因此,在模型检测效果上依旧存在着提升空间。

针对目前鱼眼图像下行人检测算法存在的问题,本文设计了采用无锚框结构的旋转框行人检测器(Anchor-Free Rotation-aware Person Detector, AFRPD),无锚框结构避免了预设锚框引入的超参数设计,减小了计算复杂度;将真实框的角度信息引入到高斯核函数中,得到带有角度旋转的高斯核函数,更加准确地拟合了具有几何畸变的行人的分布;在鱼眼图像的边缘位置,由于行人成像较小,造成边界框的回归质量较差,为了提升图像边缘小目标的回归能力,设计具有尺度自适应的高斯核函数;基于距离交并比(Distance Intersection over Union, DIoU)[16]的思考,提出了角度交并比(Angle Intersection over Union, AIoU)损失用于边界框的回归,同时使用IoU损失以及旋转框参数的Ln范数损失,能够提升边界框的回归效果。最后本文所提出的行人检测算法在预测中心热图上使用2D最大池化确定预测框,代替了NMS,从而在模型推理过程中完全避免了基于旋转框的IoU的计算,在实时性上有了较大提升。

## 1 基于旋转框的行人检测算法

实际应用中往往对算法的实时性有较高的要求,因此,本文直接在原始的鱼眼图像上进行检测,而现有的检测算法[14]在NMS操作上需要进行大量的基于旋转框的IoU的计算,导致推理效率不高。从替换NMS

的角度出发，模型框架基于TTFNet(Training-Time-Friendly Network)[17]，如图1所示，包含进行特征提取的骨干网络，进行特征融合的金字塔网络(Feature Pyramid Network，FPN)以及预测中心点位置和边界框参数的检测网络。骨干网络采用残差结构提升信息前后向传播的流畅度，同时输出4个层次的特征图，对应的下采样步长由$s_k$给出；在特征金字塔网络中采用自顶向下以及横向连接的方式，使得输出特征图能够结合高层特征的语义信息以及低层特征的高分辨率信息。不同于TTFNet算法仅仅局限于轴对齐边界框的回归，本文对旋转边界框的回归进行了探索，提出了适用于鱼眼相机场景下的AFRPD算法，其中心点位置预测采取和CenterNet[18]相同的方式，利用带有角度旋转的高斯核函数生成行人分布的热图，使网络在中心点位置有更高的激活输出，通过2D最大池化筛选出局部极大值的位置，从而确定边界框；当进行边界框回归时，将高斯核函数值非零位置对应的特征点看作边界框的回归样本，通过角度信息的引入减少了背景特征点的干扰；鱼眼图像中不同位置的行人，利用高斯核函数计算出的分布差异较大，导致在回归样本数量上存在着不均衡的问题，针对该问题，设计具有尺度自适应的高斯核函数；此外，提出AIoU损失用于提升预测边界框的回归效果。

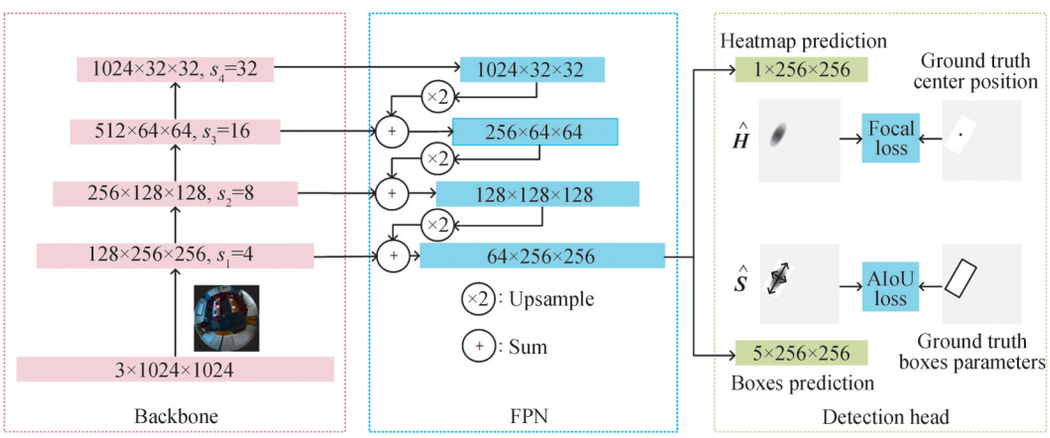



图1　鱼眼图像下的无锚框行人检测网络模型

Fig.1　Anchor-free person detection network under fisheye images

对于鱼眼图像$I$，检测网络输出为$\hat{\boldsymbol{H}}\in R^{N\times C\times\frac{H_I}{r}\times\frac{W_I}{r}}$和$\hat{\boldsymbol{S}}\in R^{N\times 5\times\frac{H_I}{r}\times\frac{W_I}{r}}$，其中，$\hat{\boldsymbol{H}}$为预测热图，包含中心点位置信息，$\hat{\boldsymbol{S}}$为预测的边界框参数，在特征图$\hat{\boldsymbol{S}}$上的每一个特征点位置对应着一个预测边界框，包括边界框的尺度和角度信息，$N$、$C$、$H_I$、$W_I$、$r$分别表示批大小、预测类别数、输入图像的高和宽以及下采样步长，并在实验中设置$C=1$，$r=4$，在接下来的表述中忽略掉$N$。图像$I$中第$m$个真实边界框用$B_m^{\mathrm{gt}}$表示，$B_m^{\mathrm{gt}}=(x_m^{\mathrm{gt}},y_m^{\mathrm{gt}},w_m^{\mathrm{gt}},h_m^{\mathrm{gt}},\theta_m^{\mathrm{gt}})$，$(x_m^{\mathrm{gt}},y_m^{\mathrm{gt}})$为中心点坐标，规定$w_m^{\mathrm{gt}}<h_m^{\mathrm{gt}}$，$\theta_m^{\mathrm{gt}}$为边界框长轴$h_m^{\mathrm{gt}}$和图像竖直方向的夹角，顺时针为正，取值范围为$[-90^\circ,90^\circ)$。

## 1.1　改进的高斯核函数

### 1.1.1　旋转高斯核函数的引入

在目标检测任务中，常常需要在模型预测的特征图$\hat{\boldsymbol{S}}$上选取正样本进行边界框的回归，而不同的研究者采取了不同的策略。FCOS(Fully Convolutional One-Stage object detector)[19]将边界框中的所有特征点视作回归样本，如图2(a)所示；FoveaBox[20]中将阴影矩形区域对应的特征点视作回归样本，如图2(b)所示；CenterNet[18]中仅将边界框中心点视作回归样本，如图2中(c)所示；TTFNet中改进(c)方法，将高斯区域对应的特征点作为回归样本，如图2中(d)所示，通过边界框尺寸以及高斯概率值对回归损失进行加权，能够更充分地利用边界框中的特征点进行边界框回归。上述(b)～(d)方法的改动都基于一个假设：在进行旋转框回归时，边界框区域中的点并不都是可信任的，即并不是所有的点都有利于边界框回归。因此，如何选取有效的回归样本成为本文需要解决的一个问题。

受TTFNet基于轴对齐边界框选取高斯区域的启发，本文针对旋转边界框引入角度自适应对高斯核函数进行扩展。由于旋转框中行人中心点位置并未发生改变，一种做法是在旋转框中使用轴对齐的高斯核函数，如图2(e)所示，然而边界框中除了行人对应的特征外，还有大量的背景特征，直接使用轴对齐的高斯核函数会导致非行人部分的特征点用于边界框的回归，这种现象随着高斯核方差增加逐渐显著。因此，本文

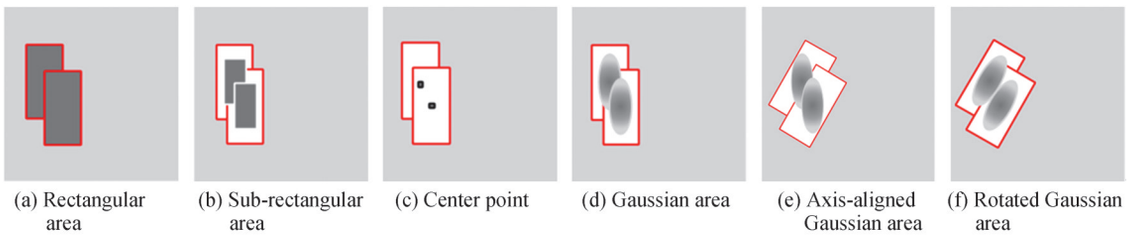



图 2 用作边界框回归样本的不同策略

Fig.2 Different strategies used as bounding box regression samples

在上述假设的基础上进一步假定：边界框中和行人对应的特征点更值得信赖，而和背景对应的点可信度较低。因此，本文为高斯核函数增加角度先验，如图 2 中的(f)所示，将旋转高斯区域中的特征点看作回归样本，通过角度信息的引入，使选取的回归样本区域能够和行人区域更加吻合，从而实现更有效的回归。检测网络在计算中心点位置损失和边界框回归损失时同时使用旋转高斯核函数，并通过超参数调整高斯核的大小。

### 1.1.2 旋转高斯核函数的计算

直接计算带有角度旋转的高斯核函数比较困难，本文采取投影方式来获取旋转高斯核函数的取值，如图 3 所示。边界框 $B_m^{\mathrm{gt}}$ 中一点 $(x_m, y_m)$，沿角度 $\theta_m^{\mathrm{gt}}$ 在 $w_m^{\mathrm{gt}}$、$h_m^{\mathrm{gt}}$ 上投影分别计作 $x_m^{\mathrm{projection}}$、$y_m^{\mathrm{projection}}$，$l$ 为 $(x_m, y_m)$ 到边界框中心点 $(x_m^{\mathrm{gt}}, y_m^{\mathrm{gt}})$ 的距离。旋转高斯核 $K_m^{\mathrm{rotate}}(x, y)$ 在 $(x_m, y_m)$ 位置的取值可以使用轴对齐高斯核 $K_m^{\mathrm{axis-align}}(x, y)$ 在 $(x_m^{\mathrm{projection}}, y_m^{\mathrm{projection}})$ 位置的取值代替，如式(1)所示。$\sigma_x = \alpha w_m^{\mathrm{gt}}/6$、$\sigma_y = \alpha h_m^{\mathrm{gt}}/6$ 分别表示在 $x$ 和 $y$ 方向上的标准差，与边界框的长宽相关，在计算中心点位置对应的高斯核时，设置 $\alpha = 0.54$。边界框 $B_m^{\mathrm{gt}}$ 对应的高斯区域记作 $\boldsymbol{H}_m \in R^{1\times\frac{H_I}{r}\times\frac{W_I}{r}}$，取值由 $K_m^{\mathrm{rotate}}(x_m, y_m)$ 给出，对于图像 $I$ 中剩余的边界框，利用逐元素最大化的方式进行更新，得到所有行人的分布 $\boldsymbol{H} \in R^{1\times\frac{H_I}{r}\times\frac{W_I}{r}}$，将其用于中心点位置损失的计算中。

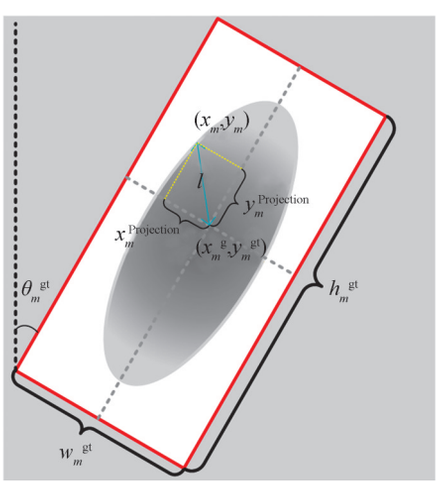



图 3 计算旋转高斯核的投影方法

Fig.3 Projection method for calculating rotated Gaussian kernel

$$\begin{cases} K_m^{\mathrm{rotate}}(x_m, y_m) = K_m^{\mathrm{axis-align}}(x_m^{\mathrm{projection}}, y_m^{\mathrm{projection}}) = \exp\left[-\dfrac{(x_m^{\mathrm{projection}})^2}{2\sigma_x^2} - \dfrac{(y_m^{\mathrm{projection}})^2}{2\sigma_y^2}\right] \\ x_m^{\mathrm{projection}} = l\cos\left[\arctan\left(\dfrac{y_m - y_m^{\mathrm{gt}}}{x_m - x_m^{\mathrm{gt}}}\right) - \theta_m^{\mathrm{gt}}\right] \\ y_m^{\mathrm{projection}} = l\sin\left[\arctan\left(\dfrac{y_m - y_m^{\mathrm{gt}}}{x_m - x_m^{\mathrm{gt}}}\right) - \theta_m^{\mathrm{gt}}\right] \\ l = \sqrt{(y_m - y_m^{\mathrm{gt}})^2 + (x_m - x_m^{\mathrm{gt}})^2} \end{cases} \tag{1}$$

#### 1.1.3 尺度自适应高斯核函数的引入

由于鱼眼图像径向畸变的存在，观察鱼眼图像下的行人边界框，可以发现行人边界框的大小随其中心点到鱼眼图像中心距离增加而减小。就行人检测任务而言，对于鱼眼图像半径较大位置的行人和半径较小位置的应当同等看待；然而如图4中(a)所示，可以看到在回归样本选取的过程中，半径较大位置的行人由于其较小的尺寸，使用旋转高斯核选取的回归样本在数量上相比于大目标要少很多，大小目标在回归样本数据上的差异很容易造成回归边界框质量上的差异，为了解决大小目标在回归损失上的差异，TTFNet中对损失权重进行调整，不过回归样本在数量上的差异问题依旧存在。

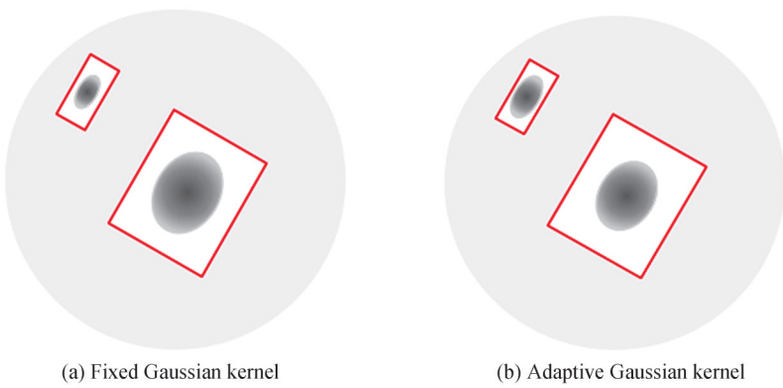



图4 固定高斯核和自适应高斯核对比

Fig.4 Comparison between fixed Gaussian kernel and adaptive Gaussian kernel

本文基于鱼眼图像的先验信息，在引入TTFNet损失权重的同时，使用尺度自适应的高斯核函数来调整不同位置目标的回归样本数量，当计算边界框回归部分的高斯核函数时，不再使用固定的$\alpha$，而是引入随边界框位置改变而变化的参数$\alpha^{\text{adaptive}}$，用于调节高斯核函数的大小，如图4(b)所示，随着行人边界框中心点到鱼眼图像中心距离的增加，增大$\alpha^{\text{adaptive}}$的值，为图像边缘位置的小目标引入更多的回归样本，从而提升小目标的边界框回归效果。

$\alpha^{\text{adaptive}}$的表现形式由式(2)给出，其中$\alpha^{\min}$和$\lambda$为预先设定的常数，$H_I$, $W_I$分别为图像$I$对应的长和宽。

$$\alpha^{\text{adaptive}}=\alpha^{\min}+\text{sigmoid}\left[\lambda\cdot\frac{\sqrt{(x_m^{\text{gt}}-W_I/2)^2+(y_m^{\text{gt}}-H_I/2)^2}}{\max(H_I,W_I)/2}\right] \tag{2}$$

### 1.2 中心点位置预测损失

给定图像$I$，网络输出中心点位置预测值$\hat{\boldsymbol{H}}\in R^{1\times\frac{H}{r}\times\frac{W}{r}}$，对应的真实值$\boldsymbol{H}\in R^{1\times\frac{H}{r}\times\frac{W}{r}}$可以利用旋转高斯核函数计算得出，网络需要确定边界框中心点的位置，因此，仅仅将中心点位置看作正样本，也就是高斯概率值为1的样本点，其余样本为负样本，对应的损失值$L_{\text{loc}}$为

$$L_{\text{loc}}=\frac{1}{M_{\text{positive}}}\sum_{ij}\begin{cases}(1-\hat{\boldsymbol{H}}^{ij})^{\alpha_{\text{f}}}\log(\hat{\boldsymbol{H}}^{ij}) & \boldsymbol{H}^{ij}=1\\ (1-\boldsymbol{H}^{ij})^{\beta_{\text{f}}}(\hat{\boldsymbol{H}}^{ij})^{\alpha_{\text{f}}}\log(1-\hat{\boldsymbol{H}}^{ij}) & \boldsymbol{H}^{ij}\neq 1\end{cases} \tag{3}$$

式中，$\alpha_{\text{f}}$，$\beta_{\text{f}}$为焦点损失中的超参数，在实验设置$\alpha_{\text{f}}=2$，$\beta_{\text{f}}=4$，$M_{\text{positive}}$为图像$I$中包含的正样本数量。

### 1.3 旋转框的回归

#### 1.3.1 回归损失的计算

当进行边界框回归时，对于图像$I$中的第$m$个真实框，可以计算其对应的旋转高斯核$\boldsymbol{G}_m\in R^{1\times\frac{H_I}{r}\times\frac{W_I}{r}}$，将非零区域记作$A_m$，表示回归到第$m$个真实框的所有样本；同样对图像$I$中所有真实框对应的高斯核逐元素取最大值，得到回归部分的高斯核$\boldsymbol{G}\in R^{1\times\frac{H_I}{r}\times\frac{W_I}{r}}$，其非零区域记作$A$。

$A_m$中的任意一点$(i,j)$，网络输出步长为$r$，回归预测对应到图像$I$的第$m$个真实框上，在该点的回归预测输出定义为$(i\cdot r,j\cdot r)$沿着边界框$B_m^{\text{gt}}$短边$w_m^{\text{gt}}$和长边$h_m^{\text{gt}}$方向到边界框4个边的距离的$1/s$以及角度值，表示为$\hat{\boldsymbol{S}}_m^{ij}=(\hat{w}_{\text{l}},\hat{h}_{\text{t}},\hat{w}_{\text{r}},\hat{h}_{\text{b}},\hat{\theta})_m^{ij}$，设置$s=16$来增大回归预测结果，为了计算IoU损失，利用式(4)对网络回归

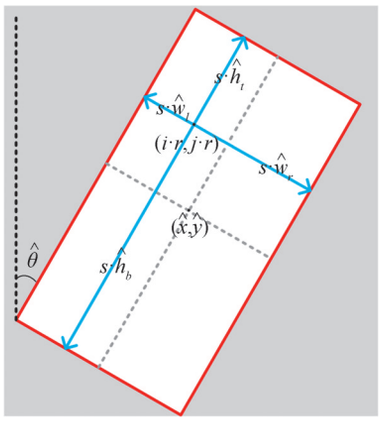



图5 变换网络回归输出到预测边界框

Fig.5 Transformation from network regression outputs to predicted bounding boxes

输出$(\hat{w}_l,\hat{h}_t,\hat{w}_r,\hat{h}_b,\hat{\theta})_m^{ij}$进行变换，得到的预测边界框为$\hat{B}_m^{ij}=(\hat{x},\hat{y},\hat{w},\hat{h},\hat{\theta})_m^{ij}$，对应图5中的红色矩形框，同理可以计算非零高斯区域$A$中包含特征点对应的预测边界框，记作$\hat{B}$，$\hat{B}\in R^{N_{reg}\times 5}$，其中$N_{reg}$表示图像$I$中的回归样本总数；预测边界框对应的真实值为$B\in R^{N_{reg}\times 5}$，输入单张图片的回归损失函数$L_{reg}$由式(5)给出，$W_m^{ij}$在TTFNet[17]中给出，表示特征图$(i,j)$位置回归到第$m$个边界框时对应的损失权重，和边界框的面积$a_m$以及$(i,j)$位置的高斯概率值$\boldsymbol{G}_m(i,j)$相关，使用基于IoU的损失函数对预测边界框参数$\hat{B}_m^{ij}$进行优化，并将其记作regloss$(\hat{B}_m^{ij},B_m^{gt})$，接着进行加权得到最终的$L_{reg}$。

$$\begin{cases}\hat{x}=i\cdot r+\sin\hat{\theta}s\cdot\hat{w}_r-s\cdot\hat{w}_l)/2+\cos(\hat{\theta})(s\cdot\hat{h}_b-s\cdot\hat{h}_t)/2\\ \hat{y}=j\cdot r-\sin(\hat{\theta})(s\cdot\hat{h}_b-s\cdot\hat{h}_t)/2+\cos(\hat{\theta})(s\cdot\hat{w}_r-s\cdot\hat{w}_l)/2\\ \hat{w}=s\cdot\hat{w}_r+s\cdot\hat{w}_l\\ \hat{h}=s\cdot\hat{h}_b+s\cdot\hat{h}_t\\ \hat{\theta}=\hat{\theta}\end{cases}\tag{4}$$

$$L_{reg}=\frac{1}{N_{reg}}\sum_{A_m\in A}\sum_{(i,j)\in A_m}\text{regloss}(\hat{B}_m^{ij},B_m^{gt})\times W_m^{ij}\tag{5}$$

$$W_m^{ij}=\begin{cases}\log(a_m)\times\dfrac{\boldsymbol{G}_m(i,j)}{\sum_{(x,y)\in A_m}\boldsymbol{G}_m(x,y)} & (i,j)\in A_m\\ 0 & (i,j)\notin A_m\end{cases}\tag{6}$$

### 1.3.2 AIoU损失的表现形式

目前研究者提出的基于IoU的损失函数包括通用交并比(Generalized Intersection over Union，GIoU)[15]，DIoU[16]，完整交并比(Complete Intersection over Union，CIoU)[16]以及高效交并比(Efficient Intersection over Union，EIoU)[21]，这些损失都是在IoU loss[22]的基础上增加正则项，以提升预测边界框的质量，不过上述损失都是基于轴对齐的边界框，当被应用到旋转框时，使用像素交并比(Pixels Intersection over Union，PIoU)[23]近似计算$\hat{B}_m^{ij}$，$B_m^{gt}$之间的IoU，同时将两旋转框对应的最小闭包近似为两者的最小外接轴对齐矩形，如图6所示，记作AAR$(\hat{B}_m^{ij},B_m^{gt})$。

当进行旋转框回归时，除了中心点以及长宽正则项外，由于角度参数的存在，一个很直接的想法就是将角度信息作为正则项引入到损失计算中，而角度参数的引入需要解决两个问题：角度的周期性问题以及角度正则项同PIoU的回归一致性问题。本文采用周期性的角度损失处理第一个问题，然而在实验过程中发现单纯地引入周期性损失无法取得更好的结果，这是由于角度的回归同PIoU回归存在着不一致的问题，如图7所示，假定$\hat{B}_m^{ij}$，$B_m^{gt}$取相同的中心点坐标、长以及宽，观察不同长宽比下角度差变化对两者IoU的影响，图中横坐标表示为$\hat{B}_m^{ij}$和$B_m^{gt}$之间的角度差，纵坐标表示为$\hat{B}_m^{ij}$和$B_m^{gt}$之间的IoU。本文引入的周期性角度正则项由(8)式中$L_{angle}$给出，最小化$L_{angle}$会使得$\hat{B}_m^{ij}$和$B_m^{gt}$角度差$\Phi_{diff}$向0和π方向收敛，并且收敛方向不受$h_m^{gt}/w_m^{gt}$值

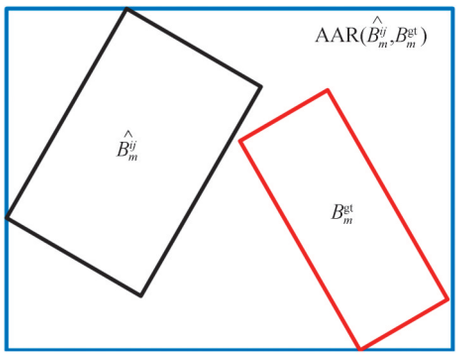



图6　两个旋转框的最小外接轴对齐矩形

Fig.6　Minimum axis-aligned rectangle of two rotated bounding boxes

的影响，但从图中可以看出，当$h_m^{gt}/w_m^{gt}$取值较小时，这里取$h_m^{gt}/w_m^{gt}=1.0$进行分析，PIoU损失最小化对应着角度差$\Phi_{diff}$的收敛目标有3个：0，π/2以及π，此时在回归过程中会出现角度正则项损失减小而PIoU损失增大的情况，导致增加角度正则项后边界框的回归效果变差。因此，本文在损失函数中进一步引入指示函数$\chi$以及超参数$\beta$来改善第二个问题，仅在$h_m^{gt}/w_m^{gt}>\beta$时加入角度正则项，避免不一致情况出现的同时为回归过程提供额外的先验信息。基于上述分析，本文加入角度正则项得到损失函数的表现形式如式(7)所示，记作AIoU。

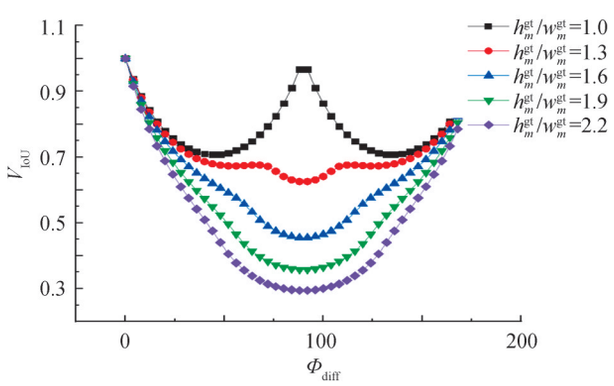



图7　不同长宽比下IoU与角度差之间的关系

Fig.7　Relationship between IoU and angle difference under different aspect ratios

$$\begin{aligned}\text{regloss}(\hat{B}_m^{ij},B_m^{gt})&=\text{AIoU}(\hat{B}_m^{ij},B_m^{gt})=1-\text{PIoU}(\hat{B}_m^{ij},B_m^{gt})+[\rho^2(\hat{x},x_m^{gt})+\rho^2(\hat{y},y_m^{gt})+\rho^2(\hat{w},w_m^{gt})+\\&\rho^2(\hat{h},h_m^{gt})]/c^2+\chi L_{angle}\end{aligned}\tag{7}$$

$$\begin{cases}\chi=\begin{cases}1,\beta w_m^{gt}<h_m^{gt}\\0,\beta w_m^{gt}\geqslant h_m^{gt}\end{cases}\\D_{angle}=\left[\text{mod}(\hat{\theta}-\theta_m^{gt}-\dfrac{\pi}{2},\pi)-\dfrac{\pi}{2}\right]\Big/\dfrac{\pi}{2}\\L_{angle}=f(D_{angle}/2)\end{cases}\tag{8}$$

式中，$\sqrt{\rho^2(\hat{x},x_m^{gt})+\rho^2(\hat{y},y_m^{gt})}$表示$\hat{B}_m^{ij}$，$B_m^{gt}$中心点的L2距离，$\rho(\hat{w},w_m^{gt})$表示$\hat{B}_m^{ij}$、$B_m^{gt}$宽的L2距离，$\rho(\hat{h},h_m^{gt})$表示$\hat{B}_m^{ij}$、$B_m^{gt}$高的L2距离，$c$表示$\hat{B}_m^{ij}$、$B_m^{gt}$最小外接轴对齐矩形AAR($\hat{B}_m^{ij}$,$B_m^{gt}$)的对角线长度。$D_{angle}$为$\hat{B}_m^{ij}$和$B_m^{gt}$的归一化周期性角度距离，为了平衡损失函数中各个正则项之间的差异，采用$D_{angle}/2$以减小角度损失，$f(\cdot)$在实验中取为$(\cdot)^2$。

### 1.4　总损失的计算

总损失$L=w_{loc}L_{loc}+w_{reg}L_{reg}$，包括中心点的位置损失和边界框的回归损失两部分，实验中设置$w_{loc}=1$，$w_{reg}=5$。

## 2 实验结果及分析

### 2.1 实验数据集与网络参数设定

在鱼眼图像实验中，首先使用MS COCO 2017数据集进行预训练，接着采取交叉验证的方式在公开鱼眼图像数据集下进行微调，即从MW-R数据集[14]、HABBOF数据集[11]以及CEPDOF数据集[14]中选择两个作为训练集，剩余数据集用作测试。CEPDOF数据集中包含三组低光照数据，当用于训练时，仅采用红外场景下的一组低光照数据（IRill）；当用于测试时，使用全部数据。

本文实验平台为Ubuntu18.04操作系统，使用NVIDIA RTX 2080 Ti加速训练，软件环境为torch1.6.0+torchvision0.7.0。模型骨干网络采用ResNet和Darknet，在预训练阶段，对COCO train2017图像进行随机旋转，为真实边界框引入角度参数。使用随机梯度下降法进行网络参数的更新，训练参数并没有进行择优，而是采取和TTFNet相同设置。对于Darknet，基础学习率为0.001 875，动量为0.9，权重衰减为0.000 4；对于ResNet，基础学习率为0.002，动量为0.9，权重衰减为0.000 4，在第18和22个epoch学习率分别衰减10倍。在前500步中采取预热策略，训练阶段采用了随机上下翻转、随机旋转等数据增强方式。实验采用目标检测中常用的平均类别精度（mean Average Precision，mAP）、帧率（Frames Per Second，FPS）以及F1分数进行评估。

### 2.2 对比实验

为了验证本文引入的旋转高斯核函数、AIoU以及自适应高斯核函数，选取不同的IoU损失函数进行对比实验，模型将ResNet18用作骨干网络，使用的图片大小为512×512。

#### 2.2.1 高斯核函数对比实验

在MW-R、HABBOF以及CEPDOF数据集上，本文采取交叉验证的方式，加载ImageNet上的预训练模型，从头开始训练24个epoch，由于PIoU在训练初期，边界框损失很难下降，所以使用CIoU作为损失函数。分别采用轴对齐高斯核函数以及旋转高斯核函数进行实验，结果如表1所示。从交叉验证的实验结果上来看，使用旋转高斯核函数在HABBOF以及CEPDOF上能够取得更好的结果，而在MW-R数据集上性能和轴对齐高斯核函数相当。因此，在回归样本中真实目标对应的特征点相对于背景点来说更值得关注。

**表1 高斯核函数对比实验**

**Table 1 Comparative experiment of Gaussian kernel function**

| Gaussian kernel | MW-R/% | | HABBOF/% | | CEPDOF/% | |
|---|---|---|---|---|---|---|
| | mAP | $AP_{75}$ | mAP | $AP_{75}$ | mAP | $AP_{75}$ |
| Axis-aligned Gaussian kernel | 31.9 | 23.3 | 40.5 | 35.2 | 17.1 | 6.4 |
| Rotated Gaussian kernel | 31.7 | **24.4** | **42.7** | **37.2** | **18.3** | **8.9** |

#### 2.2.2 AIoU对比实验

该部分对本文所设计的AIoU损失与现有的其他IoU的损失函数进行对比。首先在COCO数据集上预训练12个epoch，接着在鱼眼数据集上训练12个epoch，得到结果如表2所示。对比PIoU，EIoU以及AIoU，可以发现增加边界框参数作为正则项能够为回归过程提供更多信息，从而更有利于边界框的回归。对比CIoU和EIoU，可以发现两者均引入了长宽比信息，而CIoU的回归结果相比于PIoU要差，这是因为在真实框$B_m^{\mathrm{gt}}$中规定$w_m^{\mathrm{gt}}<h_m^{\mathrm{gt}}$，而模型预测阶段会出现$\hat{w}\geqslant\hat{h}$的情况，CIoU中的长宽比正则项会使得长宽的收敛方向无法确定，增大$\hat{h}$或者减小$\hat{w}$均可以减小损失，存在着同PIoU损失收敛不一致的可能，从而导致回归结果较差。同样GIoU中最小化边界框轴对齐外接矩形AAR($\hat{B}_m^{ij}$, $B_m^{\mathrm{gt}}$)与边界框并集($\hat{B}_m^{ij}\cup B_m^{\mathrm{gt}}$)的比值，应用到旋转框回归时，由于角度参数的引入无法给出明确的回归方向。而EIoU中直接对长宽值进行回归，明确给出回归方向，从而得到较好的结果。增加角度信息得到的AIoU在上述损失中取得了最佳的表现结果，对比EIoU的结果，AIoU引入了角度值的正则项，使得中心点和长宽预测较为准确的边界框能够旋转到和真实框重叠更大的方向上，从而在mAP和$AP_{75}$上取得提升。

在表3中对AIoU进行了超参数的测试实验。当使用较小的$\beta$时，角度正则项与PIoU在收敛方向上的不一致导致性能较差；随着$\beta$的增加，收敛不一致的情况逐渐得到改善，同时角度正则项的引入为边界框回

归提供了更多的信息，此时模型的性能表现会有提升；但当$\beta$进一步增大时，指示函数$\chi$值为1的情况大大减少，大量边界框的角度正则项$\chi \times L_{angle}$计算结果为0，无法在回归过程中结合角度信息，导致模型性能有所下降。

**表2　IoU损失对比实验**

**Table 2　Comparative experiment of IoU loss**

| Loss type | MW-R/% | | HABBOF/% | | CEPDOF/% | |
|---|---|---|---|---|---|---|
| | mAP | $AP_{75}$ | mAP | $AP_{75}$ | mAP | $AP_{75}$ |
| PIoU | 43.6 | 33.9 | 49.2 | 47.9 | 27.8 | 14.8 |
| GIoU | 27.3 | 13.2 | 15.2 | 6.0 | 9.3 | 1.7 |
| CIoU | 42.3 | 31.7 | 49.5 | 46.8 | 26.2 | 14.8 |
| EIoU | 45.4 | 37.2 | 49.3 | 48.1 | 29.2 | 16.1 |
| AIoU | **46.4** | **40.3** | **50.0** | **49.4** | **29.6** | **17.0** |

**表3　超参数测试实验**

**Table 3　Ablation experiment of hyperparameter**

| $\beta$ | MW-R/% | | |
|---|---|---|---|
| | mAP | $AP_{50}$ | $AP_{75}$ |
| 1.0 | 45.4 | 90.1 | 37.9 |
| 1.3 | 45.2 | 90.3 | 36.3 |
| 1.6 | 45.5 | 90.2 | 38.5 |
| 1.9 | **46.4** | **90.9** | **40.3** |
| 2.2 | 46.4 | 90.4 | 39.6 |
| 2.5 | 45.8 | 90.6 | 37.3 |

#### 2.2.3　自适应高斯核函数对比实验

在3个公开鱼眼图像数据集上对本文提出的自适应高斯核函数进行验证。首先在COCO数据集上预训练12个epoch，接着引入自适应高斯核函数在鱼眼数据集上训练12个epoch，实验结果分别使用PIoU_A、CIoU_A、EIoU_A以及AIoU_A表示，其中$mAP_s$为小目标的检测结果，从表4中可以看出自适应高斯核函数的引入在提升模型整体检测性能中起到一定的效果，同时能够提升小目标的回归效果。

**表4　自适应高斯核函数对比实验**

**Table 4　Comparative experiment of Gaussian kernel function**

| Loss type | MW-R/% | | | HABBOF/% | | | CEPDOF/% | | |
|---|---|---|---|---|---|---|---|---|---|
| | mAP | $AP_{75}$ | $mAP_s$ | mAP | $AP_{75}$ | $mAP_s$ | mAP | $AP_{75}$ | $mAP_s$ |
| PIoU | 43.6 | 33.9 | 34.3 | 49.2 | 47.9 | 54.7 | 27.8 | 14.8 | 18.3 |
| PIoU_A | **44.3** | **35.3** | **34.9** | **49.7** | **48.3** | **57.2** | **28.1** | **15.5** | **19.6** |
| CIoU | 42.3 | 31.7 | 33.0 | 49.5 | 46.8 | 53.4 | 26.2 | 14.8 | 15.9 |
| CIoU_A | **42.9** | **32.2** | **33.4** | **49.6** | **47.8** | **53.8** | **29.0** | **15.7** | **19.8** |
| EIoU | 45.4 | 37.2 | 33.4 | 49.3 | 48.1 | 53.7 | 29.2 | 16.1 | 19.0 |
| EIoU_A | **46.2** | **38.4** | **34.7** | **49.8** | **50.3** | **55.3** | **30.1** | **18.4** | **19.1** |
| AIoU | 46.4 | 40.3 | 36.7 | 50.0 | 49.4 | 55.8 | 29.6 | 17.0 | 19.7 |
| AIoU_A | **47.2** | **40.6** | 35.9 | 49.6 | **50.0** | 54.1 | **30.1** | **18.0** | 19.6 |

### 2.3　和现有算法效果的对比

在本文中，将RAPiD算法用于对比实验，使用官方提供的模型参数以及代码，在本地进行推理，得到RAPiD方法对应的实验结果。为了验证本文模型在训练收敛速度上的优势，使用官方代码。首先在COCO数据集上训练24个epoch，接着使用鱼眼图像训练24个epoch，得到了表5中RAPiD_2x对应的实验结果，基于RAPiD算法的实验中骨干网络均采用Darknet53；在AFRPD实验中，骨干网络采用Resnet34以及

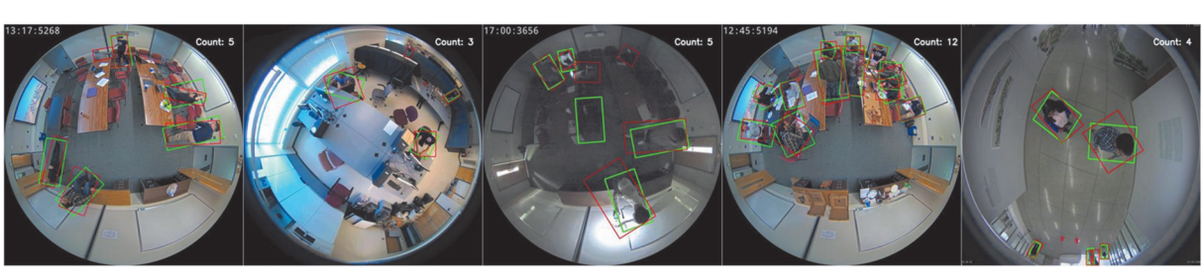
Darknet53，对应网络分别记作AFRPD-34/53。首先在COCO上预训练24个epoch，图像大小为608；接着在不同大小的鱼眼图像上训练24个epoch，得到最终结果。在计算AP参数时，IoU阈值为0.01；计算$F$参数时，IoU阈值为0.3。FPS的结果基于CEPDOF数据集以及NVIDIA RTX 2080 Ti。

**表5 与最先进算法检测结果的比较**

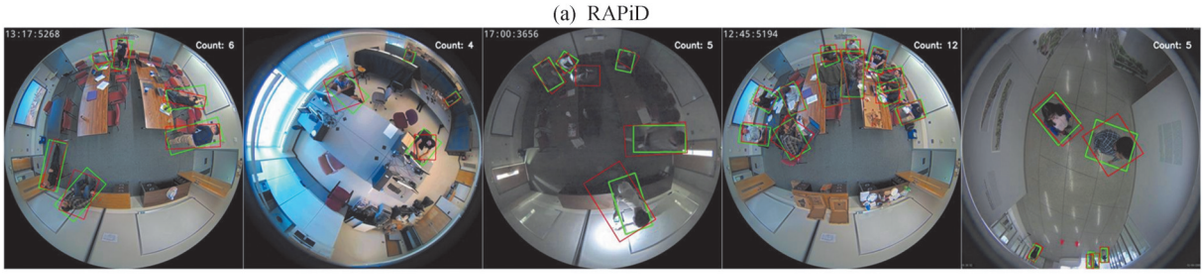
**Table 5 Comparison of the state-of-the-art algorithm and proposed algorithm**

| Method | Size | FPS | MW-R/% | | | | HABBOF/% | | | | CEPDOF/% | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | mAP | $AP_{50}$ | $AP_{75}$ | $F$ | mAP | $AP_{50}$ | $AP_{75}$ | $F$ | mAP | $AP_{50}$ | $AP_{75}$ | $F$ |
| RAPiD | 608 | 20.5 | 53.4 | 96.8 | 50.9 | 94.1 | 56.7 | 96.7 | 62.6 | 95.8 | 38.2 | 83.1 | 27.6 | 79.3 |
| RAPiD | 1 024 | 13.6 | 53.3 | 96.7 | 48.8 | 93.5 | 57.3 | 97.8 | 59.1 | 96.9 | 39.4 | 86.4 | 26.7 | 83.6 |
| RAPiD_2x | 608 | 17.4 | 42.5 | 93.2 | 26.3 | 89.3 | 43.7 | 90.7 | 31.2 | 88.5 | 29.3 | 78.3 | 11.6 | 78.0 |
| AFRPD-34 | 608 | 78.5 | 53.3 | 95.9 | 53.0 | 93.8 | 54.1 | 95.9 | 56.7 | 93.2 | 36.6 | 84.2 | 23.7 | 81.8 |
| AFRPD-34 | 1 024 | 39.0 | 55.5 | 97.9 | 55.9 | 96.2 | 59.4 | 97.2 | 63.5 | 94.4 | 40.1 | 86.1 | 29.2 | 84.5 |
| AFRPD-53 | 608 | 49.0 | **55.3** | **97.0** | **57.4** | **94.8** | **59.1** | **97.6** | **65.7** | **97.3** | **39.6** | **87.5** | **28.3** | **85.5** |
| AFRPD-53 | 1 024 | 23.0 | **56.4** | **98.1** | **60.4** | **95.4** | **59.0** | 97.7 | **66.3** | **96.9** | **40.6** | **87.4** | **29.5** | **85.2** |

从表5中RAPiD模型和RAPiD_2x模型的实验结果可以发现，相同图像尺寸下，RAPiD_2x模型的推理速度要差于RAPiD模型。实际上RAPiD算法的推理效率受制于模型性能和被检测图像的属性，较差的预测模型以及图像中较多的目标，会导致NMS阶段进行基于旋转框的IoU计算时花费更多的时间。而本文中提出的算法利用最大池化代替了NMS，从而解决了上述问题。对比表5中的实验结果，本文算法在检测性能上能够达到甚至优于最先进的检测算法，在MW-R、HABBOF以及CEPDOF数据集上，AFRPD-53模型的mAP均达到最佳，并且在$AP_{75}$上的表现得到大幅提升，这主要是得益于旋转高斯核函数以及AIoU的引入，使得算法能够更加准确地确定行人的位置。在综合考虑精度和召回率的情况时，本文提出的算法取得了最佳的表现；此外，图像尺寸为608和1 024时，模型推理速度提升分别为139%和69%；同时算法具有更快的收敛速度。图8中给出了RAPiD模型和AFRPD模型在不同场景下的定性检测结果，图中使用绿色框表示预测值，红色框表示真实值，从检测结果中可以看出，相比于RAPiD模型，本文提出的AFRPD模型在一定程度上能够避免错检，减少漏检，对于拥挤小目标也具有较好的检测能力。




(a) RAPiD




(b) AFRPD

图8 定性检测结果比较

Fig.8 Comparison of qualitative results

## 3 结论

本文将无锚框结构应用到旋转框行人检测中，引入中心热图确定边界框中心点，避免了基于旋转框的

IoU计算的引入,使得模型在实时性上有了较大的提升;通过矩阵操作引入带有角度旋转的高斯核函数,能够减少背景样本的干扰,帮助模型选取到更加有效的回归样本;设计具有尺度自适应的高斯核函数调整不同位置目标在回归样本数量上的差异,提升小目标的检测效果。与此同时,本文探究了各种IoU损失应用到基于旋转框的行人检测上的性能表现,在此基础上本文提出的AIoU能够取得更好的结果。本文提出的算法在相同任务上的检测性能优于当前最先进的检测算法,同时在推理速度上有了较大提升,更快的推理速度使得其在移动设备或者嵌入式端上同样具备优势。

目标检测任务的一个趋势是在预测类别分数中结合边界框的质量信息,希望在后续工作中能够结合网络预测中心热图和边界框回归的质量信息在基于鱼眼图像的行人检测任务上取得更好的检测效果。

## 参考文献


[1] LIN Hongli L, KONG Zhenzhen, WANG Weisheng, et al. Pedestrian detection in fish-eye images using deep learning: combine faster R-CNN with an effective cutting method[C]. Proceedings of the 2018 International Conference on Signal Processing and Machine Learning, New York: Association for Computing Machinery, 2018:55 - 59.

[2] KUBO Y, KITAGUCHI T, YAMAGUCHI J. Human tracking using fisheye images[C]. SICE Annual Conference 2007, Takamatsu: IEEE, 2007:2013-2017.

[3] MEINEL L, FINDEISEN M, HEß M, et al. Automated real-time surveillance for ambient assisted living using an omnidirectional camera[C]. 2014 IEEE International Conference on Consumer Electronics, Las Vegas: IEEE, 2014: 396-399.

[4] JIN Ganfeng, Person detection and tracking using fisheye images based on multi-feature fusion[D]. Xi′an: Xidian University,2017.
金敢峰.基于多特征融合的鱼眼图像行人检测与跟踪[D].西安:西安电子科技大学,2017.

[5] AHMAD M, AHMED I, ULLAH K , et al. Person detection from overhead view: a survey[J]. International Journal of Advanced Computer Science and Applications, 2019, 10(4):567-577.

[6] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-Time object detection[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas: IEEE, 2016:779-788.

[7] LIU Wei, ANGUELOV D, ERHAN D, et al. SSD: single shot multi box detector[C]. European Conference on Computer Vision, Amsterdam: Springer, 2016:21-37.

[8] REN Shaoqing, HE Kaiming, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[9] LAW Hei, DENG Jia. CornerNet: detecting objects as paired keypoints[J]. International Journal of Computer Vision, 2020,128:642-656.

[10] SEIDEL R, APITZSCH A, HIRTZ G. Improved person detection on omnidirectional images with non-maxima suppression[C]. 14th International Conference on Computer Vision Theory and Applications, 2019.

[11] LI Shengye, TEZCAN M, ISHWAR P, et al. Supervised people counting using an overhead fisheye camera [C]. 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance, Taipei: IEEE, 2019:1-8.

[12] CHIANG Shengho, WANG Tsaipei, CHEN Yifu. Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches [J]. Image and Vision Computing, 2021, 105:104069.

[13] TAMURA M, HORIGUCHI S, MURAKAMI T. Omnidirectional pedestrian detection by rotation invariant training[C]. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019.

[14] DUAN Zhihao, TEZCAN M, NAKAMURA H, et al. RAPiD: rotation-aware people detection in overhead fisheye images[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle: IEEE, 2020:2700-2709.

[15] REZATOFIGHI H, TSOI N, GWAK J, et al. Generalized intersection over union: a metric and a Loss for bounding box regression [C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach: IEEE, 2019: 658-666.

[16] ZHENG Zhaohui, WANG Ping, LIU Wei, et al. Distance-IoU loss: faster and better learning for bounding box regression [C]. Proceedings of the AAAI Conference on Artificial Intelligence, New York: AAAI,2020:12993-13000.

[17] LIU Zili, ZHENG Tu, XU Guodong, et al. Training-time-friendly network for real-time object detection [C]. Proceedings of the AAAI Conference on Artificial Intelligence, New York: AAAI, 2020:11685-11692.

[18] ZHOU Xingyi, WANG Dequan, KRÄHENBÜHL P. Objects as points[J]. arXiv preprint arXiv: 1904.07850, 2019.

[19] TIAN Zhi, SHEN Chunhua, CHEN Hao, et al. FCOS: fully convolutional one-stage object detection [C]. 2019 IEEE/CVF International Conference on Computer Vision, Seoul: IEEE,2019: 9626-9635.

[20] KONG Tao, SUN Fuchun, LIU Huaping, et al. FoveaBox: beyond anchor-based object detector[J]. arXiv preprint arXiv: 1904.03797, 2019.

[21] ZHANG Yifan, REN Weiqiang, ZHANG Zhang, et al. Focal and efficient IOU loss for accurate bounding box regression [J]. arXiv preprint arXiv: 2101.08158,2021.

[22] YU Jiahui, JIANG Yuning, WANG Zhangyang, et al. Unitbox: An advanced object detection network[C]. Proceedings of the 24th ACM international conference on Multimedia, 2016: 516-520.

[23] CHEN Zhiming, CHEN Kean, LI Weiyao, et al. PIoU loss: towards accurate oriented object detection in complex environments[C]. European Conference on Computer Vision, Glasgow: Springer,2020:195-211.


**Foundation item**: Quality and Technology Infrastructure Project of Zhejiang Provincial Market Supervision and Administration (No. 20200125)