doi：10.13413/j.cnki.jdxblxb.2023213

一种边缘梯度插值的感兴趣区域池化算法

周跃进[1,2]，丁家益[1]

（1. 安徽理工大学 数学与大数据学院，安徽 淮南 232001；
2. 深部煤矿采动响应与灾害防控国家重点实验室，安徽 淮南 232001）

摘要：针对现有主流的目标检测算法存在检测精确率低、图像边缘区域分割不全等问题，提出一种基于 Mask RCNN 模型的感兴趣区域池化算法. 首先，通过 Otsu 阈值分割法将感兴趣区域特征图划分为边缘区域和非边缘区域；其次，对边缘区域使用边缘梯度插值算法进行插值，对非边缘区域使用双线性插值算法进行插值，从而将离散的特征图映射到一个连续空间中；再次，将插值后的特征图均匀分割成 $k\times k$ 个单元；最后，对每个单元利用二重积分求均值以完成池化操作. 对比实验结果表明，该算法基于 Mask RCNN 模型在数据集 COCO (2014)上比现有算法的检测精确率有一定提升，对图像边缘区域的细节分割效果较好.

关键词：Mask RCNN 模型；感兴趣区域池化；Otsu 阈值分割；边缘梯度插值；双线性插值
中图分类号：TP391 **文献标志码**：A **文章编号**：1671-5489(2024)03-0643-12

A Region of Interest Pooling Algorithm for Edge Gradient Interpolation

ZHOU Yuejin[1,2], DING Jiayi[1]

(1. *School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan* 232001, *Anhui Province, China*; 2. *State Key Laboratory of Mining Response and Disaster Prevention and Control in Deep Coal Mines, Huainan* 232001, *Anhui Province, China*)

Abstract: Aiming at the problems that the existing mainstream target detection algorithms had low detection accuracy and incomplete segmentation in the image edge regions, we proposed a region of interest pooling algorithm based on Mask RCNN model. Firstly, the feature maps of the regions of interest were divided into edge regions and non-edge regions by the Otsu threshold segmentation method. Secondly, the edge gradient interpolation algorithm was used to interpolate for the edge regions, and the bilinear interpolation algorithm was used to interpolate for the non-edge regions so that the discrete feature map was mapped into a continuous space. Thirdly, the interpolated feature maps were evenly divided into $k\times k$ units. Finally, the double integral was used to calculate the average value of each unit to complete the pooling operation. The comparative experimental results show that the proposed algorithm, based on the Mask RCNN model, has a certain improvement in detection accuracy compared with existing algorithms on COCO (2014) dataset, and has a good segmentation effect on the details of the image edge regions.

收稿日期：2023-06-02.

第一作者简介：周跃进(1977—)，男，汉族，博士，教授，从事统计机器学习和因果分析的研究，E-mail：yjzhou@aust.edu.cn.
通信作者简介：丁家益(1998—)，男，汉族，硕士研究生，从事统计机器学习的研究，E-mail：2806166640@qq.com.

基金项目：国家自然科学基金（批准号：61703005）和深部煤矿采动响应与灾害防控国家重点实验室基金（批准号：SKLMRDPC22KF03）.

Keywords: Mask RCNN model; region of interest pooling; Otsu threshold segmentation; edge gradient interpolation; bilinear interpolation

目标检测和图像分割已广泛应用于自动驾驶、视频监控、机器人等多个领域[1]. 在实际应用场景中，目标检测和图像分割的精确率常会受背景、遮挡、光照等不确定因素的影响. 因此，目标检测和图像分割是一个具有挑战性的研究课题[2].

目标检测模型主要分为两种：单阶段检测模型和两阶段检测模型. 单阶段检测模型检测速度快，但检测精确率相对较低. 单阶段检测模型主要包括 YOLO[3] 目标检测模型和 SSD[4] 目标检测模型. 前者是一种端到端的一步检测模型，以 45 帧/s 达到了接近实时的目标检测速度；后者引入了特征金字塔和多长宽比多尺度的密集锚点设计，在数据集 VOC 上的检测精确率达到 74.3%，检测帧率也提高到 46 帧/s. 两阶段检测模型检测精确率高，但检测速度相对较慢，在对检测精确率要求高的实际场景中应用更广泛. 两阶段检测模型主要包括以下模型：根据 AlexNet[5] 在图像特征提取方面提出的 RCNN[6] 目标检测模型，利用选择性搜索算法生成大量候选区域，再对每个候选区域进行特征提取、识别，并使用回归器修正候选区域的位置；Girshick[7] 提出的 Fast RCNN 目标检测模型，首次引入了感兴趣区域 RoI(region of interest)池化层，在数据集 VOC 上的检测精确率达到 68.4%；为解决选择性搜索算法生成候选区域的严重耗时问题，Ren 等[8] 提出了 Faster RCNN 目标检测模型，应用区域候选网络 RPN(region proposal network)生成候选区域，进一步提高了目标检测的精确率和帧率；为满足图像分割的需求，文献[9]提出的 Mask RCNN 目标检测模型在 Faster RCNN 模型的结构上添加了一个用于预测目标掩模的分支，并具有一个感兴趣区域 RoI Align 池化层，在数据集 COCO 上取得了良好的目标检测和图像分割效果.

在两阶段检测模型中，Fast RCNN 模型和 Faster RCNN 模型所使用的 RoI 池化算法存在两次量化操作，导致原图中的像素与特征图中的像素不对齐，候选框位置产生偏差. Mask RCNN 模型所使用的 RoI Align 池化算法利用双线性差值算法[10] 确定感兴趣区域中点的像素值，易导致图像边缘信息丢失以及需要插值点的数目难以自适应的问题.

针对上述问题，本文提出一种更精确的感兴趣区域池化算法，即 MpRoI(more precise region of interest)池化算法. 为评估 MpRoI 池化算法的性能，基于 Mask RCNN 模型使用数据集 COCO(2014)与 RoI 池化算法、RoI Align 池化算法进行比较. 对比实验结果表明，MpRoI 池化算法的检测精确率和稳定性高于另外两种池化算法，且具有更低的分类损失和边界框回归损失，明显改善了边缘区域图像分割的锯齿现象，使目标的定位更准确.

1 RoI 池化算法

RoI 池化层位于区域候选网络层和全连接层之间，作用是将不同尺寸的候选特征图转化为固定数据输出. RoI 池化算法的输入由卷积神经网络输出的特征图和区域候选网络输出的候选框两部分组成. RoI 池化算法的输出为一组向量，向量个数由区域候选网络输出的候选框数量确定，向量大小为 $C\times W\times H$，其中 C 为通道数，W 和 H 为超参数. RoI 池化算法的工作流程如图 1 所示.

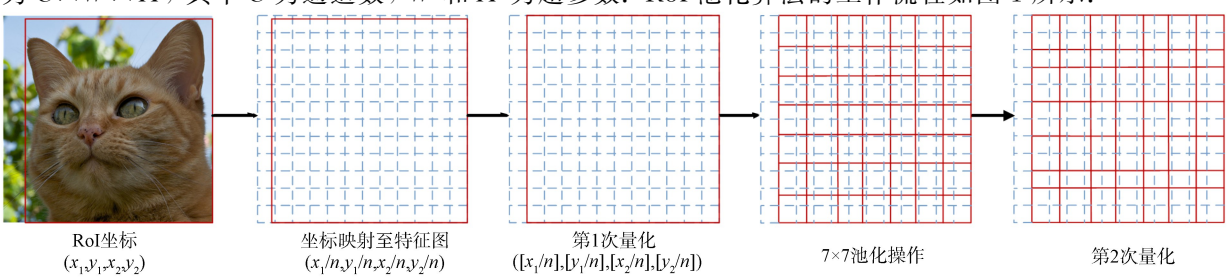


图 1 RoI 池化算法工作流程

Fig. 1 Working flow chart of RoI pooling algorithm

由图 1 可见，RoI 池化算法首先将候选区域映射到卷积神经网络输出的特征图上，由于卷积神经网络的池化操作，特征图的尺寸相比于原图缩小了 n 倍，然后对映射后的特征图区域进行特征提取，使不同尺寸大小的特征图区域转化为一个固定维度的输出向量. 由于 RoI 池化算法存在两次量化操作，从而导致像素的位置产生偏差，降低了目标检测的精确率[11].

2　RoI Align 池化算法

RoI Align 池化算法首先遍历特征图上的每个候选区域，保持浮点数边界不进行量化操作，然后将候选区域均匀分割成 $k\times k$ 个单元(bin)，每个单元的高度和宽度数值也保持浮点数边界不做量化，再在每个单元中通过计算确定 4 个点的坐标(均匀选取 4 个采样点)，利用双线性差值算法计算得出 4 个点的像素值，最后对 4 个点的像素值进行最大池化操作得到每个单元的值. RoI Align 池化算法中最重要的方法是使用了双线性差值算法计算得出采样点的像素值[12]，避免了量化操作引入的误差，即特征图中的像素与原图中的像素完全对齐. 双线性插值算法的工作流程如图 2 所示.

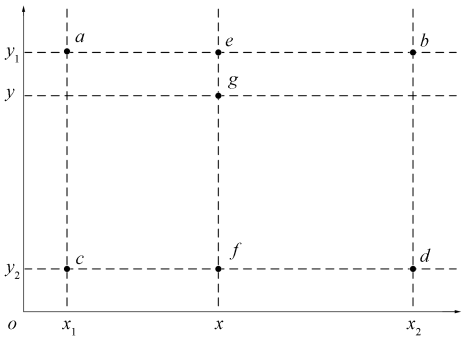


图 2　双线性插值算法工作流程

Fig. 2　Working flow chart of bilinear interpolation algorithm

假设将点 g 设为需要插值的点，已知 4 个点 a,b,c,d 的值，通过点 a 和点 b 做线性插值得到点 e，通过点 c 和点 d 做线性插值得到点 f，计算公式如下：

$$\begin{aligned} f(f) &\approx \frac{x_2-x}{x_2-x_1}\cdot f(c)+\frac{x-x_1}{x_2-x_1}\cdot f(d),\\ f(e) &\approx \frac{x_2-x}{x_2-x_1}\cdot f(a)+\frac{x-x_1}{x_2-x_1}\cdot f(b); \end{aligned} \tag{1}$$

再由点 e 和点 f 做线性插值得到点 g，计算公式如下：

$$f(g)\approx \frac{y_2-y}{y_2-y_1}\cdot f(f)+\frac{y-y_1}{y_2-y_1}\cdot f(e); \tag{2}$$

最终合并为

$$\begin{aligned} f(x,y)\approx &\frac{f(c)}{(x_2-x_1)(y_2-y_1)}\cdot(x_2-x)(y_2-y)+\frac{f(d)}{(x_2-x_1)(y_2-y_1)}\cdot(x-x_1)(y_2-y)+\\ &\frac{f(a)}{(x_2-x_1)(y_2-y_1)}\cdot(x_2-x)(y-y_1)+\frac{f(b)}{(x_2-x_1)(y_2-y_1)}\cdot(x-x_1)(y-y_1). \end{aligned} \tag{3}$$

相比于 RoI 池化算法，RoI Align 池化算法遍历的取样点数量较少，但性能更好，这主要是由于其解决了区域不匹配的问题. RoI Align 池化算法的工作流程如图 3 所示.

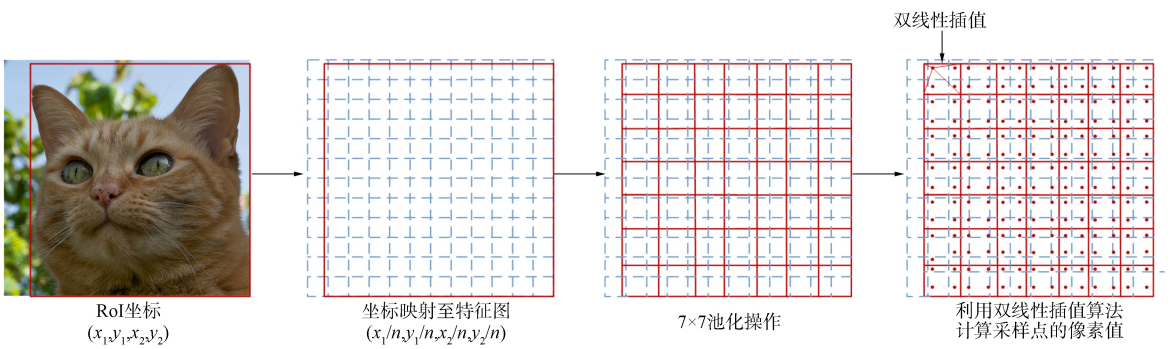


图 3　RoI Align 池化算法工作流程

Fig. 3　Working flow chart of RoI Align pooling algorithm

3 MpRoI池化算法

为消除RoI池化算法存在两次量化操作对候选框位置产生的偏差，解决RoI Align池化算法需要插值点的数目难以自适应的问题及使用双线性插值算法处理图像易导致图像边缘信息丢失的问题，本文提出一种更精确的感兴趣区域池化算法，即MpRoI池化算法. 首先，通过Otsu阈值分割法[13]将感兴趣区域特征图划分为边缘区域和非边缘区域；其次，对边缘区域使用边缘梯度插值算法进行插值，对非边缘区域使用双线性插值算法进行插值，从而将离散的特征图映射到一个连续空间中，再将插值后的特征图均匀分割成 $k\times k$ 个单元；最后对每个单元利用二重积分求均值完成池化操作.

3.1 Otsu阈值分割法

Otsu阈值分割法是一种自适应于双峰情况自动求取阈值的方法，以图像的灰度直方图为依据，选取阈值是以目标区域与背景区域平均灰度的最大类间方差为基准[14]. Otsu阈值分割法的基本思想如下：设图像中灰度值为 i 的像素个数为 n_i，灰度值的取值范围为$[0,L-1]$，记 $G=\{0,1,2,\cdots,L-1\}$，则像素总数为

$$N=\sum_{i=0}^{L-1}n_i. \tag{4}$$

灰度值为 i 的像素出现的概率为

$$p_i=n_i/N. \tag{5}$$

对于 p_i，有

$$\sum_{i=0}^{L-1}p_i=1. \tag{6}$$

将图像中像素用阈值 T 分为两类 C_0 和 C_1，灰度值取值范围为$[0,T-1]$的像素归为 C_0 类，取值范围为$[T,L-1]$的像素归为 C_1 类，则 C_0 类和 C_1 类的概率分别为

$$p_0=\sum_{i=0}^{T-1}p_i, \tag{7}$$

$$p_1=\sum_{i=T}^{L-1}p_i. \tag{8}$$

C_0 类和 C_1 类的平均灰度值分别为

$$u_0=\frac{1}{p_0}\sum_{i=0}^{T-1}ip_i, \tag{9}$$

$$u_1=\frac{1}{p_1}\sum_{i=T}^{L-1}ip_i. \tag{10}$$

图像的平均灰度值可表示为

$$u=\sum_{i=0}^{T-1}ip_i+\sum_{i=T}^{L-1}ip_i=p_0u_0+p_1u_1. \tag{11}$$

图像两种类别的总方差为

$$\sigma^2=p_0(u_0-u)^2+p_1(u_1-u)^2=p_0p_1(u_0-u_1)^2. \tag{12}$$

当 σ^2 达到最大值时对应的灰度值为最优分割阈值，即Otsu阈值：

$$T=\underset{T\in G}{\operatorname{argmax}}\{\sigma^2\}. \tag{13}$$

图像非边缘区域像素的灰度值与其邻域像素的平均灰度值接近，而边缘区域像素的灰度值会在一定程度上高于或者低于其邻域像素的平均灰度值. 因此，Otsu阈值分割法通过将待插值像素邻域的4个已知像素的平均灰度值与最优分割阈值 T 相比较以确定为非边缘区域或边缘区域.

3.2 边缘梯度插值算法

传统的双线性插值算法具有低通滤波的性质，易使图像的高频分量产生损失、图像边缘在一定程度上变得较模糊[15]. 因此，本文提出一种边缘梯度插值算法对特征图的边缘区域进行插值. 本文的插值算法主要有以下几个步骤.

首先，对特征图进行高斯滤波处理以减少噪声对边缘检测的影响，本文使用一个 5×5 的滤波窗口对特征图进行处理，滤波窗口如下：

$$\frac{1}{159}\begin{pmatrix}2 & 4 & 5 & 4 & 2\\ 4 & 9 & 12 & 9 & 4\\ 5 & 12 & 15 & 12 & 5\\ 4 & 9 & 12 & 9 & 4\\ 2 & 4 & 5 & 4 & 2\end{pmatrix}. \tag{14}$$

其次，利用如图 4 所示的 Sobel 算子[16]梯度计算模板分别计算高斯滤波后的特征图中像素点水平方向和垂直方向的梯度分量 G_x 和 G_y.

−1	0	+1
−2	0	+2
−2	0	+1

(A) 水平方向

+1	+2	+1
0	0	0
−1	−2	−1

(B) 垂直方向

图 4　梯度计算模板

Fig. 4　Gradient computing template

由于特征图的梯度方向所在直线与边缘主导方向所在直线相互垂直，因此定义边缘主导方向为其所在直线与水平轴正方向的逆时针夹角 θ 方向[17]为

$$\theta=\arctan\left(\frac{\sum_{i=1}^{n}\mathrm{Sign}(Gx_i)\cdot(-Gy_i)}{\sum_{i=1}^{n}\mathrm{Sign}(Gx_i)\cdot Gx_i}\right)+\frac{\pi}{2}, \tag{15}$$

$$\mathrm{Sign}(x)=\begin{cases}1, & x>0,\\ -1, & x<0,\end{cases} \tag{16}$$

其中 θ 的取值范围为$[0,\pi)$，n 为区域像素点的个数.

最后，在特征图的边缘区域待插值位置沿边缘主导方向进行插值. 特征图的边缘区域插值存在 6 种可能，如图 5 所示. 图 6 为其中一种可能情形的表示.

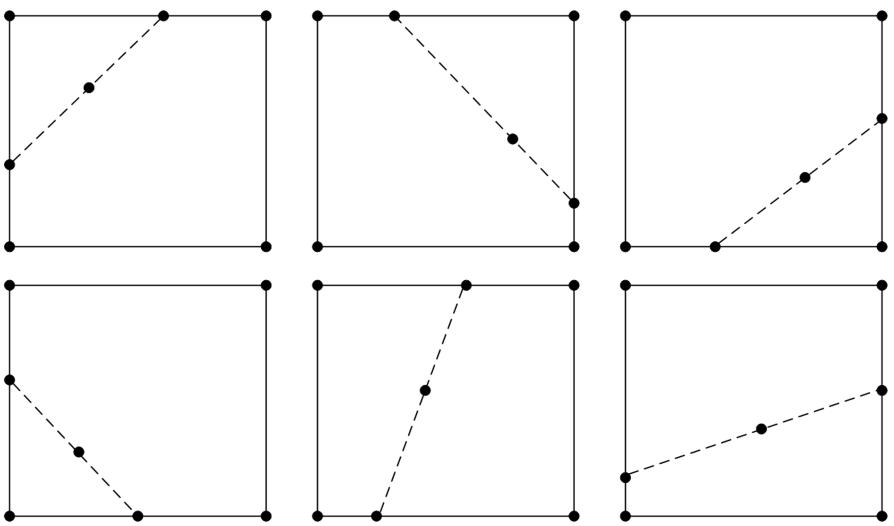


图 5　特征图边缘区域插值可能存在的情形

Fig. 5　Possible cases of interpolation in edge region of feature graph

由图6可见，过待插值点 g 做一条平行于边缘主导方向的直线，交于区域边缘 bc 和 ab 于 f 和 e 两点. 点 f 的像素值可由 b，c 两点像素值进行线性插值得到，同理，点 e 的像素值可由 a，b 两点像素值进行线性插值得到. 最后，待插值点 g 的像素值可由 f，e 两点像素值进行线性插值得到. 图5中特征图边缘区域其他情形的插值操作类似.

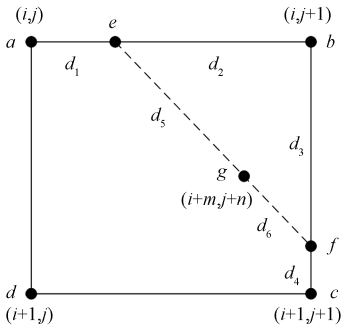


图6 插值实例

Fig. 6 Interpolation example

3.3 MpRoI池化算法流程

给定一个图像感兴趣区域的特征图 A，令 (i,j) 为特征图上的坐标. 对非边缘区域，利用双线性插值算法将离散的特征图区域 $\omega_{i,j}$ 映射到一个连续空间中，映射公式为

$$f_1(x,y)=\sum_{i,j} IC(x,y,i,j)\times\omega_{i,j}, \tag{17}$$

其中：$f_1(x,y)$ 表示经过双线性插值算法插值后连续的特征图；$IC(x,y,i,j)$ 为双线性插值算法的插值系数，可表示为

$$IC(x,y,i,j)=\max\{0,1-|x-i|\}\times\max\{0,1-|y-j|\}. \tag{18}$$

将插值后的特征图均匀分割成 $k\times k$ 个单元，再对分割后的单元进行二重积分求均值操作，计算公式为

$$\mathrm{MpRoI}_1(\mathrm{bin},A)=\frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2} f_1(x,y)\mathrm{d}x\mathrm{d}y}{(x_2-x_1)\times(y_2-y_1)}. \tag{19}$$

对 $\mathrm{MpRoI}_1(\mathrm{bin},A)$ 求 x_1 偏导数可得

$$\begin{aligned}\frac{\partial\mathrm{MpRoI}_1(\mathrm{bin},A)}{\partial x_1}&=\frac{\partial\left\{\int_{y_1}^{y_2}\int_{x_1}^{x_2} f_1(x,y)\mathrm{d}x\mathrm{d}y/[(x_2-x_1)\times(y_2-y_1)]\right\}}{\partial x_1}=\\&\frac{\partial\int_{y_1}^{y_2}\int_{x_1}^{x_2} f_1(x,y)\mathrm{d}x\mathrm{d}y/\partial x_1\times(x_2-x_1)\times(y_2-y_1)}{[(x_2-x_1)\times(y_2-y_1)]^2}-\\&\frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2} f_1(x,y)\mathrm{d}x\mathrm{d}y\times\dfrac{\partial(x_2-x_1)\times(y_2-y_1)}{\partial x_1}}{[(x_2-x_1)\times(y_2-y_1)]^2}=\\&\frac{\partial\int_{y_1}^{y_2}\int_{x_1}^{x_2} f_1(x,y)\mathrm{d}x\mathrm{d}y/\partial x_1}{(x_2-x_1)(y_2-y_1)}-\frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2} f_1(x,y)\mathrm{d}x\mathrm{d}y\times[-1\times(y_2-y_1)]}{[(x_2-x_1)\times(y_2-y_1)]^2}=\\&\frac{\mathrm{MpRoI}_1(\mathrm{bin},A)}{(x_2-x_1)}-\frac{\int_{y_1}^{y_2} f_1(x,y)\mathrm{d}y}{(x_2-x_1)\times(y_2-y_1)}.\end{aligned} \tag{20}$$

对 $\mathrm{MpRoI}_1(\mathrm{bin},A)$ 求 x_2 偏导数类似式(20).

对边缘区域，利用边缘梯度插值算法将离散的特征图区域 $\omega_{m,n}$ 映射到一个连续空间中，映射公式为

$$f_2(x,y)=\sum_{m,n} ID(x,y,m,n)\times\omega_{m,n}, \tag{21}$$

其中：$f_2(x,y)$ 为经过边缘梯度插值算法插值后连续的特征图；$ID(x,y,m,n)$ 为边缘梯度插值算法的插值系数，可表示为

$$ID(x,y,m,n)=\max\{0,1-|\theta x-m|\}\times\max\{0,1-|y/\theta-n|\}, \tag{22}$$

式中 θ 为特征图的梯度方向所在直线与水平轴正方向的逆时针夹角.

将插值后的特征图均匀分割成 $k\times k$ 个单元，再对分割后的单元进行二重积分求均值操作：

$$\mathrm{MpRoI}_2(\mathrm{bin},A)=\frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2}f_2(x,y)\mathrm{d}x\mathrm{d}y}{(x_2-x_1)\times(y_2-y_1)}. \tag{23}$$

对 $\mathrm{MpRoI}_2(\mathrm{bin},A)$求 x_1 偏导数可得

$$\frac{\partial\mathrm{MpRoI}_2(\mathrm{bin},A)}{\partial x_1}=\frac{\partial\left\{\int_{y_1}^{y_2}\int_{x_1}^{x_2}f_2(x,y)\mathrm{d}x\mathrm{d}y/[(x_2-x_1)\times(y_2-y_1)]\right\}}{\partial x_1}=$$

$$\frac{\partial\int_{y_1}^{y_2}\int_{x_1}^{x_2}f_2(x,y)\mathrm{d}x\mathrm{d}y/\partial x_1\times(x_2-x_1)\times(y_2-y_1)}{[(x_2-x_1)\times(y_2-y_1)]^2}-$$

$$\frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2}f_2(x,y)\mathrm{d}x\mathrm{d}y\times[\partial(x_2-x_1)\times(y_2-y_1)/\partial x_1]}{[(x_2-x_1)\times(y_2-y_1)]^2}=$$

$$\frac{\partial\int_{y_1}^{y_2}\int_{x_1}^{x_2}f_2(x,y)\mathrm{d}x\mathrm{d}y/\partial x_1}{(x_2-x_1)\times(y_2-y_1)}-\frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2}f_2(x,y)\mathrm{d}x\mathrm{d}y\times[-1\times(y_2-y_1)]}{[(x_2-x_1)\times(y_2-y_1)]^2}=$$

$$\frac{\mathrm{MpRoI}_2(\mathrm{bin},A)}{(x_2-x_1)}-\frac{\int_{y_1}^{y_2}f_2(x,y)\mathrm{d}y}{(x_2-x_1)\times(y_2-y_1)}. \tag{24}$$

对 $\mathrm{MpRoI}_2(\mathrm{bin},A)$求 x_2 偏导数类似式(24).

由式(20),(24)可知，函数 $\mathrm{MpRoI}_1(\mathrm{bin},A)$和 $\mathrm{MpRoI}_2(\mathrm{bin},A)$是连续可微的，通过双线性插值算法、边缘梯度插值算法和二重积分求均值等操作避免了感兴趣区域池化中的量化，从而有效降低了量化操作对特征提取带来的精度损失[18].

算法 1　MpRoI 池化算法.

1) begin

2) 初始化 $x\leftarrow0$，$y\leftarrow0$，$i\leftarrow0$，$j\leftarrow0$，$m\leftarrow0$，$n\leftarrow0$

3) 参数 $p_i,C_0,C_1,u_0,u_1,\omega_{i,j},\omega_{m,n},\theta$ //p_i 为灰度值为 i 的像素出现的概率，C_0 为灰度值取值范围为$[0,T-1]$的像素所属类别，C_1 为灰度值取值范围为$[T,L-1]$的像素所属类别，u_0 为 C_0 类像素的平均灰度值，u_1 为 C_1 类像素的平均灰度值，$\omega_{i,j},\omega_{m,n}$为特征图区域，$\theta$ 为特征图的梯度方向所在直线与水平轴正方向的逆时针夹角

4) 给定 $p_0=\sum_{i=0}^{T-1}p_i$，$p_1=\sum_{i=T}^{L-1}p_i$，$u_0=\frac{1}{p_0}\sum_{i=0}^{T-1}ip_i$，$u_1=\frac{1}{p_1}\sum_{i=T}^{L-1}ip_i$

5) $u=p_0u_0+p_1u_1$ //图像的平均灰度值

6) $\sigma^2=p_0p_1(u_0-u_1)^2$ //图像两种类别的总方差

7) $\Gamma=\underset{T\in G}{\mathrm{argmax}}\{\sigma^2\}$ //最优分割阈值

8) if (局部平均灰度值$<\Gamma$)

{

$f_1(x,y)=\sum_{i,j}IC(x,y,i,j)\times\omega_{i,j}$ //非边缘区域

$IC(x,y,i,j)=\max\{0,1-|x-i|\}\times\max\{0,1-|y-i|\}$ //双线性插值算法的插值系数

}

else

{

$f_2(x,y)=\sum_{m,n}ID(x,y,m,n)\times\omega_{m,n}$ //边缘区域

$ID(x,y,m,n)=\max\{0,1-|\theta x-m|\}\times\max\{0,1-|y/\theta-n|\}$ //边缘梯度插值算法的插值系数

}

9) 分割成 $k\times k$ 个单元，二重积分求均值

10) end.

4 实验与结果分析

4.1 实验数据集

数据集 COCO 是微软团队出资标注的一个可用于目标检测、分割和图像描述的数据集，主要为从日常复杂场景中选取的自然图像以及生活中常见的目标图像. 数据集提供了 80 类目标，超出 33 万张图像，其中图像主要以关键点检测、物体检测、实例分割、全景分割、图像标注等 5 种类型进行标注，以 json 文件格式存储. 本文实验使用数据集 COCO(2014)，包括 82 783 张训练图像、40 775 张测试图像及 40 504 张验证图像.

4.2 模型的训练

实验采用的显卡为 AMD RX 6900XT，显存为 16 GB. 模型的训练基于 Python 语言环境下的 TensorFlow 深度学习框架进行. 实验参数设置如下：初始学习率为 0.001，网络迭代次数(epoch)为 300 次，每个 epoch 迭代 30 次，权重衰减系数为 5×10^{-4}，正则化为 1.6×10^{-3}. 一次完整的训练过程需进行 9 000 次训练.

4.3 评价指标

4.3.1 精确率

精确率(Precision)又称为查准率，其为被分类器正确识别出的样本个数占总识别出样本个数的百分数，计算公式为

$$p=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\times100\%, \tag{25}$$

其中 TP 表示被正确识别出的样本个数，FP 表示未被正确识别出的样本个数. 本文以 mAP(IoU=0.50∶0.05∶0.95)，AP50(IoU=0.50)和 AP75(IoU=0.75)3 个精确率评价标准分析实验结果.

4.3.2 损失函数

损失函数是用来衡量算法输出的预测值与真实值之间的偏离程度. 本文实验使用分类损失函数和边界框回归损失函数评估算法的性能. 分类损失函数 L_1 计算公式为

$$L_1=\frac{1}{N_{\mathrm{cls}}}\sum_i L_{\mathrm{cls}}(p_i,p_i^*), \tag{26}$$

$$L_{\mathrm{cls}}=-[p_i^*\log(p_i)+(1-p_i^*)\log(1-p_i)], \tag{27}$$

其中：N_{cls}表示一个小批量(mini-batch)中所有样本的数量；p_i 表示第 i 个锚框(anchor)预测为真实标签的概率；当预测样本为正样本时，$p_i^*=1$；当预测样本为负样本时，$p_i^*=0$. 边界框回归损失函数 L_2 的计算公式为

$$L_2=\frac{\lambda}{N_{\mathrm{reg}}}\sum_i p_i^* L_{\mathrm{reg}}(t_i,t_i^*), \tag{28}$$

$$L_{\mathrm{reg}}(t_i,t_i^*)=\sum_i \mathrm{Smooth}(t_i^*-t_i), \tag{29}$$

$$\mathrm{Smooth}(x)=\begin{cases}0.5x^2, & |x|<1,\\ |x|-0.5, & \text{其他},\end{cases} \tag{30}$$

其中 λ 为平衡系数，N_{reg}为锚框位置的个数，t_i 为第 i 个锚框对应的边界框回归参数，t_i^* 为预测的第 i 个锚框的边界框回归参数.

4.3.3 平均处理时间

平均处理时间能有效评价模型的识别速度，可作为模型的实时性评价标准[18]，计算公式为

$$\hat{T}=T^*/N^*, \tag{31}$$

其中 $\hat{T}$ 为平均处理时间，N^* 为测试图片数量，T^* 为测试运行时间.

4.4　实验结果及分析

为更好地评估 MpRoI 池化算法的性能，本文基于 Mask RCNN 模型使用数据集 COCO(2014)与 RoI 池化算法、RoI Align 池化算法进行比较. 图 7 和图 8 分别为各算法经过 300 次迭代的精确率(mAP,AP50,AP75)和损失(分类损失、边界框回归损失)变化的比较结果.

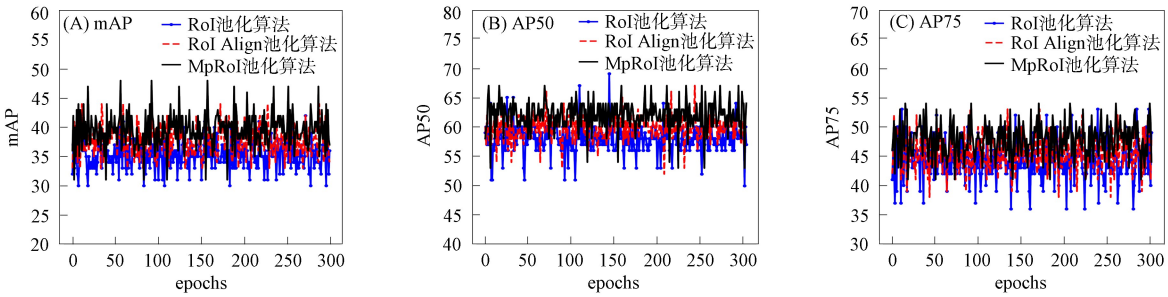


图 7　各算法精确率的比较

Fig. 7　Comparison of accuracy of each algorithm

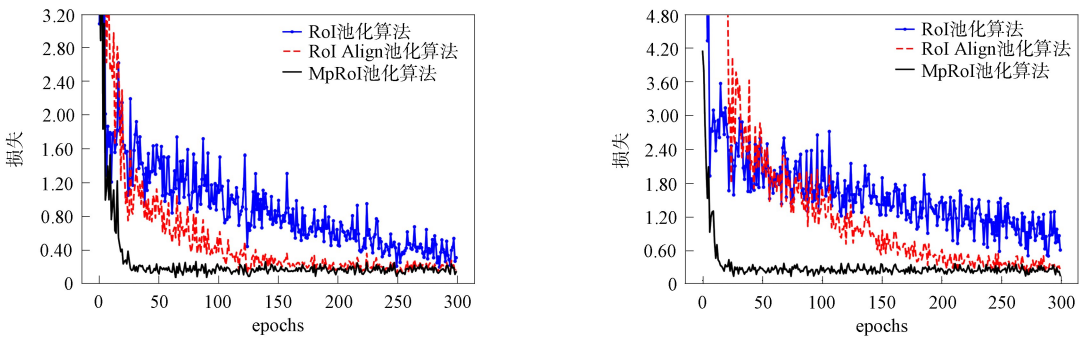


图 8　各算法损失变化的比较

Fig. 8　Comparison of loss changes of each algorithm

由图 7 可见：MpRoI 池化算法的精确率和稳定性高于 RoI 池化算法和 RoI Align 池化算法；RoI 池化算法的精确率最低，且在 80<epochs<130 时出现了大幅度震荡；RoI Align 池化算法的精确率和稳定性虽然明显高于 RoI 池化算法，但相比于 MpRoI 池化算法略显不足. 实验结果表明，MpRoI 池化算法在精确率和稳定性方面具有良好的性能. 由图 8 可见，MpRoI 池化算法具有更低的分类损失和边界框回归损失. 在迭代过程中，MpRoI 池化算法的分类损失和边界框回归损失也一直保持稳定下降的趋势，并且以更少的迭代次数达到最小值. 实验结果表明，MpRoI 池化算法具有良好的鲁棒性.

表 1 列出了各算法 300 次迭代的精确率平均值，表 2 列出了各算法经过 300 次迭代后最终的分类损失值和边界框回归损失值.

表 1　各算法迭代的精确率平均值

Table 1　Average accuracy of each algorithm iteration

算法	mAP/%	AP50/%	AP75/%	迭代次数
RoI 池化	35.08	58.13	44.21	300
RoI Align 池化	37.12	60.02	46.32	300
MpRoI 池化	38.04	60.78	47.19	300

表 2　各算法最终的分类损失值和边界框回归损失值

Table 2　Final classification loss values and bounding box regression loss values of each algorithm

算法	分类损失值	边界框回归损失值	迭代次数
RoI 池化	0.279	0.614	300
RoI Align 池化	0.267	0.320	300
MpRoI 池化	0.261	0.312	300

由表 1 可见，MpRoI 池化算法的精确率较 RoI Align 池化算法、RoI 池化算法得到了一定的提升，

与RoI Align池化算法相比，mAP值提高了0.92个百分点，AP50值提高了0.76个百分点，AP75值提高了0.87个百分点. 由表2可见，MpRoI池化算法的分类损失值和边界框回归损失值最低，RoI Align池化算法次之，RoI池化算法最高. 从RoI池化算法到RoI Align池化算法，边界框回归损失值降低的幅度较大，为0.294，这是因为RoI Align池化算法使用双线性差值算法避免了量化操作引入的误差. 此外，从RoI Align池化算法到MpRoI池化算法，分类损失值和边界框回归损失值都有一定程度降低，表明引入边缘梯度插值算法和二重积分求均值等操作可进一步避免量化引入的误差.

时间测试实验也基于Mask RCNN模型在数据集COCO(2014)上进行，比较各算法进行目标检测与图像分割的参数量和平均耗时，结果列于表3.

表3 各算法进行目标检测与图像分割的参数量和平均耗时的比较

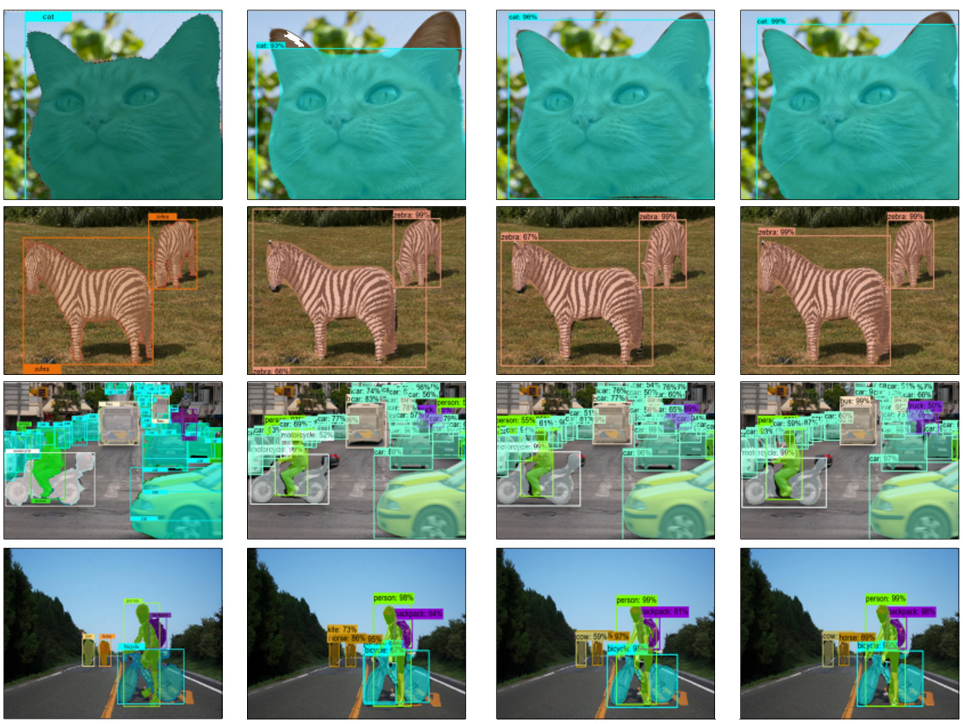
Table 3 Comparison of parameter quantities and average time consumption of target detection and image segmentation by each algorithm

算法	平均处理时间/ms	参数量	迭代次数
RoI池化	216	32.18×10^6	300
RoI Align池化	288	34.07×10^6	300
MpRoI池化	319	35.89×10^6	300

由表3可见，MpRoI池化算法的参数量最高，这是由于其空间复杂度和计算复杂度提高以及基于Mask RCNN模型添加了图像分割分支所致. 此外，在数据集COCO(2014)实验中，相比于RoI池化算法、RoI Align池化算法，MpRoI池化算法进行目标检测与图像分割的平均处理时间分别延长了103 ms和31 ms. 但MpRoI池化算法能以较少的时间花费换取更高的目标检测与图像分割精确率以及更低的分类损失和边界框回归损失，达到满意的目标检测与图像分割效果.

4.4.1 算法综合性能定性分析

为更好地说明MpRoI池化算法基于Mask RCNN模型的泛化性和检测性能，与其他算法在数据集COCO(2014)上进行检测对比，选取部分具有代表性的检测结果如图9所示.



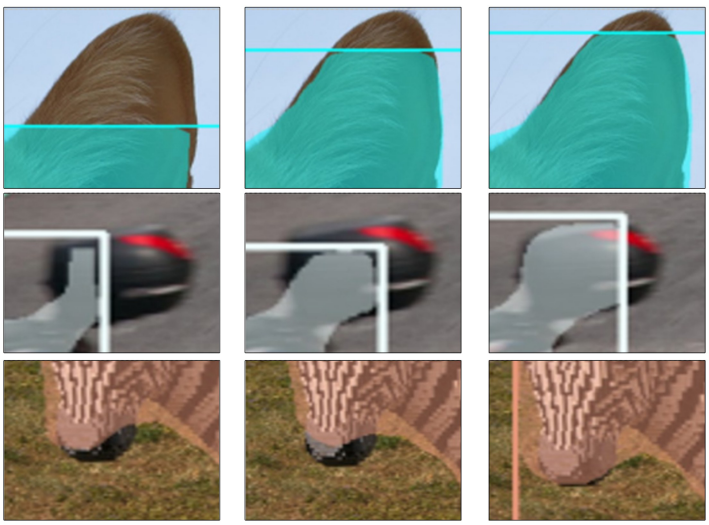
(A) 真实图像 (B) RoI池化算法 (C) RoI Align池化算法 (D) MpRoI池化算法

图9 各算法基于Mask RCNN模型在数据集COCO(2014)上的检测结果对比

Fig. 9 Comparison of detection results of each algorithm based on Mask RCNN model on COCO(2014) dataset

由图9可见，无论是单目标图像还是多目标图像，本文提出的MpRoI池化算法都能检测出更多的

目标. 对同样识别出的目标，MpRoI 池化算法的得分最高. 对高度重叠的目标，MpRoI 池化算法修正了 RoI 池化算法和 RoI Align 池化算法在检测时出现的目标漏检问题. 此外，MpRoI 池化算法明显改善了边缘区域图像分割的锯齿现象，也使目标的定位更准确. 图 10 为图 9 中部分边缘区域的细节放大情况.



(A) RoI池化算法　　(B) RoI Align池化算法　　(C) MpRoI池化算法

图 10　图 9 中部分边缘区域的细节放大图像

Fig. 10　Some detailed enlarged views of edge regoins in Fig. 9

由图 10 可见，对图像的边缘区域，MpRoI 池化算法的分割效果最好，边界特征更明显，结果更准确. RoI 池化算法和 RoI Align 池化算法易出现区域图像分割不全，且区域轮廓的分割质量相对较低. 因此，MpRoI 池化算法克服复杂背景的干扰能力更强，图像边缘区域的细节分割更完善，且能适应图像边缘区域的尺度形状变化，整体的目标检测性能得到一定提升.

4.4.2　算法综合性能定量分析

为进一步分析 MpRoI 池化算法基于 Mask RCNN 模型的目标检测性能，选取数据集 COCO(2014)中具有代表性的 20 类检测目标与 RoI 池化算法、RoI Align 池化算法进行检测对比(AP50)，检测对比结果列于表 4. 由表 4 可见，在数据集 COCO(2014)上的 20 类检测目标中，MpRoI 池化算法在 table,chair,car 等多个类别上的检测精确率高于 RoI 池化算法和 RoI Align 池化算法.

表 4　各算法基于 Mask RCNN 模型在数据集 COCO(2014)上的检测对比结果

Table 4　Detection comparison results of each algorithm based on Mask RCNN model on COCO(2014) dataset　%

算法	table	chair	car	bottle	bird	aero	bike	boat	bus	cat
RoI 池化	56.7	53.2	63.5	54.8	51.5	67.3	59.9	53.5	64.7	65.1
RoI Align 池化	58.2	55.7	64.3	57.2	51.9	70.2	62.3	53.7	65.2	66.7
MpRoI 池化	61.9	55.9	67.6	56.9	52.1	72.1	61.5	55.0	68.5	67.1
算法	dog	motor	plant	sofa	TV	train	sheep	person	horse	cow
RoI 池化	64.9	58.1	38.5	54.2	63.5	74.2	54.5	71.1	64.7	52.3
RoI Align 池化	65.2	60.4	38.7	55.2	65.1	75.5	55.6	72.6	65.1	53.5
MpRoI 池化	65.5	59.7	38.9	57.9	64.6	77.8	57.4	73.4	66.5	55.4

综上所述，针对现有主流的目标检测算法存在检测精确率低、图像边缘区域分割不全等问题，本文提出了一种基于 Mask RCNN 模型的感兴趣区域池化算法，其能克服复杂背景的干扰，改善边缘区域图像分割的锯齿现象，适应图像边缘区域的尺度形状变化，也使目标的定位更准确. 在数据集 COCO(2014)上，将本文算法与 RoI 池化算法和 RoI Align 池化算法进行了对比分析. 结果表明，相比于另外两种池化算法，本文算法的 mAP 值分别提高了 2.96 个百分点和 0.92 个百分点，AP50 值分别提高了 2.65,0.76 个百分点，AP75 值分别提高了 2.98,0.87 个百分点，分类损失值降低了 0.018 和

0.006，边界框回归损失值降低了0.302和0.008. 因此，本文算法在精确率和稳定性方面具有优异的性能，且具有良好的鲁棒性，能以较少的时间花费换取更高的目标检测与图像分割精确率以及更低的分类损失和边界框回归损失，达到令人满意的目标检测与图像分割效果.

参 考 文 献

[1] BORJI A, CHENG M M, JIANG H Z, et al. Salient Object Detection: A Benchmark [J]. IEEE Transactions on Image Processing, 2015, 24(12): 5706-5722.

[2] 周炫余，刘娟，卢笑，等. 一种联合文本和图像信息的行人检测方法 [J]. 电子学报，2017，45(1)：140-146. (ZHOU X Y, LIU J, LU X, et al. A Method for Pedestrian Detection by Combining Textual and Visual Information [J]. Acta Electronica Sinica, 2017, 45(1): 140-146.)

[3] REDMON J, DIVVALA S, GIRSHICK R, et al. You Only Look Once: Unified, Real-Time Object Detection [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 779-788.

[4] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single Shot Multibox Detector [C]//Computer Vision-ECCV. Berlin: Springer, 2016: 21-37.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks [J]. Advances in Neural Information Processing Systems, 2012, 25: 1106-1114.

[6] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2014: 580-587.

[7] GIRSHICK R. Fast RCNN [C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE, 2015: 1440-1448.

[8] REN S Q, HE K M, 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] HE K M, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN [C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE, 2017: 2961-2969.

[10] 王森，杨克俭. 基于双线性插值的图像缩放算法的研究与实现 [J]. 自动化技术与应用，2008，27(7)：44-45. (WANG S, YANG K J. An Image Scaling Algorithm Based on Bilinear Interpolation with VC++ [J]. Techniques of Automation and Applications, 2008, 27(7): 44-45.)

[11] 陈泽，叶学义，钱丁炜，等. 基于改进 Faster R-CNN 的小尺度行人检测 [J]. 计算机工程，2020，46(9)：226-232. (CHEN Z, YE X Y, QIAN D W, et al. Small-Scale Pedestrian Detection Based on Improved Faster R-CNN [J]. Computer Engineering, 2020, 46(9): 226-232.)

[12] 石杰，周亚丽，张奇志. 基于改进 Mask RCNN 和 Kinect 的服务机器人物品识别系统 [J]. 仪器仪表学报，2019，40(4)：216-228. (SHI J, ZHOU Y L, ZHANG Q Z. Service Robot Item Recognition System Based on Improved Mask RCNN and Kinect [J]. Chinese Journal of Scientific Instrument, 2019, 40(4): 216-228.)

[13] OTSU N. A Threshold Selection Method from Gray-Level Histograms [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66.

[14] ZHANG Z. Proficient in Matlab Digital Image Processing and Recognition [M]. Beijing: Posts & Telecom Press, 2013: 1-186.

[15] PANDA J, MEHER S. An Efficient Image Interpolatio Using Edge-Error Based Sharpening [C]//2020 IEEE 17th India Council International Conference (INDICON). Piscataway, NJ: IEEE, 2020: 1-6.

[16] VINCENT O R, FOLORUNSO O. A Descriptive Algorithm for Sobel Image Edge Detection [C]//Proceedings of Informing Science & IT Education Conference (InSITE). [S.l.]: Informing Science Institute, 2009: 97-107.

[17] 李跃. 基于边缘定向的图像插值算法研究 [D]. 广州：广东工业大学，2015. (LI Y. Edge-Oriented Image Interpolation [D]. Guangzhou: Guangdong University of Technology, 2015.)

[18] 李鑫然，李书琴，刘斌. 基于改进 Faster R_CNN 的苹果叶片病害检测模型 [J]. 计算机工程，2021，47(11)：298-304. (LI X R, LI S Q, LIU B. Apple Leaf Disease Detection Model Based on Improved Faster R_CNN [J]. Computer Engineering, 2021, 47(11): 298-304.)

(责任编辑：韩 啸)