doi:10.11887/j.cn.202104018 http://journal.nudt.edu.cn

融合神经网络与超像素的候选区域优化算法*

王春哲[1,2],安军社[1],姜秀杰[1],邢笑雪[3],崔天舒[1,2]
(1. 中国科学院国家空间科学中心 复杂航天系统电子信息技术重点实验室,北京 100190;
2. 中国科学院大学,北京 100049;3. 长春大学 电子信息工程学院,吉林 长春 130022)

摘 要:为解决目标检测中候选区域召回率低的问题,提出融合神经网络与超像素的目标候选区域算法。该算法利用神经网络提取更能清楚表达目标边界的特征,并使用聚类、相似性等策略,计算每个滑动窗口所含有的边缘信息量;将待测图像使用简单线性迭代聚类算法分割成若干个超像素,并利用超像素的空间位置、完整性、相邻超像素间的对比度信息,计算各个超像素的显著性得分及每个滑动窗口的显著性得分;根据每个滑动窗口的边缘信息及显著性得分筛选滑动窗口。在PASCAL VOC 2007测试集上进行对比实验,其实验结果表明:所述算法能够快速产生定位质量高的候选区域。

关键词:计算机视觉;目标检测;候选区域;卷积神经网络;超像素
中图分类号:TP394.1 **文献标志码**:A **文章编号**:1001-2486(2021)04-145-11

Region proposals optimization algorithm combining neural networks and superpixels

WANG Chunzhe[1,2], *AN Junshe*[1], *JIANG Xiujie*[1], *XING Xiaoxue*[3], *CUI Tianshu*[1,2]
(1. Key Laboratory of Electronics and Information Technology for Space Systems, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China; 3. School of Electronic and Information Engineering, Changchun University, Changchun 130022, China)

Abstract: In order to solve the low recall problem of the region proposals in object detection, the object region proposals algorithm, which combines neural networks and superpixels, was proposed. The edge features, which can be represented clearly by neural networks, were extracted from the images to be detected, and the score of edge information for per sliding window was computed by the strategy of edge clustering and the affinities between the edge groups. The several superpixels of this images were obtained by simple linear iterative clustering algorithm, and the salient object score of a superpixel was calculated using the location, integrity of this superpixel and the contrast with neighbors. The salient objects score of per sliding window was received by these saliency scores of superpixels according to the Euler distance strategy between the sliding window and these superpixels. The region proposals were determined by two components including edge information scores and salient object scores. The comparative experiments were conducted in PASCAL VOC 2007 test set, and the experiment results show that the proposed algorithm can fast generate a set of region proposal with higher localization.

Keywords: computer vision; object detection; region proposals; convolutional neural networks; superpixels

在诸如目标检测、目标跟踪等计算机视觉任务中,候选区域算法有着广泛的应用。所谓候选区域,即使用目标的颜色、纹理等信息寻找图像中更可能出现的目标的区域框[1]。

在目标检测及跟踪等任务中,需要将图像中的目标进行识别与定位。解决这一任务的传统策略是在图像中密集采样滑动窗口,并判别每个滑动窗口是否含有目标。由于该范式下生成的滑动窗口质量不高,因此需要训练复杂的分类器,浪费了计算资源[2-3]。在文献[1-3]中指出,仅在单尺度下,每张图像需要处理$10^4 \sim 10^5$个滑动窗口,而且当前的目标检测要求检测算法处理不同尺度及不同宽高比下的目标,极大地增加了算法的复杂度。

使用候选区域算法能够有效提高目标的检测效率,如在基于快速区域卷积神经网络(Fast Regions with Convolutional Neural Network, Fast RCNN)的检测算法中,使用选择性搜索(Selective

* **收稿日期**:2019-12-18
基金项目:国家自然科学基金资助项目(61805021)
作者简介:王春哲(1989—),男,吉林松原人,博士研究生,E-mail:wangchunzhe163@sina.com;
姜秀杰(通信作者),女,研究员,博士,博士生导师,E-mail:jiangxj@nssc.ac.cn

Search,SS)算法[4]生成大约 2 000 个候选框;在 Faster RCNN 中,使用候选区域网络(Region Proposals Network, RPN)生成大约 800 个候选框[5]。当前主流候选区域算法主要有 Objectness[6],BING[7]及 Edge Boxes[8]。

随着深度神经网络的不断发展,其已经在目标检测、图像哈希(Image Hashing,IH)、图像细分类、视觉描述与生成、视觉问答等方面有着广泛的应用[9]。特别地,文献[10]使用循环神经网络作为代理来构建哈希函数以及序列化学习策略(Sequential Learning Strategy,SLS)来完成图像哈希;文献[11]则通过神经网络提出一种细粒度的视觉 - 文本(Visual-Textual,VT)表达学习方法来完成图像的细分类。

目标的边缘和边界常被定义为具有目标的语义信息[12]。Edge Boxes 通过统计滑动窗口中出现目标的边缘信息量来确定候选区域,但由于 Edge Boxes 仍使用的是传统的边缘生成算法,不能够准确地描述目标的边界,具有一定的局限性[3]。由于卷积神经网络(Convolutional Neural Networks,CNNs)通过模拟人类的感知系统,通过自适应学习方式能够更准确地描述目标的边缘,生成更富有语义信息的边缘特征,有助于提高目标候选区域的质量。

目标显著性[13-17],是在图像的多尺度层面统计图像中目标与背景的对比度、形状等信息,通过合理的数学模型来模拟人类视觉感知系统,快速地将目标从背景中区别出来。在视频分类、图像细分类、显著性目标分割等领域有着广泛的应用。

文献[18]从运动学的角度,将视频帧分成显著性区域和非显著性区域,并使用不同的网络分别对显著性、非显著区域建模以达到视频分类的目的。文献[19]使用了一种全局平均池化(Global Average Pooling,GAP)层的神经网络,称之为显著性提取网络(Saliency Extraction Network,SEN)来提取每张图像的显著性信息,并配合检测框架完成图像的细分类。此外,文献[20]联合了目标显著性的先验知识,精调显著性图及语义分割数据的预训练策略来完成显著性分割任务。

在目标显著性检测中,常用超像素算法提取目标信息特征。由于自然图像具有高度结构化特性[12],若将能够描述图像局部信息的超像素引入候选区域算法,可有效提高候选区域的召回率。

本文从神经网络、目标显著性两个线索来研究目标的候选区域算法。使用深度卷积神经网络提取更能表达目标边界的边缘特征;利用超像素的空间位置、完整性及相邻超像素间的对比度策略来描述每个超像素的显著性得分;最后统计每个滑动窗口中含有目标的边缘信息量及包含超像素的显著性得分,筛选滑动窗口。

1 卷积边缘特征与目标显著性

所提算法主要包括三部分:①边缘特征图的生成、边缘点聚合及边缘簇权重的计算;②超像素的显著性得分;③筛选滑动窗口。首先,使用丰富卷积特征(Richer Convolutional Features,RCF)网络生成富有语义信息的卷积边缘特征图,并结合边缘点聚类获取边缘簇、边缘簇间的相似性等策略获取每个边缘簇权重;然后,在整张图像上使用简单线性迭代的聚类(Simple Linear Iterative Clustering, SLIC)算法将图像分割成若干图像块,并利用相邻超像素间颜色直方图的卡方距离(Chi-Square Distance,CSD)、超像素的空间位置及完整性等策略,统计每个滑动窗口的显著性得分;最后,根据每个滑动窗口含有的边缘信息得分、显著性得分,筛选滑动窗口,确定候选区域。其算法结构如图 1 所示。

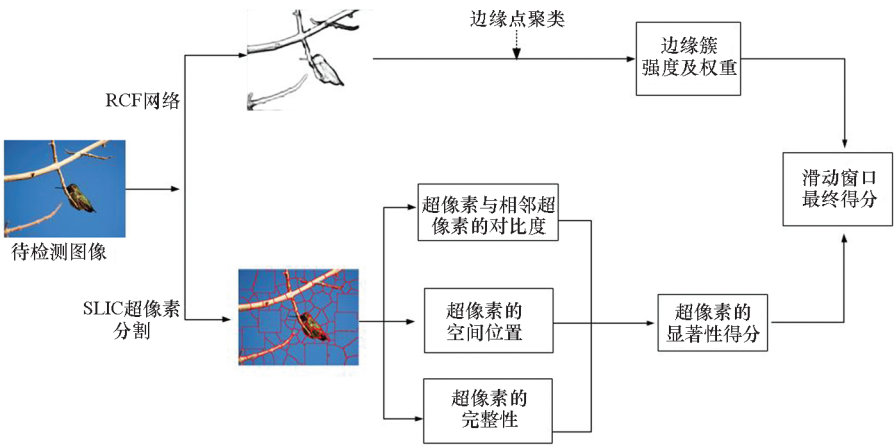


图 1 所提算法的实现框图

Fig. 1 Block diagram of the proposed algorithm

1.1　卷积边缘特征

RCF 结构的骨架是 VGG－16 网络,由卷积层、concat 层和 cross-entropy 层组成[12],其结构见文献[3,12]。为更直观地说明卷积边缘特征,在边缘检测数据集 BSD500 任意选取一张原始图像(见图 2(a)),及使用几种边缘检测算子生成的边缘特征图(见图 2(c)～图 2(e)),图 2(b)为真实的边缘特征图。由图 2(c)可知,传统边缘检测算子 Canny 生成的边缘特征,目标轮廓较差,目标语义信息较弱;由图 2(e)可知,使用 RCF 网络生成的边缘特征,目标轮廓更加明显,目标语义信息丰富。丰富的语义信息可使用相对简单的分类器进行目标分类,有效降低了算法的复杂度。

给定一张图像 X(见图 2(a)),使用 RCF 网络生成边缘特征图 $\hat{X}$(见图 2(e))。为更有效地统计边缘点,将 RCF 网络生成的边缘特征图 $\hat{X}$ 上具有一定相似性的边缘点进行聚类。因此,$\hat{X}$ 将会聚合成若干个边缘簇 s。边缘簇 s 算法详见文献[3]。

给定任意一个边缘簇 s,任取 s 中的任意一个边缘点 p,用四维向量$[m_p,\theta_p,x_p,y_p]$表示。向量中的参数分别为边缘点 p 的边缘强度、方向角及空间位置坐标。根据边缘点 p 可确定边缘簇 s 的空间位置。

$$\begin{cases} x_s = \sum_{p\in P} m_p x_p \\ y_s = \sum_{p\in P} m_p y_p \end{cases} \tag{1}$$

其中,P 是 s 中所有边缘点组成的集合。

因此,边缘簇 s 的方向角 θ_s 为:

$$\theta_s = \arctan\left(\frac{\Delta y}{\Delta x}\right) \tag{2}$$

式中,$\Delta x = \sum_{p\in P} m_p\cos\theta_p$,$\Delta y = \sum_{p\in P} m_p\sin\theta_p$。

因此,一张特征图 $\hat{X}$ 会生成若干个边缘簇,用集合 T 表示。从集合 T 中任取两个边缘簇,记为 t_i 与 t_j。则利用 t_i 的方向角 θ_i、t_j 的方向角 θ_j 及两边缘簇重心连线之间的方向角 θ_{ij},确定两边缘簇之间的相似度。

$$a(t_i,t_j) = |\cos(\theta_i-\theta_{ij})\cos(\theta_j-\theta_{ij})|^{\gamma} \tag{3}$$

式中,γ 是调整方向角的变化对相似性 $a(t_i,t_j)$ 的敏感程度的参数[8],鉴于 Edge Boxes 算法的取值,取 $\gamma=2$。

给定滑动窗口 b 及边缘簇 t_k,使用参数 $w_b(t_k)\in[0,1]$来描述 t_k 是否被滑动窗口 b 包围。若 $w_b(t_k)=0$,表明滑动窗口 b 与边缘簇 t_k 不相交;若 $w_b(t_k)=1$,表明 t_k 完全在 b 中[8]。而对于其他的边缘簇 t_i,采用以下策略来确定参数 $w_b(t_i)$。

步骤 1:建立一个集合 T_b 作为与滑动窗口 b 的边界完全相交的边缘簇。若边缘簇 $t_i\in T_b$,则 $w_b(t_i)=0$。

步骤 2:对于边缘簇 t_i,取 t_i 中任意一个边缘点,位置坐标为 $\bar{x}_i$,如果 $\bar{x}_i\notin b$,则 $w_b(t_i)=0$;若 $\bar{x}_i\in b$ 且 $t_i\notin T_b$,则

$$w_b(t_i) = 1-\max_E\prod_j^{|E|-1} a(e_j,e_{j+1}) \tag{4}$$

式中,E 表示由 $e_1=t_j\in T_b$ 到 $e_{|E|}=t_i$ 的路径。由式(4)知,$\max_E\prod_j^{|E|-1} a(e_j,e_{j+1})$ 描述在集合 T_b 中寻找与 t_i 最相似的一个边缘簇,因此边缘簇 t_i 的权重 $w_b(t_i)$ 为 $1-\max_E\prod_j^{|E|-1} a(e_j,e_{j+1})$。所以,由滑动窗口 b 包含的边缘簇 t_i 的强度 m_i 及对应的权重值 $w_b(t_i)$ 定义该窗口的边缘信息得分[2-3]。

$$h_b = \frac{\sum_i [w_b(t_i)m_i]}{2(b_w+b_h)^{\varepsilon}} \tag{5}$$

其中:b_w 与 b_h 是窗口 b 的高和宽;ε 为调节 h_b 对滑动窗口大小的敏感度的参数[8],鉴于 Edge Boxes 算法的策略,取 $\varepsilon=1.5$;$m_i=\sum_{p\in t_i} m_p$。

(a) 原图　(b) 真实边缘特征　(c) Canny　(d) 结构化的边缘　(e) RCF
(a) Original image　(b) Real Edge Features　(d) Structured edges

图 2　几种边缘特征图的对比
Fig. 2　Comparisons of several edge features

1.2 显著性得分

1.2.1 超像素

图 3 为使用 SLIC 算法[21]分割的超像素示意图。图 3 中每一个闭合区域为一个超像素。

从图 3 可知:①任意一个超像素块与相邻超像素块颜色的对比度较大;②靠近图像中心的超像素更可能含有目标;③在图像边缘像素个数越多的区域更可能成为背景,如图 3 中的 br。含有目标区域的 bc 无边缘像素,br 含有相对较多的边缘像素。为方便起见,把包含图像边缘像素的数目作为指标来定义一个超像素的完整性。

对于一张图像 X,其中心坐标为(x_0,y_0)。首先使用 SLIC 算法将其过分割成 L 个超像素$\{c_i\}$ $(i=1,\cdots,L)$。SLIC 算法对不同图源的图像具有通用性,其算法流程及初值选取情况如下所示。

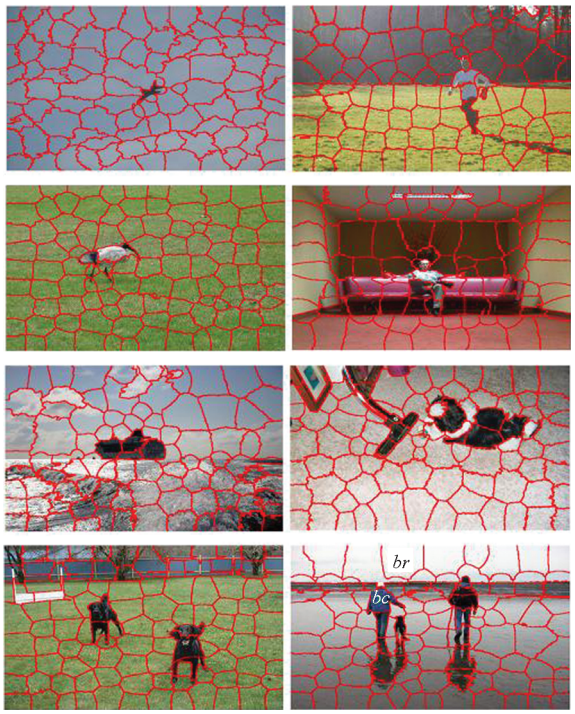


图 3 SLIC 算法生成超像素

Fig. 3 Superpixels generated from SLIC algorithm

步骤 1:初始化。首先在图像 X 上进行采样,采样步长为 S,同时将聚类中心初始化为 $\boldsymbol{C}_k^{(0)}=[l_k,a_k,b_k,x_k,y_k]^{\mathrm{T}}$。其中:$l_k,a_k,b_k$ 为颜色空间(LAB color space, LAB)对应的像素值;x_k,y_k 为聚类中心 $\boldsymbol{C}_k^{(0)}$ 的坐标值;$S=\sqrt{N/K}$,N 为像素个数、K 为聚类中心的个数。然后将聚类中心 $\boldsymbol{C}_k^{(0)}$ 移动到该聚类中心 3×3 邻域内最低梯度的位置 $\boldsymbol{C}_k^{(1)}$,并令图像 X 的每个像素 $\boldsymbol{C}_k^{(1)}$ 设置标签 $l(i)=-1$,及距离 $d(i)=\infty$。

步骤 2:像素的分配。对于聚类中心 $\boldsymbol{C}_k^{(1)}$ 以及在 $\boldsymbol{C}_k^{(1)}$ 周围的 $2S\times 2S$ 区域内的每个像素 i,计算聚类中心 $\boldsymbol{C}_k^{(1)}$ 与像素 i 的欧氏距离 D,如果 $D<d(i)$,设置 $d(i)=D$。通过像素点 i 与各聚类中心最小的欧氏距离原则,为每个像素点 i 分配对应的聚类中心,记 $l(i)=k$。

步骤 3:更新。计算第 $m+1$ 次迭代后的聚类中心 $\boldsymbol{C}_k^{(m+1)}$,计算残差 $\boldsymbol{E}=\boldsymbol{C}_k^{(m+1)}-\boldsymbol{C}_k^{(m)}$,直到 $\boldsymbol{E}\leqslant threshold$。

与超像素 c_i 相邻的超像素记为$\{n_j\}$ $(j=1,\cdots,L_i)$,L_i 为与 c_i 相邻的超像素的数目。为衡量超像素 c_i 的显著性,首先,将图像 X 的每个通道的像素平均分成 $nbin$ 个区间:$\left[\frac{256}{nbin}(k-1),\frac{256}{nbin}(k)-1\right](k=1,\cdots,nbin)$。然后分别统计超像素 c_i 与 n_j 中像素落入每个区间内的数目,即$\{h_\beta^k(c_i)\}$和$\{h_\beta^k(n_j)\}$ $(\beta=r,g,b)$。则得到:

R 通道颜色直方图的卡方距离

$$\chi(h_r(c_i),h_r(n_j))=\sum_{k=1}^{nbin}\frac{[h_r^k(c_i)-h_r^k(n_j)]^2}{h_r^k(c_i)+h_r^k(n_j)} \tag{6}$$

G 通道颜色直方图的卡方距离

$$\chi(h_g(c_i),h_g(n_j))=\sum_{k=1}^{nbin}\frac{[h_g^k(c_i)-h_g^k(n_j)]^2}{h_g^k(c_i)+h_g^k(n_j)} \tag{7}$$

B 通道颜色直方图的卡方距离

$$\chi(h_b(c_i),h_b(n_j))=\sum_{k=1}^{nbin}\frac{[h_b^k(c_i)-h_b^k(n_j)]^2}{h_b^k(c_i)+h_b^k(n_j)} \tag{8}$$

为考虑计算成本,取 $nbin=8$。则超像素 c_i 与 n_j 直方图的卡方距离为:

$$d(c_i,n_j)=\frac{1}{3}\sum_{\omega=r,g,b}\chi(h_\omega(c_i),h_\omega(n_j)) \tag{9}$$

常使用与相邻超像素 n_j 间的卡方距离 $d(c_i,n_j)$、超像素 c_i 的空间位置 $g(x_{c_i},y_{c_i})$ 及完整性 $q(u)$ 来描述超像素 c_i 的显著性[16]。因此,超像素 c_i 的显著性[16] 为:

$$f(c_i)=\sum_{j=1}^{N_i}w_{ij}p(d(c_i,n_j))\cdot g(x_{c_i},y_{c_i})\cdot q(u) \tag{10}$$

式中,w_{ij} 是给对应的 $p(d(c_i,n_j))$ 赋予的权重值,其值的大小为:

$$w_{ij}=\frac{count(n_j)}{\sum_{j=1}^{L_i}count(n_j)} \tag{11}$$

式中,$count(o)$表示含有 o 的个数。

$$p(\varphi) = -\lg(1-\varphi) \tag{12}$$

式中,函数 $p(\varphi)$ 目的是保证输入为 φ 时,输出为正值。

由此可知,超像素 c_i 与 n_j 直方图的卡方距离越大,$p(d(c_i,n_j))$ 值也将越大。$g(x_{c_i},y_{c_i})$ 描述超像素 c_i 的中心 (x_{c_i},y_{c_i}) 与图像中心 (x_0,y_0) 归一化的空间距离:

$$g(x_{c_i},y_{c_i}) = \exp\left[-\frac{(x_{c_i}-x_0)^2}{2\delta_x^2}-\frac{(y_{c_i}-y_0)^2}{2\delta_y^2}\right] \tag{13}$$

式中,δ_x^2 和 δ_y^2 分别是超像素 c_i 的方差,其数值分别为图像 X 的宽与高的 1/3。由式(13)可以知道,越靠近图像中心的超像素 c_i 的显著性越强。

如前所述,一个完整的超像素应是一个闭合(连通)区域,如 bc。而对于超像素 br,由于位于图像的边缘,并不是一个完整的超像素。因此,引入描述超像素的完整性参数 $q(u)$。

$$q(u) = \begin{cases} \exp\left(-\dfrac{\mu}{\lambda\times E}\right), \dfrac{\mu}{E} \leqslant \eta \\ 0, \text{其他} \end{cases} \tag{14}$$

其中:μ 为超像素 c_i 所包含在图像边缘像素的数目;E 为图像 X 中所有边缘像素的数目;λ 用来控制 E 对 $q(u)$ 的影响强度;η 是一个阈值。鉴于文献[16]的取值,取 $\lambda=0.05$,$\eta=0.07$。

由式(14)知,当 $\mu=0$ 时,$q(u)=1$,表明超像素 c_i 不在图像的边缘;当 $\mu\neq 0$ 时,$q(u)$ 是一个取值范围在[0,1]之间的正数。

由此可知,超像素 c_i 与所有相邻的超像素 n_j 间的显著性 $f(c_i)$ 的值越大,超像素 c_i 包含目标的可能性越大。

1.2.2　滑动窗口的显著性得分

给定滑动窗口 b,用四维向量 $[b_x,b_y,b_w,b_h]$ 表示。为确定滑动窗口 b 包含超像素 c_i 的程度,首先,计算滑动窗口 b 的中心位置坐标 (b_{mx},b_{my}):

$$\begin{cases} b_{mx} = b_x + \dfrac{b_w}{2} \\ b_{my} = b_y + \dfrac{b_h}{2} \end{cases} \tag{15}$$

其中:b_x,b_y 分别为滑动窗口 b 左上角的位置坐标;b_w,b_h 分别为滑动窗口 b 的宽与高。

然后计算图像 X 上所有超像素的中心位置坐标 (x_{c_i},y_{c_i}) $(i=1,\cdots,L)$。确定超像素 c_i 的中心位置坐标的算法,见算法 1。

算法 1　超像素 c_i 的中心位置策略

Alg. 1　Strategy of the center position of superpixel c_i

输入:图像 X 的宽 N、高 M,及超像素分布图 $F(X$ 中每个像素点属于某个超像素),其中 $F(j,k)=i$ 表示 X 中的第 j 行、第 k 列属于超像素 c_i $(i=1,\cdots,L)$

输出:超像素 c_i 的中心位置坐标 (x_{c_i},y_{c_i}) $(i=1,\cdots,L)$

步骤 1:初始化 c_i 的中心位置横坐标 x_{c_i} $(x_{c_i}=0)$、纵坐标 y_{c_i} $(y_{c_i}=0)$ 及属于 c_i 横纵坐标的像素数目 $labelx$、$labely$ $(labelx=0, labely=0)$;

步骤 2:遍历整张图像,统计属于 c_i 的横坐标 j 及纵坐标 k,寻找属于 c_i 的像素点位置,即

如果 $(F(j,k)==i)$,则 $x_{c_i}=x_{c_i}+j$, $y_{c_i}=y_{c_i}+k$, $labelx=labelx+1$, $labely=labely+1$;

步骤 3:计算每一个超像素 c_i 的中心位置:$x_{c_i}=x_{c_i}/labelx$; $y_{c_i}=y_{c_i}/labely$

目标显著性得分情况如图 4 所示。图 4 中,超像素 2、5 被滑动窗口完全包围,超像素 1、3、4 被滑动窗口部分包围。为确定滑动窗口 b 包含超像素 c_i 的程度,使用 b 的中心位置 (b_{mx},b_{my}) 与超像素 c_i 中心位置 (x_{c_i},y_{c_i}) 之间的欧氏距离 $dis(b,c_i)$ 是否满足:

$$dis(b,c_i) \leqslant \delta \tag{16}$$

如果满足式(16),则表示滑动窗口 b 包含超像素 c_i。式(16)中:$\delta=\tau\sqrt{\left(\frac{b_{mx}}{2}\right)^2+\left(\frac{b_{my}}{2}\right)^2}$,$0<\tau<1$,$\tau$ 是用来调节 b 包含 c_i 的紧密程度。τ 值越小,表明包含越紧密,见图 4 中标记“2、5”的超像素;τ 值越大,表明 b 可能只包含 c_i 的部分区域,见图 4 中标记“1”的超像素。

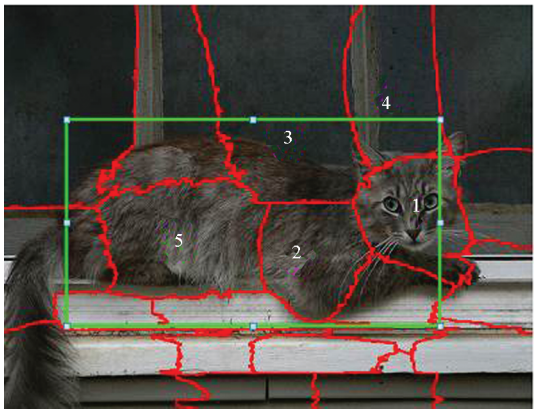


图 4　目标显著性得分示意图

Fig. 4　Illustration of object saliency scores

$$dis(b,c_i) = \sqrt{(b_{mx}-x_{c_i})^2+(b_{my}-y_{c_i})^2} \tag{17}$$

使用 b 中包含所有超像素 $\{c_\psi\}$ 的显著性得分作为滑动窗口 b 的显著性得分：

$$S_{\text{sal}}(b) = \sum_{\psi=1}^{N_b} f(c_\psi) \tag{18}$$

式中，N_b 表示 b 中含有的超像素的个数。

1.3 筛选候选框

将上述获得滑动窗口的边缘信息得分 h_b 以及显著性得分 $S_{\text{sal}}(b)$，并给予恰当权重值，作为此滑动窗口 b 含有目标的得分。

$$h_b^{(\text{rev})} = h_b + \alpha \cdot S_{\text{sal}}(b) \tag{19}$$

式中，参数 α 用于调整 $S_{\text{sal}}(b)$ 在 $h_b^{(\text{rev})}$ 中的重要程度。

最后，按照每个候选区域 b 的得分从高到低排列，选取指定个数的候选区域进行后续的目标检测。

2 数据分析与性能比较

2.1 数据集选取、评价指标及 RCF 网络的训练

在目标检测领域中广泛使用 PASCAL VOC 2007 数据集进行测试。该数据集由训练集、验证集与测试集组成。包含 20 类、共 24 640 个目标，分布在 9 963 张图像中。

使用召回率来衡量候选区域算法的性能，召回率是描述候选区域算法生成有效的目标候选框占所有目标候选框的比重[3]。

借鉴文献[12]中关于 RCF 网络的训练方法，即直接使用 Liu 等训练好的 RCF 网络[12]，在 PASCAL VOC 2007 数据集中获取对应每张图像的边缘特征。关于 RCF 网络超参数的设置见文献[12]。

2.2 参数确定

根据文献[8]所述 Edge Boxes 算法确定参数的策略，在 PASCAL VOC 2007 验证集上进行实验。本文算法 $h_b^{(\text{rev})}$ 共有权值 τ、α 共两个参数。选取 τ 的值为 0.4，0.6 及 0.8；α 的值为 0，0.2 及 0.5 进行测试。表 1 列出候选框数为 1 000、参数 α、τ 的几种不同组合，在不同交并比 *IoU* 下的召回率。

由表 1 可知：当 $\alpha=0.2$、$\tau=0.8$、*IoU* = 0.7 及 0.9 时，PRPA4 取得最高的召回率。在高 *IoU* 的取值下，获得召回率值最高的参数组合，表明候选框与标注候选框重合面积越大，其定位性能越好，因此选择 PRPA4，即参数 $\alpha=0.2$、$\tau=0.8$。

表 1 所提算法在 VOC 2007 验证集的召回率

Tab. 1 Recalls of proposed algorithm on VOC 2007 validation dataset

算法	召回率/%			参数	
	IoU = 0.5	*IoU* = 0.7	*IoU* = 0.9	α	τ
PRPA1	92.50	77.85	10.67	0	任意值
PRPA2	92.48	77.82	10.64	0.2	0.4
PRPA3	92.50	77.85	10.43	0.2	0.6
PRPA4	92.47	77.88	10.86	0.2	0.8
PRPA5	92.44	77.87	10.56	0.5	0.4
PRPA6	92.47	77.83	10.50	0.5	0.6
PRPA7	92.39	77.71	10.80	0.5	0.8

2.3 数据性能分析

为论证所述算法的性能，选取近几年来较流行的算法如：SS[4]、Object-ness[6]、BING[7]、Edge Boxes[8]、CPMC[22]、Randomized Prim′s[23]、Geodesic[24]、MCG[25]、Rantalankila[26]，在 VOC 2007 测试集上进行对比实验。

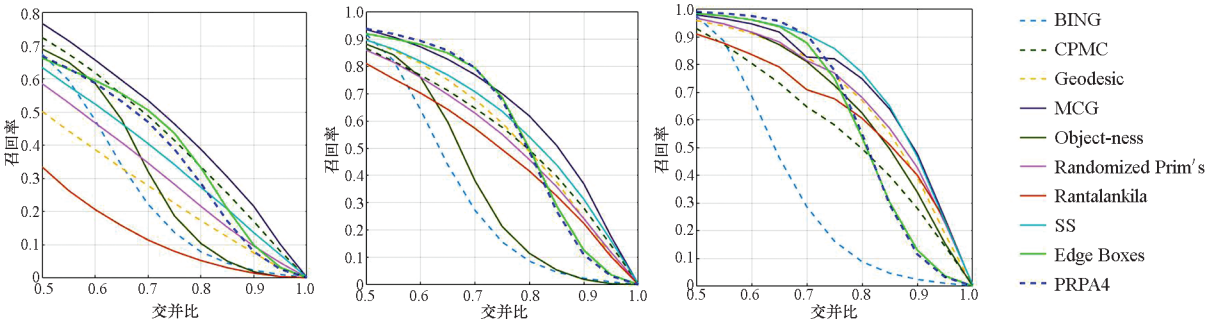
固定候选区域数目，研究各种算法在不同 *IoU* 下的召回率，如图 5 所示。当取得较少候选框数 100 时，MCG 及 CPMC 算法性能略高于所提算法 PRPA4，但 PRPA4 性能却优于近年主流算法 SS[4]；当候选框数为 1 000 及 10 000 时，交并比为 0.5 ~ 0.7 时，PRPA4 的召回率最高，这表明所提算法能够生成高质量的候选框。

接下来，固定交并比，研究 10 种算法在不同候选框数目下的召回率，如图 6 所示。从图 6(a)及图 6(b)可以看出，当交并比为 0.5、0.7 时，随着候选框数目的不断增加，PRPA4 的召回率不断升高，最终可获得最高的召回率。图 6(c)为各算法在交并比取[0.5,1.0]时的平均召回率。由图 6(c)可知，随着候选框数目的增加，所提算法 PRPA4 的平均召回率(Average Recall，AR)逐渐超过 Edge Boxes 算法，其整体性能表现优越。

2.4 候选区域算法对不同尺寸目标性能的影响

在 VOC 2007 测试集中测试了 PRPA4 对不同尺寸目标性能的影响。使用目标区域的面积来衡量不同尺寸目标，即：如果目标候选框的面积 *BoxArea* ≤ 32 像素 × 32 像素，则为小尺寸目标；如果 *BoxArea* > 32 像素 × 32 像素，则为较大尺寸目标。

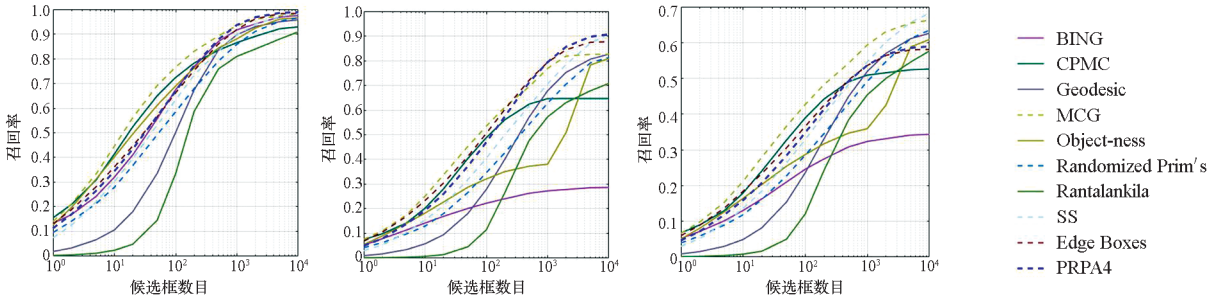
选取 1 000 个候选框，以及常用的 *IoU* 为 0.5、0.6 及 0.7 进行实验，其结果见表 2。



(a) 100 个候选框　(b) 1 000 个候选框　(c) 10 000 个候选框
(a) 100 proposals　(b) 1 000 proposals　(c) 10 000 proposals

图 5　交并比与召回率的关系

Fig. 5　Recall versus *IoU* threshold



(a) 交并比为 0.5　(b) 交并比为 0.7　(c) 交并比为[0.5,1]
(a) *IoU* is 0.5　(b) *IoU* is 0.7　(c) *IoU* is between [0.5,1]

图 6　候选框数目与召回率的关系

Fig. 6　Recall versus number of proposals

表 2　10 种候选区域算法不同尺寸目标的性能

Tab. 2　Performance of different sizes objects using 10 region proposal algorithm

算法	小尺寸目标			较大尺寸目标			*t*/s
	IoU = 0.5	*IoU* = 0.6	*IoU* = 0.7	*IoU* = 0.5	*IoU* = 0.6	*IoU* = 0.7	
Object-ness	0.443 9	0.243 9	0.087 8	0.940 5	0.853 3	0.487 6	3.000 0
BING	0.551 2	0.258 5	0.102 4	0.961 4	0.715 1	0.358 1	0.200 0
CPMC	0.551 2	0.336 6	0.185 4	0.930 9	0.858 6	0.767 9	250.000 0
SS	0.775 6	0.536 6	0.380 5	0.955 2	0.904 7	0.833 6	10.000 0
Edge Boxes	0.717 1	0.507 3	0.292 7	0.964 6	0.941 9	0.879 7	0.250 0
Rantalankila	0.287 8	0.185 4	0.087 8	0.898 5	0.824 8	0.726 6	10.000 0
Rand. Prim′s	0.609 8	0.390 2	0.229 3	0.923 6	0.862 3	0.762 2	1.000 0
MCG	0.751 2	0.565 9	0.346 3	0.968 8	0.937 3	0.873 7	30.000 0
Geodesic	0.429 3	0.268 3	0.102 4	0.945 6	0.889 6	0.795 7	1.000 0
PRPA4	0.765 9	0.604 9	0.395 1	0.973 6	0.951 1	0.885 4	0.549 8

由表 2 可知:对于较大尺寸目标,*IoU* = 0.5、0.6、0.7 时,PRPRA4 均能达到最高的召回率;对于较小尺寸目标,PRPA4 在 *IoU* = 0.6、0.7 时,可获得最高的召回率,在 *IoU* = 0.5 时,略低于 SS 算法的召回率;结合各算法运算时间可知,在处理较大尺寸目标时,PRPA4 能够生成质量最高的目标候选框。

2.5　RCF 网络及显著性得分对候选区域算法的影响

使用 Canny 及 RCF 两种边缘检测算子(均使用参数 $\alpha = 0.2$、$\tau = 0.8$),选定 500 个候选框,在 VOC 2007 验证集上进行测试,实验结果见表 3。

表3 不同检测算子的性能

Tab. 3 Performance of different edge detectors

检测算子	召回率/%			
	$IoU=0.7$		$IoU=0.9$	
Canny	47.69/47.95	↑0.26	5.47/5.36	↓0.11
RCF	69.05/69.22	↑0.17	9.94/10.15	↑0.21

表3中:符号"/"左侧为未引进显著性的召回率;符号"/"右侧为引进显著性的召回率;符号"↑"代表召回率提高;符号"↓"代表召回率下降。

由表3可以看出,Canny算子在$IoU=0.9$时,加入显著性得分后,召回率略有下降(下降了0.11%),在其余的情况下,引入显著性得分均可改善候选区域的质量。

另一方面,在未加入显著性得分时,相较于Canny算子,RCF生成的目标候选框的召回率明显提高。因此,基于卷积神经网络生成的边缘特征图和显著性得分这两部分都有助于提高所生成目标候选框的质量。

2.6 所提算法在Fast RCNN上检测性能的表现

为确定所提算法在检测框架Fast RCNN[27]上的检测性能。选取了Fast RCNN的3种基本模型分别是:Model-S(即CaffeNet)、Model-M(即VGG_CNN_M_1024)、Model-L(即VGG16)。

选取2组对比实验,2 000个候选框在Fast RCNN的3种模型的检测精度见表4。

1)未重训练。选取由SS算法生成的候选框(VOC 2007训练集),分别训练Fast RCNN的3种模型,获得训练参数,并对其他9种候选区域算法生成的候选框进行测试(VOC 2007测试集),其各算法的平均检测度(mean Average Precision,mAP)分别位于表4中符号"/"的左侧。

2)重训练。在10种候选区域算法各自生成的候选框(VOC 2007训练集)上,分别训练Fast RCNN的3种模型,使用训练参数,分别测试各算法在测试集上生成的候选框(VOC 2007测试集),其检测精度位于表4中符号"/"的右侧。

表4中:符号"+"代表检测精度mAP值增加。符号"-"代表mAP值减小。

由表4可知:在检测模型Model-M中,在"未重训练"的情况下,PRPA4的检测精度要优于Edge Boxes算法,这说明PRPA4确实提高了候选区域的质量。在"重训练"的情况下,PRPA4在3种模型中,检测精度均要优于Edge Boxes算法;同时,在Model-M及Model-L模型中,PRPA4均能获得最高的mAP值,这也说明PRPA4能够获得高质量的目标候选区域。

另外,从表4也可发现:像Object-ness、BING、Edge Boxes、PRPA4算法,在Model-S、Model-M、Model-L的3种模型中,"重训练"均能大幅提高目标的检测精度(精度升高的变化范围为1.64%~8.40%)。

表4 2 000个候选框在Fast R-CNN的3种模型的检测精度

Tab. 4 Detection precision of 3 models of Fast R-CNN using 2 000 region proposals

%

算法	Model-S		Model-M		Model-L	
	mAP	Δtrain	mAP	Δtrain	mAP	Δtrain
Object-ness	46.13 / 52.26	+ 6.13	46.31 / 54.71	+ 8.40	57.74/ 65.02	+ 7.28
BING	41.11 / 46.62	+ 5.51	43.03/ 50.26	+ 7.23	55.95/ 62.08	+ 6.13
CPMC	55.65 / 53.23	- 2.42	57.27/ 54.69	- 2.58	65.10 / 63.40	- 1.70
SS	58.21 / 58.21	+ 0.00	60.04 / 60.04	+ 0.00	67.59 /67.59	+ 0.00
Edge Boxes	55.79 /57.77	+ 1.98	57.80 / 60.90	+ 3.10	67.26 /68.90	+ 1.64
Rantalankila	55.81 / 54.49	- 1.32	58.06/ 55.93	- 2.13	64.73 /63.71	-1.02
Rand. Prim's	57.74 / 56.95	- 0.79	60.31/ 58.94	- 1.37	67.40 / 67.10	- 0.30
MCG	58.22 / 58.49	+ 0.27	61.06/ 61.02	- 0.04	68.45 /68.23	- 0.22
Geodesic	57.28 / 56.62	- 0.66	59.40/ 58.67	- 0.73	66.29 / 65.18	- 1.11
PRPA4	55.40 /57.84	+ 2.44	57.83 / 61.33	+ 3.50	66.86 /69.24	+ 2.38

在表4的Model-S模型中,"重训练"的PRPA4的检测精度mAP值要小于MCG算法。为说明此现象的原因,首先观看图5。

由图5可知,当交并比IoU取值为0.8~1.0时,MCG算法生成的候选框要比PRPA4的召回率高,这表明:相比于PRPA4算法,MCG算法生

成的目标候选框和真实的目标标注框有较高的重叠率;当在相对较浅的网络 Model-S 训练时,由于浅层网络不能很好地抓住目标的语义信息,PRPA4 算法生成定位质量相对较差的目标候选框。由于引入了额外的背景信息,其平均检测精度要小于 MCG 算法生成的目标候选框的检测精度。

而随着检测网络的深入,如 Model-M 及 Model-L 模型时,这些网络能够很好地抓住目标的语义信息;且在训练这两个模型的过程中,相对于 MCG 算法,PRPA4 算法生成的目标候选框有较多的正样本(Positive Samples,PS),这将促进两个模型的目标检测准确度。因此在 Model-M、Model-L 模型中,使用"重训练"模式,PRPA4 生成的目标候选框的平均检测精度要高于 MCG 算法。

表 5 列出了在模型 Model-L 下选取 2 000 个候选框,"重训练"模式,各算法在 VOC 2007 测试集上的检测精度。同时,为每个算法给出 20 类目标的 mAP 值。表 5 中,每类目标的最高检测精度值用"加粗"字体标识。

表 5　VOC 2007 测试集中 20 类目标的检测精度

Tab. 5　Detection precision of 20 classes objects on VOC 2007 test dataset

%

算法	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
Object-ness	66.92	72.21	64.27	52.02	37.64	75.50	75.63	78.42	45.99	72.72	60.13	81.08	79.44	71.78	66.26	34.79	64.10	62.88	74.22	64.32	65.02
BING	63.96	68.14	62.58	44.76	43.94	68.12	75.30	78.40	40.86	71.86	62.40	72.60	75.54	67.61	65.09	27.86	61.18	60.66	70.52	60.29	62.08
CPMC	67.63	67.88	61.68	51.25	29.71	77.05	73.76	81.95	35.81	72.33	67.55	79.50	78.65	72.25	63.50	26.90	61.27	63.48	76.50	59.38	63.40
SS	**72.59**	**79.07**	70.79	57.42	39.43	75.54	78.73	78.49	45.58	73.15	68.20	77.34	78.57	74.69	72.60	32.98	68.44	**67.47**	75.25	**65.49**	67.59
Edge Boxes	68.04	78.32	67.63	57.37	**48.24**	80.28	78.37	80.07	49.25	**74.15**	66.72	81.47	80.44	**77.28**	74.71	36.91	**71.87**	66.29	75.91	64.65	68.90
Rantalankila	69.65	69.27	65.64	49.45	29.49	76.04	69.28	**83.66**	37.87	71.11	69.30	76.96	79.84	74.18	63.38	25.15	63.58	65.83	75.54	59.01	63.71
Rand. Prim's	76.28	77.27	69.43	51.44	34.55	78.78	76.14	83.05	44.74	74.09	**70.04**	**82.33**	78.87	77.04	66.10	29.96	64.77	67.04	**78.21**	61.92	67.10
MCG	71.57	77.60	67.54	55.40	44.27	**82.28**	78.60	78.71	**49.65**	73.73	68.76	77.02	80.34	74.64	74.00	36.09	64.23	67.37	77.76	65.12	68.23
Geodesic	67.68	75.93	64.04	52.13	35.15	77.48	77.54	79.18	42.12	73.09	64.40	76.59	79.72	72.70	67.15	29.36	65.04	64.79	75.92	63.65	65.18
PRPA4	69.42	78.86	**71.28**	**58.58**	47.54	81.16	**78.90**	83.64	49.41	74.14	65.28	81.74	**80.68**	75.62	**75.39**	**37.96**	69.93	65.26	74.81	65.27	**69.24**

由表 5 可知:①所提算法 PRPA4 在诸如"bird""boat""car""horse""person""plant"共 6 类目标上性能最好,这表明在遇到上述场景目标时,可优先选用 PRPA4 算法;②与其他 9 种算法相比,所提算法的检测精度为最高值的目标数为 6,远远大于 SS 算法(4 种)、Edge Boxes 算法(4 种)、Randomized Prim's 算法(3 种),这反映所提算法的检测性能有更高的鲁棒性;③所提算法的 mAP 值最高。

2.7　所提算法的运算效率

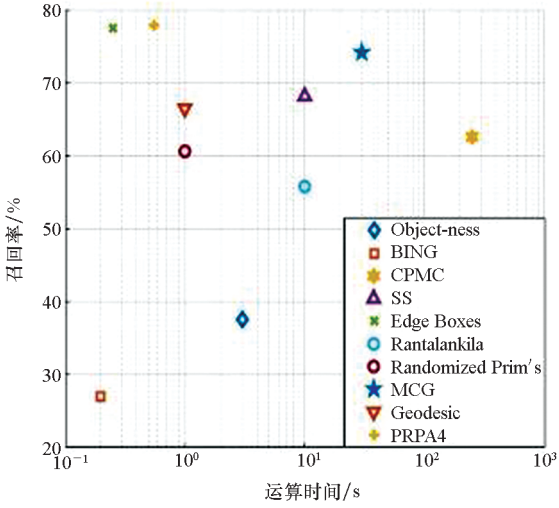
文献[28]使用召回率、候选区域的定位质量(Proposal Localization Quality,PLO)和算法的运算效率(Computational Efficiency,CE)来说明各算法所生成的候选区域的质量。本文绘制了各算法的召回率与运算效率的散点图以及各算法的定位质量与运算效率的散点图,来描述各算法的性能。

通常使用数据集中所有类别的平均最佳重叠率(Mean Average Best Overlap,MABO)衡量候选区域的定位质量。

图 7 为选择 1 000 个候选框时,各候选区域算法的召回率、MABO 以及运算时间的对比图。由图 7(a)可知:BING 算法所需时间最短,但是召回率低;PRPA4 算法所需时间相对较短,但却有最高的召回率。由图 7(b)可知:PRPA4 算法的 MABO 接近 MCG 算法,但运算时间远小于 MCG 算法。因此,所述算法使用较短的时间,就能获得高质量的候选区域。

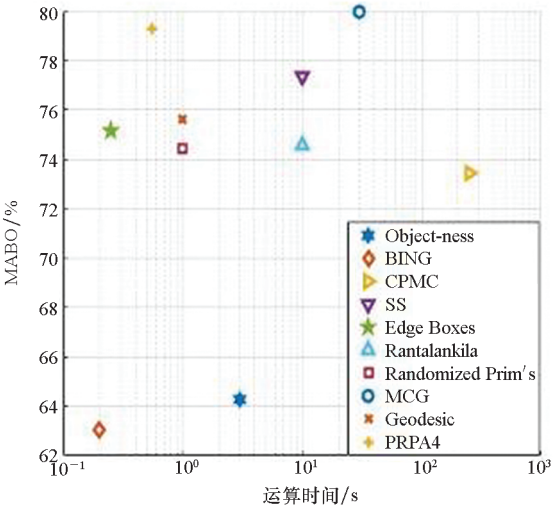
2.8　所提算法的目标检测结果

图 8 列出了各候选区域算法的目标检测结果。从图 8 可以看出:各算法检测出来的"候选框"及精度值均有差别;"候选框"越接近标



(a) 候选区域的召回率与运算时间

(a) Recall of region proposals versus computation time



(b) MABO 与运算时间

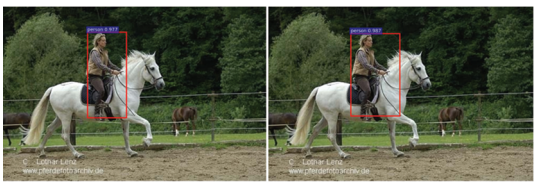
(b) MABO versus computation time

图 7 VOC 2007 数据集上各算法的性能对比

Tab. 7 Performance comparison of region proposal methods on VOC 2007 dataset

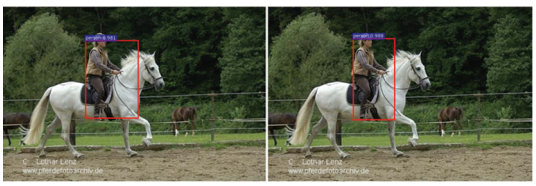
注框,检测精度越高;另外,PRPA4 算法的检测“候选框”更接近真实标注框,检测精度值也更高。

(a) BING (b) CPMC



(c) Edge Boxes (d) Geodesic

(e) MCG (f) Object-ness



(g) Randomized Prim's (h) Rantalankila

(i) SS (j) PRPA4

图 8 各候选区域算法的目标检测结果

Fig. 8 Object detection results of region proposals algorithms

3 结论

本文从卷积神经网络、超像素两方面研究目标候选区域算法。实验结果表明:由卷积神经网络生成的边缘特征具有较高的语义信息,能够更清楚地表达目标的边界,从而提高目标候选区域的质量。使用超像素算法将图像中具有相似属性的像素聚类成同一区域,并从超像素的空间位置、完整性角度统计每个滑动窗口的显著性得分,使得候选区域的召回率提高。

在目标检测框架 Fast RCNN 的检测模型 Model-M 及 Model-L 上,选取 2 000 个候选框,所提算法 PRPA4 的平均检测精度 mAP 分别为 61.33%、69.24%,较 Edge Boxes 算法的 mAP 分别提高了 0.43%、0.34%;同时,由 MABO 这一定位指标可知,所述算法能够获得定位质量较好的候选框。

所述算法的不足之处在对浅层的神经网络检测框架 Fast RCNN(Model-S),其检测精度并不是最优。针对这种情况,接下来将继续从超像素角度研究目标的显著性对目标检测精度的影响,以提高所生成的候选框的检测精度。

参考文献(References)

[1] HOSANG J, BENENSON R, DOLLÁR P, et al. What makes for effective detection proposals[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(4): 814 – 830.

[2] 王春哲, 安军社, 姜秀杰, 等. 基于颜色距离与 Edge Boxes 候选区域算法[J]. 液晶与显示, 2019, 34(7): 698 – 707.
WANG Chunzhe, AN Junshe, JIANG Xiujie, et al. Region proposals algorithm based on color distance and Edge Boxes[J]. Chinese Journal of Liquid Crystals and Displays, 2019, 34(7): 698 – 707. (in Chinese)

[3] 王春哲, 安军社, 姜秀杰, 等. 基于卷积神经网络的候选区域优化算法[J]. 中国光学, 2019, 12(6): 1348.
WANG Chunzhe, AN Junshe, JIANG Xiujie, et al. Region proposal optimization algorithm basedon convolutional neural networks[J]. Chinese Optics, 2019, 12(6): 1348. (in

Chinese)

[4] UIJLINGS J R R, VAN DE SANDE K E A, GEVERS T, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154 – 171.

[5] 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.

[6] ALEXE B, THOMAS D, VITTORIO F. Measuring the objectness of image windows[J]. IEEE Transactions on Software Engineering, 2012, 34(11): 2189 – 2202.

[7] CHENG M M, ZHANG Z M, LIN W Y, et al. BING: Binarized normed gradients for objectness estimation at 300 fps[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014: 3286 – 3293.

[8] ZITNICK C L, DOLLAR P. Edge boxes: locating object proposals from edges [C]//Proceedings of European Conference on Computer Vision, 2014: 391 – 405.

[9] 彭宇新, 綦金玮, 黄鑫. 多媒体内容理解的研究现状与展望[J]. 计算机研究与发展, 2019, 56(1): 183 – 208.
PENG Yuxin, QI Jinwei, HUANG Xin. Current research status and prospects on multimedia content understanding[J]. Journal of Computer Research and Development, 2019, 56(1): 183 – 208. (in Chinese)

[10] PENG Y X, ZHANG J, YE Z D. Deep reinforcement learning for image hashing[J]. IEEE Transactions on Multimedia, 2020, 22(8): 2061 – 2073.

[11] HE X T, PENG Y X. Fine-grained visual-textual representation learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(2): 520 – 531.

[12] LIU Y, CHENG M M, HU X W, et al. Richer convolutional features for edge detection [C]//Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 41: 1939 – 1946.

[13] 丁鹏, 张叶, 贾平, 等. 基于视觉显著性的海面舰船检测技术[J]. 电子学报, 2018, 46(1): 127 – 134.
DING Peng, ZHANG Ye, JIA Ping, et al. Ship detection on sea surface based on visual saliency[J]. Acta Electronica Sinica, 2018, 46(1): 127 – 134. (in Chinese)

[14] 李宇, 刘雪莹, 张洪群, 等. 基于卷积神经网络的光学遥感图像检索[J]. 光学精密工程, 2018, 26(1): 200 – 207.
LI Yu, LIU Xueying, ZHANG Hongqun, et al. Optical remote sensing image retrieval based on convolutional neural networks[J]. Optics and Precision Engineering, 2018, 26(1): 200 – 207. (in Chinese)

[15] LIU T, SUN J, ZHENG N N, et al. Learning to detect a salient object [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007: 1 – 8.

[16] TONG N, LU H C, ZHANG L H, et al. Saliency detection with multi-scale superpixels[J]. IEEE Signal Processing Letters, 2014, 21(9): 1035 – 1039.

[17] ISHIKURA K, KURITA N, CHANDLER D M, et al. Saliency detection based on multiscale extrema of local perceptual color differences[J]. IEEE Transactions on Image Processing, 2018, 27(2): 703 – 717.

[18] ZHAO Y Z, PENG Y X. Saliency-guided video classification via adaptively weighted learning[C]//Proceedings of IEEE International Conference on Multimedia and Expo, 2017: 847 – 852.

[19] HE XT, PENG Y X, ZHAO J J. Fine-grained discriminative localization via saliency-guided faster R-CNN [C]// Proceedings of the 25th ACM international conference on Multimedia, 2017: 627 – 635.

[20] WANG L Z, WANG L J, LU H C, et al. Saliency detection with recurrent fully convolutional networks[C]//Proceedings of 14th European Conference on Computer Vision, 2016: 825 – 841.

[21] ACHANTA R, SHAJI A, SMITH K, et al. SLIC superpixels compared to state-of-the-art superpixel methods[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11): 2274 – 2282.

[22] CARREIRA J, SMINCHISESCU C. CPMC: automatic object segmentation using constrained parametric min-cuts [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1312 – 1328.

[23] MANEN S, GUILLAUMIN M, GOOL L V. Prime object proposals with randomized prim's algorithm[C]//Proceedings of IEEE International Conference on Computer Vision, 2013: 2536 – 2543.

[24] KRÄHENBÜHL P, KOLTUN V. Geodesic object proposals[C]// Proceedings of European Conference on Computer Vision, 2014: 725 – 739.

[25] PONT-TUSET J, ARBELÁEZ P, BARRON J T, et al. Multiscale combinatorial grouping for image segmentation and object proposal generation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39 (1): 128 – 140.

[26] RANTALANKILA P, KANNALA J, RAHTU E. Generating object segmentation proposals using global and local search[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014: 2417 – 2424.

[27] GIRSHICK R. Fast R-CNN [C]//Proceedings of IEEE International Conference on Computer Vision, 2015: 1440 – 1448.

[28] ZHANG Z M, LIU Y, CHEN X, et al. Sequential optimization for efficient high-quality object proposal generation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(5): 1209 – 1223.