计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(8):105−115 [doi: 10.15888/j.cnki.csa.009210]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于端到端的多任务商标分卡模型①

张贞奘, 苏　海, 余松森

(华南师范大学 软件学院, 佛山 528225)
通信作者: 余松森, E-mail: yss8109@163.com

摘　要: 目前商标分卡处理方法是先进行文本检测再进行区域分类, 最后对不同的区域进行拆分组合形成商标分卡. 这种分步式的处理耗时长, 并且因为误差的叠加会导致最终结果准确率下降. 针对这一问题, 本文提出了多任务的网络模型 TextCls, 通过设计多任务学习模型来提升商标分卡的检测和分类模块的推理速度和精确率. 该模型包含一个特征提取网络, 以及文本检测和区域分类两个任务分支. 其中, 文本检测分支采用分割网络学习像素分类图, 然后使用像素聚合获得文本框, 像素分类图主要是学习文本像素和背景像素的信息; 区域分类分支对区域特征细分为中文、英文和图形, 着重学习不同类型区域的特征. 两个分支通过共享特征提取网络, 像素信息和区域特征相互促进学习, 最终两个任务的精确率得以提升. 为了弥补商标图像的文本检测数据集的缺失以及验证 TextCls 的有效性, 本文还收集并标注了一个由 2 000 张商标图像构成的文本检测数据集 trademark_text (https://github.com/kongbailongtian/trademark_text), 结果表明: 与最佳的文本检测算法相比, 本文的文本检测分支将精确率由 94.44% 提升至 95.16%, 调和平均值 *F*1 score 达 92.12%; 区域分类分支的 *F*1 score 也由 97.09% 提升至 98.18%.

关键词: 商标分卡; 端到端; 文本检测; 多任务学习; 数据集

引用格式: 张贞奘,苏海,余松森.基于端到端的多任务商标分卡模型.计算机系统应用,2023,32(8):105–115. http://www.c-s-a.org.cn/1003-3254/9210.html

End-to-end Multi-task Trademark Sub-card Model

ZHANG Zhen-Yan, SU Hai, YU Song-Sen

(School of Software, South China Normal University, Foshan 528225, China)

Abstract: The current trademark sub-card processing method is to first carry out text detection, then conduct area classification, and finally split and combine different areas to form a trademark sub-card. This step-by-step processing takes a long time, and the accuracy of the final results will decrease due to the superposition of errors. Therefore, this study proposes a multi-task network model TextCls, which can improve the inference speed and accuracy of the detection and classification modules. TextCls consists of a feature extraction network and two task branches of text detection and regional classification. The text detection branch uses the segmentation network to learn the pixel classification map and then employs pixel aggregation to obtain the text boxes. The pixel classification map is mainly used to learn the information of text and background pixels. The regional classification branch subdivides regional features into Chinese, English, and graphics, focusing on learning the characteristics of different types of regions. Through the shared feature extraction network, the two branches continuously learn pixel information and regional features, and finally the precision of the two tasks is improved. To make up for the lack of text detection datasets for trademark images and verify the effectiveness of TextCls, this study collects and labels a text detection dataset trademark_text (https://github.com/kongbailongtian/trademark_text), which consists of 2 000 trademark images. The results show that compared with the optimal text detection algorithm, the text detection branch of TextCls increases the accuracy rate from 94.44% to 95.16%, with the

① 基金项目: 广东省基础与应用基础研究基金区域联合基金青年基金 (2021A1515110673)
收稿时间: 2023-02-09; 修改时间: 2023-03-14, 2023-03-20; 采用时间: 2023-03-23; csa 在线出版时间: 2023-06-09
CNKI 网络首发时间: 2023-06-14

harmonic mean $F1$ score reaching 92.12%; the $F1$ score of the regional classification branch also increases from 97.09% to 98.18%.

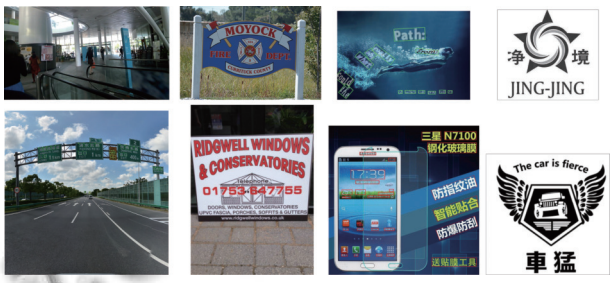
Key words: trademark sub-card; end-to-end; text detection; multi-task learning; datasets

商标图像中不仅会包含图形部分, 文本也是重要组成部分, 如果简单将整个商标图像作为商标检索系统的输入, 那么由于图形和文本两者的相互干扰将无法检索出局部相似的商标图像, 因此需要准确拆分成不同部分进行精细检索, 拆分组合形成商标图像分卡. 在实际的商标分卡场景下, 依然面临以下挑战: (1) 商标图像中文本实例往往有不同的形状表现, 且和图形部分分布相邻, 这要求检测出来的文本框能完美贴合, 才能避免切分时对图形部分造成损伤; (2) 在我国, 商标图像中的文本往往会包含中文和英文, 因此需要对文本区域进一步分类和组合.

为了完整实现商标分卡将涉及两个独立的模块: 文本检测和区域分类, 并通过分步顺序实现. 首先将商标图像输入到文本检测模块中, 在得到文本框后, 再在原图上的截取相应的区域输入到区域分类模块中, 在获得分类信息后进行拆分和组合工作. 简单的串联任务无法充分发挥深度卷积网络的潜力, 因为将两个任务完全独立, 无法进行特征共享, 而且两个模块的耗时也将使得商标分卡无法满足实时性.

端到端的文本识别[1–3] 近些年受到了越来越多的关注, 序列属性是文本的重要特征, 然而在自然场景中的建筑物、栏杆和街区由于序列外观将会呈现假阳性, 为了使得网络具有区分不同模式的能力, 通过将文本检测和文本识别两个任务进行特征共享, 并进行端到端的训练, 同时端到端的框架在推理速度上也具有一定优势[1]. 受到端到端文本识别模型的启发, 本文提出了文本检测和文本区域分类的多任务模型 TextCls, 通过共享特征, 完成端到端的模型设计, 新的商标分卡流程则是在对商标图像进行文本检测的同时, 进一步获得该区域的分类信息. 由于区域分类对中文、英文和图形特征进行加强学习, 使得文本检测分支获得的分割图更加精准, 最终得到更加拟合文本区域的文本框; 而文本检测分支所学习到的语义信息和空间信息也将反馈到区域分类分支上. TextCls 通过共享两个任务的特征, 端到端的框架拥有更快的推理速度, 同时两个任务的相互促进, 在区域分类分支获得极高的精确率的同时, 也进一步提升了文本检测分支的效果.

在自然场景下的文本检测任务中, 有大量的数据集[4–9] 被提出, 这些数据集主要采集于自然街景、合成图像或者人为拍摄等, 所采集的街景图片一般为特定任务或者单一语种的图片; 合成的图片也是对单一语种的单词进行旋转、形变和透视等操作完成组合; 而所拍摄的图片虽然包含了中文和英文文本区域, 但是存在严重的比例失调, 主要以英文为主, 然而在我国的商标图片中, 英文和中文出现的频率都是极高的. 同时, 如图 1 所展示的样本图片, 现已公开的数据集大部分都是具有复杂的背景, 而对于商标图像则是具有空白的背景; 同时大量的商标图像由图形区域和文本区域构成, 但是在商标中的图形区域通常以简易线条绘制且文本区域会以艺术字的形式出现, 导致两者具有相似的特征, 这也构成了商标图像的特殊性.



(a) 街景图片 (b) 拍摄图片 (c) 合成图片 (d) 商标图片

图 1 数据集图片展示

针对缺乏以商标图像为数据源的文本检测数据集, 本文收集和筛选了包含中英文区域的商标图像, 使用多边形标注了一个新的数据集 trademark_text. 这个数据集一共包括了 2 000 张商标图像, 以及超过 4 000 个标注实例, 并且使用了多边形进行标注, 能比较好地包裹住弯曲的文本区域. 为了保持中英文的实例数量平衡, 共有 2 260 个英文实例和 1 943 个中文实例, 同时考虑到商标图片中的图形区域也由简易线条构成, 因此也标注了图形区域用于区域分类学习.

综上所述, 本文的主要贡献可以总结如下.

(1) 本文提出了一个具有文本检测和区域分类的多任务模型 TextCls, 不仅拥有极快的推理速度, 通过共享

骨干网络, 两个分支相互促进, 在区域分类任务取得了极高的精确率的同时, 进一步提升了文本检测的效果.

(2) 为了填补商标图像的文本检测数据集的空白, 本文收集了一个由商标图像构成的文本检测数据集. 为了验证 TextCls 的有效性, 在进行区域标注的同时, 还标注了对应的类别信息.

1 国内外研究现状

针对商标分卡任务中的文本检测模块和区域分类模块, 本文通过构建多任务模型, 完成端到端训练, 不仅能加快整个任务的推理速度, 同时由于两个模块通过共享特征, 能相互促进, 在区域分类任务取得较高的精确率的同时, 也进一步提升了文本检测的效果.

1.1 常见的多任务模型

现有的多任务模型实现方法可以分为软参数共享和硬参数共享[10]. 软参数共享是指不同的任务拥有独立的模型, 模型参数彼此约束, 对模型参数的距离进行正则化来保证参数的相似, 例如在 cross-stitch networks[11] 中使用特定任务网络的每一层激活函数的线性组合作为软特征融合的方法, 而 slice networks[12] 进一步扩展该想法, 允许学习层、子空间和跳过连接的选择性共享; 而硬参数共享则是指模型的主体部分共享参数, 输出结构任务独立[13], 而为了优化不同任务可能共享不同层次的特征, 提出在不同层特征开始针对各自任务设计分支[14,15]. 软参数共享主要是针对具有较大差距的任务, 而硬参数共享则是针对目标较为一致的任务.

1.2 基于深度学习的文本检测

在文本检测模块中, 随着基于深度学习的目标检测算法的发展, 将文本区域作为检测目标也在此任务的基础上有了更进一步的发展, 目前的算法大致分为以下 3 大类: 基于锚框的方法、直接回归的方法和基于像素分割的方法.

其中基于锚框的方法, 则是预先在需要检测的特征图上设定好进行检测的 anchor, 通过 anchor 来检测文本区域. 如 Tian 等人[16] 提出的 CTPN, 不直接检测整个文本, 而是设计不同高度、等框的 anchor 来检测文本区域, 将整个文本切成一个个竖条, 然后把检测出来的区域连接作为检测结果. 之后 Shi 等人[17] 提出 segLink, 主要在竖条检测框上增加宽、高和角度的回归, 使得其能够检测多方向的文本. 针对任意形状的文本区域, 基于锚框的方法需要设计十分复杂的锚框, 导致整个程序推理速度慢, 而拟合情况也并不佳.

而基于直接回归的方法则是不预先设定 anchor, 如 Zhou 等人[18] 提出的 EAST, He 等人[19] 提出的 DDR 选择在像素级上直接回归预测该点对应的文本框的 4 个点坐标. 而 Liu 等人[1] 提出的 ABCNET 和 Zhu 等人[20] 提出的 FCENet 则是使用数学曲线对文本区域进行拟合, 对曲线表达式进行回归, 使用直接回归解决了任意形状的文本检测, 但是对于曲度过大的文本区域依然无法有效拟合.

基于像素分割的方法则是通过网络检测得到像素分割图, 然后采用后处理得到文本框. Wang 等人提出的 PSENet[21] 对文本内核区域进行预测, 并通过渐进式扩展算法, 对文本行的内核不断扩展至文本行大小, 完成目标像素聚合最终输出预测框. 因为在获得最小内核的语义分割图时能较为清楚进行区分不同的实例, 再不断加入像素来扩展不同实例的区域, 直到发现最大的内核作为预测结果. 为了轻量化场景文本检测模型, Wang 等人在 PSENet 的基础上提出的 PANNet[22] 使用级联特征金字塔增强模块在参数少的情况下获得更好的像素分割结果. 聚合时需要将文本像素聚合, 一般做法是通过固定阈值进行过滤, 而 Liao 等人[23] 提出的 DBNet 则是使用近似可微分二值化进行端到端训练学习阈值图, 可以提高后处理的速度, 并且获得更加精准的文本框. 基于像素分割的方法通过获得像素分割图, 可以表征出任意形状的文本区域, 然而相邻的文本实例的像素会出现粘连, 导致不同实例会被同一个文本框检测出.

1.3 文本检测数据集

在文本检测网络的发展迅速也得益于许多优秀的数据集不断被提出来, 数据集从自然场景图像到特定场景以及合成数据, 标注方式从矩形标注到多边形, 再到任意形状的文本区域, 这些数据集的提出极大地促进了文本检测模型的发展. ICDAR 2003 是自然场景检测的第一个基准数据集, 其使用矩形框标注了 509 张图片, 而 ICDAR 2011 和 ICDAR 2013 主要是在此基础上进行扩充和修正, 而 ICDAR 2015 是第一个提出使用四边形标记的数据集, 并且包含了一些低质量照片, 而在 ICDAR 2017 中提出的 RCTW-17[7] 包含了中文、英文文本, , 并且采用了平行四边形进行标注. 与此同时, Total-Text 则收集了包含了水平、多方向和弯曲等多种文本实例情况的图片数据, 并且采用四边形和多边形框同时进行标注. 而 SCUT-CTW1500[9] 则是专门针对弯曲文本, 采用 14 个顶点进行多边形标注.

从以上数据集的发展能看出, 数据集朝着更加精准标注和更紧凑的方向发展, 针对现实生活中文本可能出现的场景和形式做出新的调整. 本文针对商标图像背景缺乏, 图形区域和文本区域具有相似性以及中英文同时存在的特殊性, 构建了一个包含了 2 000 张商标图像的文本检测数据集.

1.4 基于卷积神经网络的图像分类

在区域分类模块中, 主要是对一个图像区域进行分类, 属于图像分类的范畴, 通过卷积神经网络 (convolutional neural networks, CNN) 可以较好地解决该类问题, 一般做法是通过堆叠卷积层和全连接层, 最后通过分类器获得最终的分类结果. AlexNet[24] 在除了使用 5 个卷积层和 3 个全连接层外, 还提出使用 ReLU 激活函数来优化梯度消失问题, 以及使用 Dropout 来防止过拟合. Simonyan 等人[25] 提出的 VGG 使用 3×3 的卷积核并保持卷积层中输出特征图尺寸不变, 可以大大减少模型训练的开销, 使其具备了加深网络的可能性, 更深的网络结构使其在图像分类上获得了更好的效果. He 等人[26] 提出来的 ResNet 则是提出残差学习, 使得网络能在大幅度增加深度的同时拥有好的效果, 大大提高了图像分类的精度, 由于其出色的特征提取能力也经常被用作其他领域的骨干网络. 而在特征提取部分保留 CNN 的卷积层构成全卷积网络, 可以满足目标检测需要输入不同尺度图像的要求, 这也是本文能进一步设计多任务模型的基础.

综上所述, 在文本检测如果采用锚框和直接回归的方式进行检测, 对于商标图像中任意形状的文本区域, 由于所预测出的文本框包含的顶点受限, 无法有效拟合, 为了对商标图像进行拆分时不同区域之间不会相互影响, 因此预测的文本框需要紧紧贴合文本区域, 而基于像素分割的方法则是具有拟合任意形状的优势, 对于相邻文本区域粘连问题则是需要对文本像素信息加强学习; 在获得文本框后, 需要对文本区域的进一步分类为中文或英文, 因此需要着重学习不同区域的特征信息. 以上两个任务本质上是对文本图像特征的学习, 在任务上具有一致性, 因此文本检测任务和区域分类任务可以在特征提取阶段进行硬参数共享, 在此基础上本文提出了 TextCls 多任务模型, 通过一个模型完成两个任务的端到端训练, 不仅仅能加快整个任务的推理速度, 同时由于两个模块通过共享特征, 能相互促进, 最终提升两个任务的效果.

2 基于端到端的多任务商标分卡模型

本文所设计的多任务模型结构如图 2 所示, 主要包含 3 个模块: 骨干网络及特征金字塔增强模块 (feature pyramid enhancement modules, FPEMs)、文本检测模块和文本区域分类模块. 骨干网络通过卷积操作对输入的商标图像进行特征提取, 再经由特征金字塔增强模块将不同尺寸的特征图进行融合作为检测和分类任务的输入; 文本检测模块对输入的特征学习得到文本区域、文本内核和实例向量 3 种带有语义的像素分割图, 经由像素聚合得到文本框; 文本区域分类模块则利用文本框在特征图上截取文本区域特征, 然后进行细化特征信息, 最终完成区域分类工作.

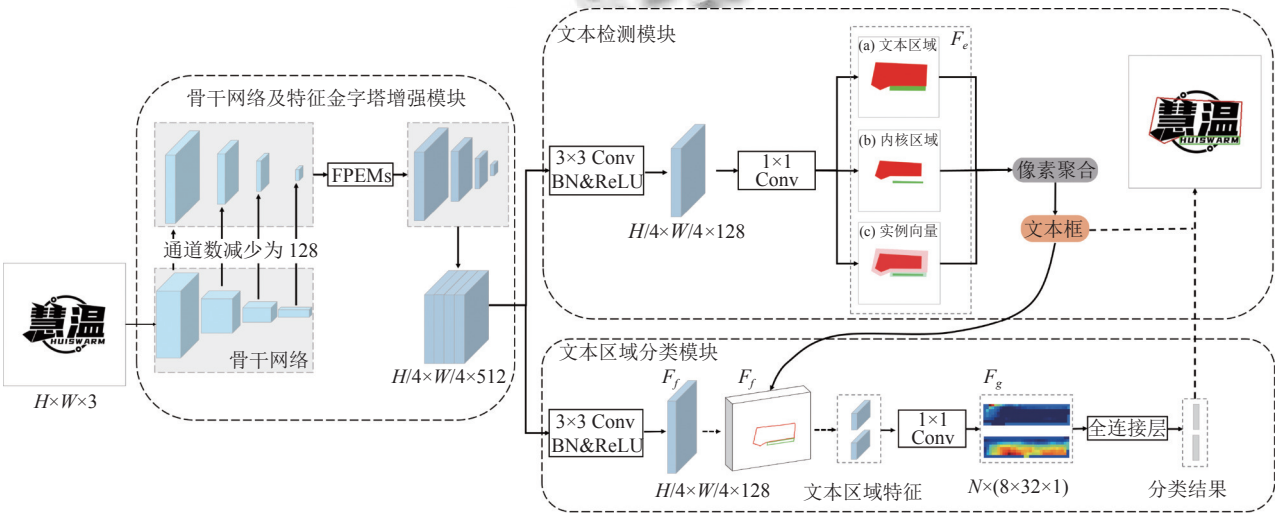


图 2 TextCls 网络结构

2.1 骨干网络及特征金字塔增强模块

首先, 将 $H\times W\times 3$ 大小的图像输入到骨干网络 ResNet18[26] 中进行特征提取, 将其最后 4 层的卷积层特征都减少为 128 通道, 然后输入到特征金字塔增强模块 FPEMs 中, 如图 3(a) 所示, 其尺寸分别为原图大小的 1/32、1/16、1/8 和 1/4.

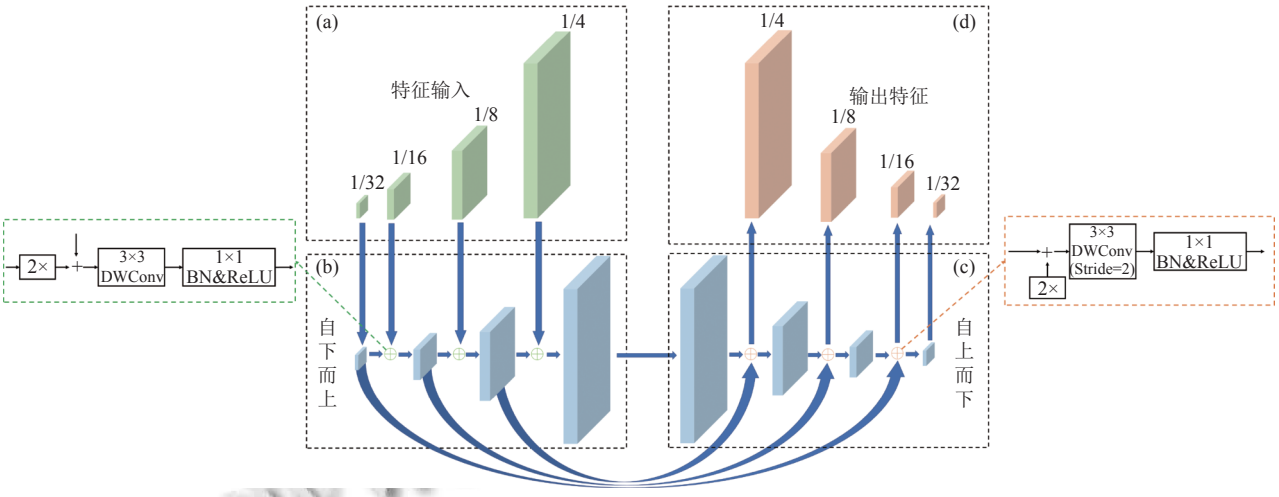


图 3 特征金字塔增强模块 FPEMs

FPEMs 可以划分成由小到大的特征增强和由大到小的特征增强两个过程.

(1) 由小到大的特征增强的过程如图 3(b) 所示, 最底层 1/32 的特征图向上 2 倍的线性插值再和 1/16 的特征图进行像素相加, 最后经过 3×3 的深度可分离卷积和 1×1 的卷积得到新的 1/16 特征图, 依次向上, 最终得到 1/4 的特征图;

(2) 而在由大到小的特征增强过程则是图 3(c) 所示, 将 1/8 的特征图使用 2 倍的线性插值再和 1/4 的特征图进行像素相加, 最后经过大小为 3×3 且步长为 2 的深度可分离卷积和 1×1 的卷积得到新的 1/8 特征图, 同理可以获得 1/16 和 1/32 的特征图, 最终获得 1/4、1/8、1/16 和 1/32 的特征图, 如图 3(d) 所示.

最后将 1/4、1/8、1/16 和 1/32 进行 concat 连接, 再和输入的特征进行相加操作, 得到了单个特征金字塔的输出结果. 从图 3 的结构可以看出, 这个结构的输入和输出特征尺寸和通道数都是一致的, 因此可进行堆叠, 将一个模块的输出特征作为下一个模块的输入, 将其感受野进一步扩大, 可以学习到更广更深的特征.

2.2 文本检测模块

本文的检测分支结构如图 2 的文本检测模块所示, 检测头主要由两个卷积层构成: 经过 FPEMs 之后得到大小为 $H/4\times W/4\times 512$ 的特征图, 首先通过一个 3×3 的卷积将其通道数降为 128, 再通过 1×1 的卷积获得特征图 F_e, 分别代表文本区域、内核区域和实例向量 3 种像素分割图, 其维度为 $2+D$, 而实例向量的维度为 D, 由于一个向量具有上下左右 4 个方向, 因此在实现过程中将 D 设置为 4.

在得到特征图 F_e 后, 使用像素聚合 (pixel aggregation, PA) 获得文本框, 基本做法为: 将不同文本区域看作为不同的聚类, 而内核区域是聚类的中心, 首先根据连通域分析法, 将不同的内核区域进行划分, 每次探索一个文本实例, 通过内核像素在文本区域内外扩展, 每次计算相邻像素的实例向量之间的欧式距离, 如果两者的欧式距离小于阈值 d 就扩展该点, 参考 PANNet[22] 将 d 固定设置为 3. 经过多次扩展, 最后输出所预测的文本框.

2.3 文本区域分类模块

本文的分类分支结构如图 2 的文本区域分类模块所示, 经过 FPEMs 之后得到大小为 $H/4\times W/4\times 512$ 的特征图, 再经过一个 3×3 的卷积将通道数降为 128, 即得到了特征图 F_f; 通过文本检测模块所获得的文本框, 在 F_f 上进行截取到文本区域特征, 并将尺寸缩放至 8×32, 输入到 1×1 的卷积中, 学习当前区域的特征信息, 并将通道降为 1, 得到大小为 $N\times(8\times32\times1)$ 的特征图 F_g, 其中 N 代表文本实例个数; 再将特征图 F_g 输入

到全连接层获取分类结果.

与一般的图像分类不同, TextCls 的文本区域分类模块并不是直接输入原始图像, 而是借助文本检测模块获得的文本框进行截取特征区域. 由于两个模块共享特征提取模块, 所截取的特征图 F_g 中的像素是带有语义信息的, 能协助提升区域分类的效果.

2.4 损失函数

本文的损失函数可以由式 (1) 表示:

$$L=\alpha Loss_{\text{det}}+\beta Loss_{\text{cls}} \tag{1}$$

其中, $Loss_{\text{det}}$ 代表文本检测的损失函数, $Loss_{\text{cls}}$ 代表文本区域分类的损失函数. 由于文本检测模块结果输出为 6 张像素分割图, 而文本区域分类模块仅学习 1 张特征图, 为了平衡 $Loss_{\text{det}}$ 和 $Loss_{\text{cls}}$ 的重要性, 将 α:β 设置为 6:1, 当固定 α 为 1 时, 则将 β 设置为 0.16.

而对于 $Loss_{\text{det}}$ 使用式 (2) 具体的展示:

$$Loss_{\text{det}}=Loss_{\text{text}}+\mu Loss_{\text{ker}}+\omega\left(Loss_{\text{agg}}+Loss_{\text{dis}}\right) \tag{2}$$

$Loss_{\text{text}}$ 代表文本区域分割的损失, $Loss_{\text{ker}}$ 代表内核区域分割的损失, 而 $Loss_{\text{agg}}$ 和 $Loss_{\text{dis}}$ 主要计算实例向量的损失, 其中 $Loss_{\text{agg}}$ 代表聚类损失, $Loss_{\text{dis}}$ 代表差异损失. μ 和 ω 用于平衡 $Loss_{\text{text}}$、$Loss_{\text{ker}}$、$Loss_{\text{agg}}$ 和 $Loss_{\text{dis}}$ 的重要性, 参考 Pan++[2] 将 μ 设置为 0.5 及 ω 设置为 0.25.

$Loss_{\text{text}}$ 和 $Loss_{\text{ker}}$ 如式 (3) 和式 (4) 所示:

$$Loss_{\text{text}}=1-\frac{2\sum_i P_{\text{text}}(i)G_{\text{text}}(i)}{\sum_i P_{\text{text}}(i)^2+\sum_i G_{\text{text}}(i)^2} \tag{3}$$

$$Loss_{\text{ker}}=1-\frac{2\sum_i P_{\text{ker}}(i)G_{\text{ker}}(i)}{\sum_i P_{\text{ker}}(i)^2+\sum_i G_{\text{ker}}(i)^2} \tag{4}$$

式 (3) 中的 $P_{\text{text}}(i)$ 和 $G_{\text{text}}(i)$ 分别代表文本区域像素分割图的第 i 个像素点的值和文本区域真实标签对应的像素值了; 而式 (4) 中的 $P_{\text{ker}}(i)$ 和 $G_{\text{ker}}(i)$ 分别代表文本内核像素分割图的第 i 个像素点的值和文本内核真实标签对应的像素值.

$Loss_{\text{agg}}$ 是为了将同一实例的像素聚合相应的内核区域, 将像素和相应的内核区域的距离最小化, 具体使用式 (5) 来表示:

$$Loss_{\text{agg}}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|T_i|}\sum_{p\notin T_i}D_1(p,K_i) \tag{5}$$

其中, N 代表文本实例数量; T_i 表示第 i 个文本实例; K_i 表示属于第 i 个文本实例的内核区域; $D_1(p, K_i)$: 文本像素 p 和内核区域之间的距离, 具体见式 (6):

$$D_1(p,K_i)=\ln\left(R\left(\|F(p)-g(K_i)\|-\delta_{\text{agg}}\right)^2+1\right) \tag{6}$$

其中, $R(\cdot)$ 代表 ReLU 函数, 用来保证输出非负; $F(p)$ 表示像素 p 的实例向量; $g(K_i)$ 表示内核区域的实例向量; δ_{agg} 表示文本像素和内核之间的距离阈值.

除了要保证文本区域能进行内聚, 还应保证内核区域之间以及和背景区域能保持一定的距离, 正是 $Loss_{\text{dis}}$ 所代表的差异损失, 具体如式 (7) 所示:

$$Loss_{\text{dis}}=\frac{1}{N^2}\sum_{i=1}^{N}\left(D_b(K_i)+\sum_{\substack{j=1\\ j\neq i}}^{N}D_2\left(K_i,K_j\right)\right) \tag{7}$$

$$D_b(K_i)=\frac{1}{|B|}\sum_{p\in B}\ln\left(R(\delta_{dis}-\|F(p)-g(K_i)\|)^2+1\right) \tag{8}$$

$$D_2\left(K_i,K_j\right)=\ln\left(R\left(\delta_{dis}-\left\|g(K_i)-g\left(K_j\right)\right\|\right)^2+1\right) \tag{9}$$

其中, B 表示背景区域; D_b 表示背景像素和内核区域之间的距离, 具体见式 (8); $D_2(K_i, K_j)$ 表示内核区域 K_i 和内核区域 K_j 之间的距离, 具体见式 (9); δ_{dis} 表示内核区域与其他内核区域或者背景之间的距离阈值.

在文本区域分类分支中, 主要是对每个不同的区域进行真实分布和概率分布之间的差异计算, 使用的交叉熵函数如式 (10):

$$Loss_{\text{cls}}=\frac{1}{M}\sum_{i=1}^{M}H(P_i,Q_i) \tag{10}$$

$$H(P_x,Q_x)=-\sum_{j=1}^{c}P\left(x_j\right)\log Q\left(x_j\right) \tag{11}$$

其中, M 表示待分类的区域数量; P_i 表示第 i 个区域特征的真实值; Q_i 表示第 i 个区域特征的预测值; $H(P_i, Q_i)$ 表示预测值和真实值之间的交叉熵大小, 具体如式 (11); c 表示区域分类的种类数, 数据集中提供了中文、英文和图形分类, 因此 c 值为 3; x_j: 对样本 x 在类别 j 上进行讨论; $P(x_j)$ 表示符号函数 (0 或 1), 如果样本 x 的真实标签等于类别 j 取 1, 否则取 0; $Q(x_j)$ 表示样本 x 属于类别 j 的预测概率.

2.5 数据集

trademark_text 共包含 2 000 张商标图像, 共计超过

4 000 个标注实例, 为了平衡中英文实例的数量, 包含 2 260 个英文实例和 1 943 个中文实例. trademark_text 采集于已公布的真实商标图样, 同时商标图像中的文本为含有中文、英文、单字分布、水平文本、竖直文本、弯曲文本及艺术字等情形, 同时文本和图形之间的空间分布可分为上下、上中下、左右以及镶嵌等情况, 具体的数据集划分 1 750 张作为训练集, 250 张作为测试集.

针对商标图像中文本区域并不是以单一矩形出现, 并且为配合进一步任意形状的文本检测算法研究, 本课题将使用曲线标注法, 使用 SCUT-CTW1500[9] 中提出的 14 点标注方式, 具体标注方式如图 4 所示. 首先如图中紫色点所示, 先确定文本区域边界的 4 个顶点. 然后用 5 个点确定文本区域的上边界, 即 5、6、7、8 和 9 所示的绿色点. 再使用 5 个点确定文本区域的下边界, 即 10、11、12、13 和 14 所示的绿色点, 最终构成了一个紧密贴合文本区域的曲线文本框. 由于商标图像中包含中文和英文区域, 因此在数据集标注时也进行了标记, 而在商标图像中, 图形区域也是由一些线条构成, 为了加强对不同区域的特征进行加强学习, 所以对图形区域也进一步标注, 数据集的标注效果如图 5 所示. trademark_text 数据集将公布在 https://github.com/kongbailongtian/trademark_text.

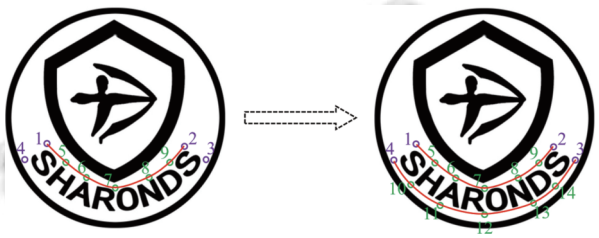


图 4 曲线文本框标注说明

图 5 trademark_text 标注结果可视化

3 实验分析

为了验证所提出来的 TextCls 的有效性, 本文将对文本检测任务和文本区域分类任务进行评估.

3.1 实验设置

本文使用在 ImageNet 上预训练得到的 ResNet18[26] 作为骨干网络; 训练和测试采用本文提出的 trademark_text 数据集; 实例向量的维度 D 设置为 4, 对应上下左右 4 个方向; 文本内核的收缩率 r 设为 0.7, PA 的距离阈值 d 设为 3, 初始学习率为 1×10^{-3}, 使用“poly”学习率策略, 其中幂设置为 0.9. 本论文的结果都是使用 PyTorch 在一个 3090Ti 上进行训练测试.

3.2 评价指标

(1) 文本检测评价指标

如何进行评估当前预测框是否正确, 需要通过预测框和真实框的重叠度 (intersection over union, *IoU*) 来表示. 为了评估文本检测网络的有效性, 通常会采用

精准率 (precision, P)、召回率 (recall, R) 和 $F1$ 分数 ($F1$) 以便进行定量比较.

1) 重叠度 IoU

IoU 主要是用于判断两个不同区域的重叠度, 其数值越高说明两个区域相交比例越高, 在文本检测领域则是通过 IoU 来判断预测框 P 是否接近标注的真实框 GT, 具体的公式如式 (12) 所示:

$$IoU = \frac{Area(P \cap GT)}{Area(P \cup GT)} \tag{12}$$

其中, $Area(P \cap GT)$代表预测框和真实框相交区域的面积, $Area(P \cup GT)$代表预测框和真实相并区域的面积, 两者的比值就是 IoU 数值. 设置一个阈值 threshold, 当 IoU 数值大于 threshold, 认定输出的预测框为正样本 TP; 当 IoU 数字小于 threshold, 则认定输出的预测框为负样本 FP.

2) 精确率 P

精准率又被称为查准率, 主要是计算所有输出为正例的预测框中有多少为真实的正例, 具体的公式如式 (13) 所示:

$$P = \frac{TP}{TP + FP} \tag{13}$$

3) 召回率 R

召回率又被称为查全率, 主要是计算输出为正例的预测框占据所有标准的真实框的比例, 具体的公式如式 (14) 所示:

$$P = \frac{TP}{TP + FN} \tag{14}$$

其中, FN 表示标注的真实框但是未被检测的样本, TP+FN 实际就是所标注的真实框的数量.

4) $F1$

$F1$ 可以看作是精确率 P 和召回率 R 的一种加权调和平均, 具体的公式如式 (15) 所示:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{15}$$

由于精确率和召回率是相互矛盾, 一般精确率高时, 召回率却比较低; 而召回率高时, 精确率却很低. 因此, 为了综合考虑两个指标, 通过计算两者的加权调和平均进行评估, 也就是 $F1$.

(2) 文本区域分类评价指标

为了验证区域分类模型的性能, 通常使用准确率 (accuracy, Acc) 来进行评估, Acc 通过计算分类准确的样本数 right 和数据集中含有的总样本数 All 的比值获得, 具体公式如式 (16) 所示:

$$Acc = \frac{right}{All} \tag{16}$$

(3) 模型基本性能评价指标

为了证明本文提出的多任务模型的先进性, 还对模型的训练时间 (time)、参数量 (parameter, Para) 和推理帧率 (frames per second, FPS) 进行了统计.

3.3 实验结果及分析

(1) 文本检测分支性能比较

为了证明本文的模型中文本检测分支的有效性, 通过与几个流行的场景文本检测模型进行对比, 具体的实验结果如表 1 所示.

表 1 TextCls 和其他模型进行比较 (%)

模型	P	R	$F1$
DBNet[23]	78.20	80.20	79.20
PSENet[21]	91.00	82.73	86.67
PANNet[22]	91.74	88.91	90.30
FCENet[20]	94.44	**89.64**	91.98
DBNet++[27]	82.61	82.91	82.76
Ours	**95.16**	89.27	**92.12**

在表 1 中, 本文所提出的模型在 trademark_text 数据集上取得了明显的优势, 精确率高达 95.16%, 相较于 DBNet[23] 和 DBNet++[27] 有着巨大提升; 而相较于 FCENet[20] 在召回率上低了 0.37%, 但 TextCls 在精确率上高于 FCENet[20] 约 0.72%, 最终 TextCls 在 $F1$ 上达到了 92.12% 取得了更好的结果.

TextCls 正是在 PANNet[22] 的基础上进行改进的, 而 TextCls 在精确率上达到了 95.16%, 相较于 PANNet[22] 提升了 3.42%, 以及在召回率上也有 0.36%的提升, 说明本文提出的模型能进一步提升对文本的检测能力, 绘制更加精准的文本框.

(2) 文本区域分类分支性能比较

此外, 针对商标拆分时需要获得文本区域的中英文分类信息, 本文提出的 TextCls 还包含一个文本区域分类分支. ResNet18[26] 作为一个轻量级卷积神经网络在图像分类问题上具有优异的表现, 因此本文所提出的 TextCls 也是将 ResNet18[26] 作为骨干网络. 公平起见, 本文将 ResNet[26] 系列和 VGG 系列[25] 的卷积神经网络和本文的文本区域分类分支进行性能比较. 测试结果如表 2 所示.

表 2 文本区域分类结果展示 (%)

模型	*Acc*
VGG11[25]	97.63
VGG16[25]	96.18
VGG19[25]	96.72
ResNet18[26]	97.09
ResNet50[26]	97.81
ResNet101[26]	94.00
Ours	**98.18**

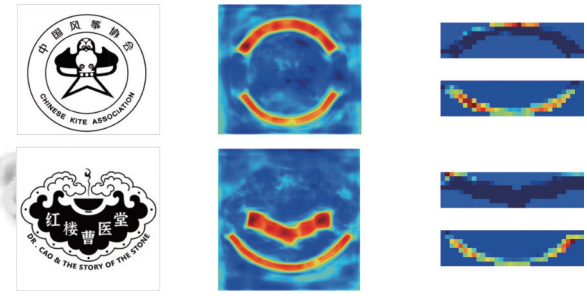
在表 2 中, TextCls 的准确率达到了 98.18%, 与其他模型相比达到了最优, 说明本文提出的模型能进一步提升文本区域分类的效果. 在 VGG 系列中, VGG16 和 VGG19 的准确率分别为 96.12% 和 96.72%, VGG11 具有最好的性能, 达到了 97.63% 的准确率, 而与 TextCls 的文本区域分类分支相比, 准确率依然有 0.55% 的差距.

在 ResNet [26] 系列中, ResNet18[26] 的准确率为 97.09%, 而 TextCls 的骨干网络采用了 ResNet18[26], 在文本区域分类分支的准确率确达到了 98.18%, 比单独使用 ResNet18[26] 相比提升了 1.09%, 随着 ResNet 层数增加, 在采用 ResNet50[26] 网络结构下, 准确率达到了 97.81%, 但是依旧有着 0.37% 的差距, 如果继续增加网络结构, 却出现了明显的过拟合现象, 如表中 ResNet101[26] 的准确率只有 94.00%, 是所有模型中最低的准确率. 通过上述分析, 即使 TextCls 为了使得模型保持轻量使用了 ResNet18[26] 作为骨干网络, 而且在训练 600 epoch 后并没有出现过拟合现象, 这表明 TextCls 所设计的多任务模型能较好平衡文本检测分支和文本区域分类分支, 但和其他模型比较也取得了最为优异的表现, 进一步证明两个分支通过共享网络进行相互学习, 进一步提升性能.

(3) TextCls 优异性能的原因分析

从上述结果展示可知, 本文提出的 TextCls 在两个任务分支都提升了性能, 正是得益于 TextCls 的多任务结构, 两个分支通过在特征提取阶段进行网络共享, 对学习到的信息进行交换, 使得文本检测分支获得的语义分割图更加精准, 并指导文本区域分类分支对区域特征的学习. 为了更加直观深入分析 TextCls 表现优异的原因, 本文将可视化 TextCls 模型学习到的特征图, 如图 6 所示.

在图 6(b) 中, 由文本检测分支获得的文本区域像素分割图能定位出文本的位置, 指导文本区域分类分支在进行特征学习时, 避免图形区域特征的干扰; 由图 6(c) 能看到文本区域分类分支获得的区域特征图分为中文 (深蓝色)、英文 (红橙色) 和背景 (淡蓝色) 这 3 种特征, 通过对区域特征的加强学习, 能协助文本检测分支获得更加精准的像素信息. 因此两个分支的信息通过共享特征提取阶段来相互交流, 最终对两个任务分支都有促进效果.



(a) 输入图像 (b) 像素分割图 (c) 区域特征图

图 6 TextCls 中间特征图展示

(4) 模型轻量级和推理速度性能比较

为了满足工业中对于实时性的要求, 本文通过构建多任务模型, 使得网络模型十分轻量, 并且具有极快的推理速度, 对比结果如表 3 所示.

表 3 模型参数量和推理速度对比

方案	参数量 (M)	速率 (FPS)
DBNet[23]+ResNet18[26]	37.45	13.48
PSENet[21] +ResNet18[26]	39.89	13.75
PANNet[22] +ResNet18[26]	23.42	25.61
FCENet[20] +ResNet18[26]	39.17	14.01
DBNet++[27] +ResNet18[26]	38.07	12.72
Ours	**12.84**	**28.39**

在表 3 中, TextCls 仅有 12.84M 的参数量, 而对于最为轻量的 PANNet[21]+ResNet[26] 包含的参数量也是 TextCls 的 1.82 倍; 而在推理速度上, TextCls 也具有明显的优势, 从 25.61 FPS 提升至 28.39 FPS, 在实际工业生产中, 文本检测模块和文本区域分类模块中间需要过渡处理, 而 TextCls 并无中间环节将具有更加明显的优势.

3.4 消融实验

为了分析确认模型中各个分支的有效性和相关性, 本节将在 trademark_text 数据集上开展消融实验. 实验的基本参数与第 3 节保持一致. 分别验证仅含有文本检测分支 det 下的效果; 再增加文本区域分类分支, 其

中区域分类分支所关注的分类区域为中文和英文区域; 最后在区域分类分支增加对背景 background 进一步训练. 结果如表 4 所示.

表 4 消融实验

模型	参数量 (M)	Time (h)	*P* (%)	*R* (%)	*F*1 (%)
det	12.25	12	91.74	88.91	90.30
det+cls	12.84	16	93.54	**89.54**	91.54
det+cls+bg	12.84	16	**95.16**	89.27	**92.12**

从表 4 中能看出, 在增加文本区域分类分支后, 其参数量增加了 0.59M 大小, 增加了 4.82% 的参数量, 而其训练只增加 4 h, 但精确率可提升 1.8%, 相应召回率增加 0.63%, 并且 *F*1 值增加了 1.24%, 正是将区域分类分支和文本检测分支构成多任务模型, 通过共享特征提取网络, 两者信息相互学习得以进一步提升性能. 而在本文工作中, 为了能加强对各个区域特征学习, 对易于文本区域混淆的图形区域也进行了分类学习, 参数量不变, 且训练时间接近, 精确率却增加了 1.62%, 而召回率仅下降了 0.27%, 最终 *F*1 增加了 0.58%, 进一步证明增加区域分类分支来加强区域特征学习的有效性.

4 结论与展望

为解决分段式商标分卡处理流程精确率较低、耗时较长等问题, 本文提出了一个包含文本检测和文本区域分类的多任务模型 TextCls, 实验证明在多任务的相互促进下, 不仅仅可以获得较好的文本区域分类结果, 精确率高达 98.18%, 并且通过对不同区域的特征进行加强学习, 能进一步促进文本检测分支的效果, 最终在 trademark_text 数据集上获得了 92.12% 的 *F*1. 接下来, 我们将针对具有更为复杂分布的商标图像进行模型设计, 进一步提升商标分卡的效果和性能.

参考文献

1 Liu YL, Chen H, Shen CH, *et al*. ABCNet: Real-time scene text spotting with adaptive bezier-curve network. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 9806–9815.

2 Wang WH, Xie EZ, Li X, *et al*. PAN++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(9): 5349–5367.

3 Huang MX, Liu YL, Peng ZH, *et al*. SwinTextSpotter: Scene text spotting via better synergy between text detection and text recognition. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 4583–4593.

4 Yuan TL, Zhu Z, Xu K, *et al*. Chinese text in the wild. arXiv: 1803.00085, 2018.

5 Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2315–2324.

6 Yao C, Bai X, Liu WY, *et al*. Detecting texts of arbitrary orientations in natural images. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence: IEEE, 2012. 1083–1090.

7 Shi BG, Yao C, Liao MH, *et al*. ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition. Kyoto: IEEE, 2017. 1429–1434.

8 Ch'ng CK, Chan CS. Total-text: A comprehensive dataset for scene text detection and recognition. Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition. Kyoto: IEEE, 2017. 935–942.

9 Liu YL, Jin LW, Zhang ST, *et al*. Detecting curve text in the wild: New dataset and new solution. arXiv:1712.02170, 2017.

10 Standley T, Zamir A, Chen D, *et al*. Which tasks should be learned together in multi-task learning? Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020. 9120–9132.

11 Misra I, Shrivastava A, Gupta A, *et al*. Cross-stitch networks for multi-task learning. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 3994–4003.

12 Ruder S, Bingel J, Augenstein I, *et al*. Latent multi-task architecture learning. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 4822–4829.

13 Kokkinos I. UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 5454–5463.

14 Lu YX, Kumar A, Zhai SF, *et al*. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. Proceedings of the 2017 IEEE

Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 1131–1140.

15 Vandenhende S, Georgoulis S, De Brabandere B, *et al*. Branched multi-task networks: Deciding what layers to share. arXiv:1904.02920, 2019.

16 Tian Z, Huang WL, He T, *et al*. Detecting text in natural image with connectionist text proposal network. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 56–72.

17 Shi BG, Bai X, Belongie S. Detecting oriented text in natural images by linking segments. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 3482–3490.

18 Zhou XY, Yao C, Wen H, *et al*. EAST: An efficient and accurate scene text detector. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 2642–2651.

19 He WH, Zhang XY, Yin F, *et al*. Deep direct regression for multi-oriented scene text detection. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 745–753.

20 Zhu YQ, Chen JY, Liang LY, *et al*. Fourier contour embedding for arbitrary-shaped text detection. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 3122–3130.

21 Wang WH, Xie EZ, Li X, *et al*. Shape robust text detection with progressive scale expansion network. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 9328–9337.

22 Wang WH, Xie EZ, Song XG, *et al*. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 8439–8448.

23 Liao MH, Wan ZY, Yao C, *et al*. Real-time scene text detection with differentiable binarization. Proceedings of the 34th AAAI Conference on Artificial Intelligence. New York: AAAI, 2020. 11474–11481.

24 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017, 60(6): 84–90. [doi: 10.1145/3065386]

25 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

26 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

27 Liao MH, Zou ZS, Wan ZY, *et al*. Real-time scene text detection with differentiable binarization and adaptive scale fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(1): 919–931. [doi: 10.1109/TPAMI.2022.3155612]

(校对责编: 牛欣悦)