YOLOv3 与顶点偏移估计相结合的车牌定位

徐光柱^{1,2)}, 匡婉¹⁾, 李兴维¹⁾, 万秋波¹⁾, 石勇涛^{1,2)}, 雷帮军^{1,2)*}

¹⁾ (三峡大学计算机与信息学院 宜昌 443002)

²⁾ (湖北省水电工程智能视觉监测重点实验室(三峡大学) 宜昌 443002)
(bangjun.lei@ieee.org.com)

摘要: 深层卷积神经网络(deep convolutional neural networks, DCNN)因其能够自动学习图像有效特征, 被广泛应用于视觉目标检测. 为克服 DCNN 目标检测算法大多因采用矩形检测框, 而无法有效地应对非约束环境下倾斜性车牌的准确定位问题. 提出一种可同时输出矩形目标检测框与关键点车牌定位解决方案, 并具体以 YOLOv3 所用网络为对象, 通过扩展其输出维度, 增设车牌顶点相对于矩形检测输出框角点的偏移量损失, 在保留其高效计算性能的前提下, 训练使其可同时输出矩形检测框及车牌顶点, 实现精准定位. 在广泛使用的大型非约束性车牌数据集 CCPD 上的实验结果显示, 所提算法不仅可以准确检测车牌顶点, 而且能够在 Base, Tilt 和 Weather 子集上取得 99% 以上的定位精度. 该方法还可扩展至其他需同时输出目标检测框及关键点的应用领域, 具有较好的应用价值.

关键词: 深度学习; 非约束车牌定位; 视觉目标检测; 点偏移估计; YOLOv3

中图法分类号: TP391.4 **DOI:** 10.3724/SP.J.1089.2021.18534

License Plate Location Based on YOLOv3 and Vertex Offset Estimation

Xu Guangzhu^{1,2)}, Kuang Wan¹⁾, Li Xingwei¹⁾, Wan Qiubo¹⁾, Shi Yongtao^{1,2)}, and Lei Bangjun^{1,2)*}

¹⁾ (College of Computer and Information Technology, China Three Gorges University, Yichang 443002)

²⁾ (Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002)

Abstract: Deep convolutional neural networks (DCNN) have been widely adopted in visual object detection as they can automatically learn effective image features and have strong generalization capabilities. Since the current mainstream visual target detection DCNN commonly output rectangular bounding boxes, they cannot effectively deal with the accurate location for tilted unconstrained license plates. To solve the problem, a solution was proposed and YOLOv3, as a specific DCNN, was trained by extending its output dimensions and establishing additional loss items about vertex offsets relative to the four bounding box corners, which empowers the trained model to output the rectangular bounding boxes and the license plate vertices meanwhile without sacrificing its high-efficiency performance. Experiments on CCPD, a widely used large scale unconstrained license plate dataset, illustrate that the proposed method can not only accurately predict the four vertices of a license plate, but also reach the precision above 99% in Base, Tilt and Weather subset. Moreover, this method can also be generalized to other application fields which require the target bounding box and key points output simultaneously and has promising application potential.

收稿日期: 2020-08-02; 修回日期: 2021-01-05. 基金项目: 国家自然科学基金(61402259, U1401252); 湖北省中央引导地方科技发展专项(2019ZYD007). 徐光柱(1979—), 男, 博士, 教授, 硕士生导师, CCF 会员, 主要研究方向为数字图像处理; 匡婉(1996—), 女, 硕士研究生, 主要研究方向为计算机视觉; 李兴维(1993—), 男, 硕士, 主要研究方向为计算机视觉; 万秋波(1996—), 男, 硕士研究生, 主要研究方向为计算机视觉; 石勇涛(1978—), 男, 博士, 副教授, 硕士生导师, 主要研究方向为数字图像处理; 雷帮军(1973—), 男, 博士, 教授, 博士生导师, 论文通讯作者, 主要研究方向为计算机视觉、人工智能.

Key words: deep learning; unconstrained license plate location; visual target detection; point offset estimation; YOLOv3

精准可靠的车牌定位算法能够在确定车牌位置的同时,尽可能地定位出与实际车牌区域吻合的候选区域,以便为后续车牌识别算法提供可靠数据.多年来,车牌定位一直是智能交通领域中的研究热点问题,特别是在非约束环境下,由于存在拍摄角度变化、拍摄相机抖动、光照条件不佳及恶劣天气等因素,使其成为一个颇具挑战性的课题.

目前,从是否使用深度学习算法的角度,车牌定位技术可分为基于传统计算机视觉和基于深层卷积神经网络(deep convolutional neural networks, DCNN)的方法.

传统车牌定位方法通常分为基于颜色特征^[1-2]、纹理特征^[3-4]以及部件特征^[5-6]的方法.基于颜色特征的方法依据车牌区域和字符颜色搭配的固定性来定位车牌.杨鼎鼎等^[1]利用颜色特征提取大致车牌区域,然后结合其他方式获取准确车牌区域.该类方法处理速度较快,但其无法处理灰度图像.由于车牌区域和图像背景的纹理信息存在明显不同,可以利用这一特点实现车牌定位,潘勇等^[3]利用车牌水平灰度投影的纹理特征获取车牌候选区域,然后根据候选区域的灰度分布特征定位车牌,其定位准确率在自收集的 157 张低光照图片上达到 97.46%,但它在其他数据集上的效果有待评估.当图像中存在交通指示牌等与车牌纹理特征类似的目标时,该类方法的定位准确率会降低.车牌是一类具有明显边框并由特定字符组成的目标,因此基于边框或字符等车牌部件的定位方法同样应用广泛.费继友等^[5]利用伽玛校正和 Canny 算子检测图像边缘,并结合车牌字符特征约束和轮廓的分布规律提取车牌候选区域,该方法定位准确率能达到 96%.对于模糊的和被遮挡的图像,车牌边缘特征及字符特征会存在一定程度丢失,从而影响该类方法的车牌定位效果.上述基于传统计算机视觉技术车牌定位方法大都选用手工设计的特征,针对特定的或相对简单的场景可取得较好的定位效果,但是在非约束场景下的定位性能表现一般.这主要是因为所采用的特征大多属于低层或中层特征,而非约束环境下的车牌图像受各种综合性因素影响严重,对其准确定位需要综合高层语义特征与中低层目标特征信息.

随着视觉大数据集与高性能图形处理器,以及大规模深层人工神经网络可训练技术的出现,

能够有效地提取层次化语义特征的 DCNN 在众多计算机视觉问题取得突破性表现,快速推动了深度学习在图像分类^[7]、视觉目标检测^[8]和语义图像分割^[9]的广泛使用.由于 DCNN 能够以端到端的方式学习并利用多层次视觉特征,从而提供较好的泛化能力,因此在车牌定位中^[10-19]也受到了广泛关注.Yonetsu 等^[11]提出一种基于 YOLOv2 的两阶段车牌检测方法,首先对图像中的车辆进行定位,然后在车辆区域中检测车牌,其在多种现实场景中表现良好.Kim 等^[12]也使用先检测车辆后检测车牌的两阶段方法实现车牌定位,它在 Caltech 数据集上的精度和召回率分别达到了 98.39%和 96.83%.Zhang 等^[13]结合卷积神经网络(convolutional neural networks, CNN)和递归神经网络(recurrent neural networks, RNN)实现车牌定位,其在非约束场景中的定位效果良好,精度与召回率在中国车牌图像数据集上分别达到 99.10%和 98.68%,但对于倾斜角度大的车牌,检测效果有待提升.上述两阶段车牌定位的方法虽然能够在一定程度降低车牌漏检的情况,但系统的实时处理能力有待提升.李祥鹏等^[14]提出一种基于 k -means++ 和 Faster R-CNN 算法^[15]的车牌定位算法,在自收集数据集和公共公路巡逻(road patrol, RP)^[16]数据集进行了测试,其准确率分别达到 98.61%和 98.99%.但其所选取的车牌测试图片均为挑战难度低的正面视图或者小角度倾斜视图,因此该算法在非约束环境中的车牌定位效果有待进一步检验.Masood 等^[17]提出一种 CNN 结构,并训练了 3 个单独的网络分别实现车牌检测、字符检测和字符识别,在包含 348 张美国车牌的数据集和包含 608 张欧洲车牌的数据集上进行了测试,车牌检测召回率均在 99%以上,但其在测试时删除了部分挑战难度大的图片.Li 等^[18]搭建了一种 CNN,通过回归网络对车牌进行定位,并对固定摄像头收集的车牌数据集进行了测试,其精度和召回率分别达到 99.2%和 98.4%;但其在数据集的选择上具有一定局限性,网络在各种非约束场景下的泛化性能有待评估.端到端的目标检测算法速度上具有一定优势,但由于使用矩形框定位车牌区域,对于倾斜的车牌无法精准定位.Xie 等^[19]提出了一种基于 CNN 的多方向车牌检测框架,通过预测带有旋转角度的矩形框实现对车牌区域的定位,其在一定程度上提

升了车牌定位的精准程度, 但只适合处理平面内的旋转问题, 对于三维空间内的倾斜车牌仍然无法精确定位. Silva 等^[20]提出一种基于 YOLO 的车牌检测与识别算法, 由车牌检测、字符分割和字符识别 3 个子任务组成, 它不仅可以对倾斜车牌进行精确定位, 同时其车牌识别也达到了一个较高的精度, 在 SSIG 数据集上的识别率为 88.56%. 该算法首先采用矩形框定位车牌大致区域, 然后通过校正算法来实现对倾斜车牌的精确定位. 虽然该算法可以解决车牌精确定位问题, 但是需要专门的校正算法, 且每次只能处理单个车牌, 会导致车牌定位速度变慢, 进而影响整个识别系统的运行效率.

由以上分析可知, 非约束环境下由于拍摄角度引起的各种车牌倾斜, 使得基于深度学习技术的目标检测算法所输出的矩形检测框不能很好地与实际车牌区域吻合, 是影响其定位精度及后续

识别率的关键原因. 而若要获取精准的车牌区域, 通常需要多个处理步骤, 会影响算法运行效率.

针对这一问题, 本文提出了一种 YOLOv3^[21]与顶点偏移估计相结合的精准车牌定位算法, 在不影响 YOLO 计算速度的基础上, 有效地扩展 YOLO 的输出功能, 使其在输出矩形目标检测框的同时, 还能够给出倾斜车牌 4 个顶点相对于输出矩形框的相对偏移量, 实现车牌精确定位. 而且, 这种改进方案也能够拓展到其他非规则视觉目标的精准定位.

1 相关算法

YOLOv3 是一种兼具精度与速度的目标检测算法, 它通过继承和融合其他目标检测算法中的优秀策略, 使其整体性能达到了一个较高的水准, 其所用网络结构如图 1 所示.

图1 YOLOv3 所用网络结构

为提升目标检测精度, 加强所用网络对小目标的检测效果, YOLOv3 不仅通过大量的残差结构加深网络深度, 还使用特征融合、多尺度预测和边界框回归等多种策略, 在检测速度与精度上实现了良好折中.

(1) 特征融合. YOLOv3 所用网络采用了类特征金字塔结构实现特征融合, 将深层网络部分所提取的低分辨率高层语义特征经过上采样后与浅层网络部分所提取的高分辨率细节特征进行融合, 在保留了位置信息的同时提升细节感知能力. 融合浅层特征与深层特征能进一步提升对纹理、颜色和边缘信息相对较少的目标的检测精度, 如图 1 中紫色虚线框部分.

(2) 多尺度预测. YOLOv3 最终输出 3 个不同尺度的特征图, 如图 1 中红色虚线框部分, 网络为

每个特征图分配 3 组大小不同的锚框, 以适应不同尺寸目标的检测. 由于 3 层输出网络的物理感受野不同, 锚框尺寸也应根据需要做调整, 实现大尺度跨越的目标检测. 针对不同尺寸目标, 锚框的大小可以通过 k -means 算法聚类得到, 以增强算法的泛化性能.

(3) 边界框回归. YOLOv3 目标框与锚框的关系如图 2 所示. YOLOv3 不直接对目标的位置及大小进行预测, 而是通过回归方式预测相对于特征图网格单元的偏移, 确定目标中心, 同时预测相对于锚框宽高的比例系数, 确定目标大小.

回归公式为

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w \times e^{t_w} \quad (3)$$

$$b_h = p_h \times e^{t_h} \quad (4)$$

其中, (b_x, b_y, b_w, b_h) 为目标边界框在特征图中的中心点坐标及宽高; (t_x, t_y, t_w, t_h) 为网络预测的边界框中心点及宽高偏移; (c_x, c_y) 为目标中心点所在网格单元左上角在特征图中的位置; (p_w, p_h) 为与目标边界框最匹配的锚框宽高; σ 表示 Sigmoid 函数。

图 2 YOLOv3 目标框与锚框的关系

Keras 版本 YOLOv3 的损失函数为

$$L_{\text{YOLOv3}} = -\lambda_{\text{coord}} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{\text{obj}}^{ij} (2 - \hat{w}_i \times \hat{h}_i) \left[\sigma(\hat{t}_{x_i}) \ln(\sigma(t_{x_i})) + (1 - \sigma(\hat{t}_{x_i})) \ln(1 - \sigma(t_{x_i})) + \sigma(\hat{t}_{y_i}) \ln(\sigma(t_{y_i})) + (1 - \sigma(\hat{t}_{y_i})) \ln(1 - \sigma(t_{y_i})) \right] + \lambda_{\text{coord}} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{\text{obj}}^{ij} (2 - \hat{w}_i \times \hat{h}_i) \left[(t_{w_i} - \hat{t}_{w_i})^2 + (t_{h_i} - \hat{t}_{h_i})^2 \right] - \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{\text{obj}}^{ij} \left[\hat{C}_i \ln(C_i) + (1 - \hat{C}_i) \ln(1 - C_i) \right] - \lambda_{\text{noobj}} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{\text{noobj}}^{ij} \left[\hat{C}_i \ln(C_i) + (1 - \hat{C}_i) \ln(1 - C_i) \right] - \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{\text{obj}}^{ij} \left[\hat{p}_i \ln(p_i) + (1 - \hat{p}_i) \ln(1 - p_i) \right] \quad (5)$$

其主要包括中心点坐标损失、宽高损失、置信度损失和类别概率损失 4 部分, 其宽高损失使用均方误差, 其余使用交叉熵计算。其中, λ_{coord} 表示坐标损失权重, 是一个动态值, 当待检测目标尺寸较大时, λ_{coord} 较小, 反之则较大; λ_{noobj} 表示不负责预测目标的网格单元的损失权重, 由于负责预测目标的网格单元的数量远小于不负责预测目标的网

格单元数量, 因此 λ_{noobj} 一般较小; 对于第 i 个网格的第 j 个锚框, 若其负责检测某个目标, 则 $I_{\text{obj}}^{ij} = 1$, $I_{\text{noobj}}^{ij} = 0$, 反之 $I_{\text{obj}}^{ij} = 0$, $I_{\text{noobj}}^{ij} = 1$; $S \times S$ 为输出层特征图尺寸; B 为特征图中每个网格单元产生的边界框数目; $(\hat{w}_i, \hat{h}_i)$ 为真实边界框宽高在图像中的相对值; $(\hat{t}_{x_i}, \hat{t}_{y_i}, \hat{t}_{w_i}, \hat{t}_{h_i})$ 为真实边界框的中心点及宽高偏移; $(t_{x_i}, t_{y_i}, t_{w_i}, t_{h_i})$ 为网络预测的边界框中心点及宽高偏移; $(\hat{C}_i, C_i)$ 分别为真实置信度与预测置信度; $(\hat{p}_i, p_i)$ 分别为真实类别概率与预测类别概率。

YOLOv3 突出的性能表现使得其在目标检测中得到了广泛的应用^[22-24]。尽管如此, 与其他基于深度学习技术的目标检测算法一样, YOLOv3 仍然使用矩形检测框来定位目标。对于近似正面拍摄的车牌图像, 该算法能较好地实现车牌定位; 但是当车牌存在一定程度倾斜时, 使用矩形框所定位到的车牌区域会包含较多冗余信息, 导致后续车牌识别过程难度增大, 影响其识别准确率。

2 本文算法

为了能够充分地利用 YOLOv3 在目标检测中的优势, 同时扩展其应对倾斜车牌的定位能力, 提出一种 YOLOv3 与顶点偏移估计相结合的算法。该算法在保持 YOLOv3 原有矩形边界框预测的基础上, 并行预测车牌顶点偏移量, 从而实现了车牌快速精准定位。

由前述内容可知, 原始 YOLOv3 目标检测算法使用矩形框实现目标区域定位, 而在自然场景特别是非约束场景中的车牌, 通常呈现出非规则四边形的形态, 导致使用锚框机制的 YOLOv3 算法无法直接得到不规则四边形的顶点坐标, 即无法实现车牌区域精准定位。但若知道矩形框的 4 个顶点坐标以及车牌区域的 4 个顶点相对矩形框的 4 个顶点的偏移量, 便可以确定车牌区域的 4 个顶点, 从而间接地实现车牌区域精定位。根据这一思想, 本文提出了 YOLOv3 与顶点偏移估计相结合的车牌定位算法, 算法整体框图如图 3 所示。

本文算法中车牌顶点位置与 YOLOv3 检测框的关系图如图 4 所示, 图中蓝色矩形框为车牌粗定位区域, 黑色虚线框为锚框, 橙色四边形实线框为真实车牌区域; 紫色、灰色、黄色和绿色箭头线段分别由车牌粗定位区域的左上、右上、右下、左下

图3 本文算法整体框图

图4 本文算法车牌顶点位置与矩形检测框的关系

顶点指向各自对应方位实际车牌顶点 V_1, V_2, V_3, V_4 的偏移。 $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ 分别为左上、右上、右下、左下对应顶点的水平与垂直偏移分量。通常, 2 个点相对位置方位关系有 4 种情况, 因此点与点之间的坐标偏移值会存在正负之分, 但由于本文算法中实际车牌 4 个顶点相对于车牌粗定位框 4 个顶点的相对位置方位是一定的, 因此设定所有的偏移值均为非负数。本文在原始 YOLOv3 的基础上增加了 4 组偏移值的预测, 以实现车牌区域的精确定位。

本文算法包括原始 YOLOv3 的目标边界框预测及新增的 4 组顶点偏移值预测 2 部分, 边界框预测如式(1)~式(4)所示。由于顶点偏移值代表的是对应顶点间的距离, 因此使用与边界框宽高相同的预测方式, 计算公式为

$$x_1 = p_w \times e^{t_{x_1}}, y_1 = p_h \times e^{t_{y_1}};$$

$$x_2 = p_w \times e^{t_{x_2}}, y_2 = p_h \times e^{t_{y_2}};$$

$$x_3 = p_w \times e^{t_{x_3}}, y_3 = p_h \times e^{t_{y_3}};$$

$$x_4 = p_w \times e^{t_{x_4}}, y_4 = p_h \times e^{t_{y_4}}.$$

其中, $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ 分别为左上、右上、右下和左下顶点的水平与垂直偏移分量; $(t_{x_1}, t_{y_1}), (t_{x_2}, t_{y_2}), (t_{x_3}, t_{y_3}), (t_{x_4}, t_{y_4})$ 分别为网络预测的 4 个顶点偏移分量的偏移; (p_w, p_h) 为与目标边界框最匹配的锚框宽高。

由于算法新增了对 4 组偏移的预测, 损失函数需要增加顶点偏移损失 L_{Vertex} 部分。偏移量损失使用均方误差计算, 计算公式为

$$L_{\text{Vertex}} = \lambda_{\text{coord}} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{\text{obj}}^{ij}.$$

$$\begin{aligned} & \left\{ (2 - \hat{x}_1^i \times \hat{y}_1^i) \left[(\hat{t}_{x_1}^i - t_{x_1}^i)^2 + (\hat{t}_{y_1}^i - t_{y_1}^i)^2 \right] + \right. \\ & (2 - \hat{x}_2^i \times \hat{y}_2^i) \left[(\hat{t}_{x_2}^i - t_{x_2}^i)^2 + (\hat{t}_{y_2}^i - t_{y_2}^i)^2 \right] + \\ & (2 - \hat{x}_3^i \times \hat{y}_3^i) \left[(\hat{t}_{x_3}^i - t_{x_3}^i)^2 + (\hat{t}_{y_3}^i - t_{y_3}^i)^2 \right] + \\ & \left. (2 - \hat{x}_4^i \times \hat{y}_4^i) \left[(\hat{t}_{x_4}^i - t_{x_4}^i)^2 + (\hat{t}_{y_4}^i - t_{y_4}^i)^2 \right] \right\} \quad (6) \end{aligned}$$

其中, $(\hat{x}_1^i, \hat{y}_1^i), (\hat{x}_2^i, \hat{y}_2^i), (\hat{x}_3^i, \hat{y}_3^i), (\hat{x}_4^i, \hat{y}_4^i)$ 分别为 4 个顶点的真实偏移分量在图像中的相对值; $(\hat{t}_{x_1}^i, \hat{t}_{y_1}^i), (\hat{t}_{x_2}^i, \hat{t}_{y_2}^i), (\hat{t}_{x_3}^i, \hat{t}_{y_3}^i), (\hat{t}_{x_4}^i, \hat{t}_{y_4}^i)$ 分别为 4 个顶点的真实偏移分量的偏移; $(t_{x_1}^i, t_{y_1}^i), (t_{x_2}^i, t_{y_2}^i), (t_{x_3}^i, t_{y_3}^i), (t_{x_4}^i, t_{y_4}^i)$ 分别为网络预测的 4 个顶点偏移分量的偏移; $S \times S, B, I_{\text{obj}}^{ij}, \lambda_{\text{coord}}$ 与式(5)中参数含义一致。

本文算法的损失函数 L 包括原始 YOLOv3 损失 L_{YOLOv3} 和顶点偏移损失 L_{Vertex} 2 个部分, 具体计算公式为 $L = L_{\text{Vertex}} + L_{\text{YOLOv3}}$ 。

3 实验结果和分析

3.1 数据准备

为验证基于顶点偏移车牌定位算法的性能, 本文选用目前使用最为广泛的非约束环境下拍摄的大规模中国城市停车数据集 (Chinese City Parking Dataset, CCPD)^[25] 作为实验数据集, 共约 35 万张图片, 包含 Base, Blur, Challenge, DB, FN, Ro-

tate, Tilt, Weather, NP 共 9 个数据子集, 分别为各种场景下获取的车牌图像, 各子集的描述如表 1 所示. 其中, Base 中为多种场景混合图像, NP 中不包含车牌图像, 其他子集均为单一场景图像. CCPD 包含车牌的外接矩形框、顶点位置和车牌号码等多个标注信息, 其图像尺寸均为 1160×720 . CCPD 数据集有 2 个版本, 实验使用的是新版本, 其相对于旧版本数据量增加了约 10 万张图片, 各子集做了一些更新并增加了子集 Blur, 新版数据集相比于旧版数据集更具有挑战性.

表 1 CCPD 数据集描述

子集	描述
Base	各种场景下的车牌图像
Blur	模糊的车牌图像
Challenge	具有挑战性的车牌图像
DB	光照过强或过暗的车牌图像
FN	拍摄距离过远或过近的车牌图像
Rotate	旋转的车牌图像
Tilt	倾斜的车牌图像
Weather	雨、雪、雾等天气获取的车牌图像
NP	不包含车牌的车辆图像

一般来说, 数据集的选取方式会在不同程度上影响模型的训练及泛化能力, 合理选取训练数据集不仅能增强模型的泛化能力, 还能提升模型精度. 通过分析 CCPD 文献可以发现, 其选取 Base 子集中的一半进行训练, 但 Base 中各类场景中的图片数量并不相同, 这会造成训练数据类型不均衡. 为了更加合理地组织训练数据, 本文从 CCPD 数据集中除 Base 和 NP 外的 7 个单一场景子集中按照一定的方式抽取了 66 000 张图片, 制作了一个各种场景图片比例相对均匀、但规模相对小的车牌训练数据集, 其中训练集和验证集的图片数量分别为 60 000 和 6 000 张.

表 2 所示为实验所使用的数据集选取方式, 其中“总数量”一栏为 7 种单一场景子集包含的图片数量, 数据规模最大的子集和最小的子集分别为 Challenge 和 Weather, 对应的图片数量为 50 003 张和 9 999 张, 两者在规模上具有较大的差异. 为了保证训练集和验证集的图片数量达到一定的规模, 未对每个单一场景子集均衡抽取图片. 7 种单一场景子集中除用于训练集和验证集以外的全部图片均作为测试集. 由表 2 可以看出, 从前 7 个数据集抽取“训练+验证集”的图片总数为 97 448 张, 但各子集间存在部分重复图片, 应去除多余的重复图

片, 为了获得更好的模型性能, 将剩余图片中车牌顶点与图片边缘距离小于 60 像素的图像去掉, 最终获取 66 000 张图片用于模型训练和验证. 7 个子集中用于测试集的图片共 54 533 张, 去除多余重复图片后为 44 486 张; 另外, 从 Base 子集中随机抽取 100 000 张图片进行测试.

表 2 训练与测试数据集

子集	训练+验证集	测试集	总数量
DB	8 832	1 300	10 132
FN	12 164	8 803	20 967
Rotate	8 832	1 221	10 053
Tilt	15 718	14 498	30 216
Weather	8 868	1 131	9 999
Challenge	29 591	20 412	50 003
Blur	13 443	7 168	20 611
合计 1	97 448	54 533	
合计 2	66 000	44 486	

相对均衡的训练数据准备完毕后, 还需对数据进行标注. 图 5 展示了本文算法的数据标签信息格式, 包含图片路径和目标信息 2 部分, 目标信息中 (x_{11}, y_{11}) 和 (x_{22}, y_{22}) 分别为目标边界框左上顶点和右下顶点坐标, 为一定程度上保证车牌顶点在目标边界框内且便于利用偏移量定位, 训练数据使用的目标边界框是在车牌外接矩形框的基础上向四周扩大 20 个像素得到的, 这也是在选取训练图片时剔除了车牌顶点距离图片边缘小于 60 个像素的图片的原因. (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) 分别表示车牌左上顶点坐标、右上顶点坐标、右下顶点坐标和左下顶点坐标; d_0 为类别编号.

图 5 本文算法标签信息格式

3.2 模型训练

本实验的相关配置环境为 Ubuntu18.04LTS, Python3.6.9, Keras2.20, Tensorflow1.14.0, GPU 为 TITAN X, CPU 为 Intel(R) Xeon(R) CPU E5-2620 v4 @2.10GHz, 内存为 96GB, Cuda10.0, 模型详细的训练参数如表 3 所示.

表3 网络训练参数

参数名称	参数值
Epochs	31
Batch Size	8
Input Size	320×320
Learning Rate	10^{-5}
Anchors	147, 85, 196, 148, 202, 106, 229, 185, 254, 124, 264, 224, 316, 148, 333, 268, 421, 195

网络训练步骤如下:

Step1. 数据预处理. 主要包括按照第3.1节所述方式从CCPD车牌数据集中抽取训练数据, 不对数据使用任何增广策略.

Step2. 数据集划分. 将数据集划分为训练集、验证集以及测试集.

Step3. 网络训练. 设置网络参数, 如学习率、训练次数和锚框大小等, 并将准备好的数据放入网络进行训练.

Step4. 网络测试与评估. 利用训练好的网络检测测试集, 然后根据输出结果计算相关评价指标, 评估网络性能; 若性能达不到预期, 转Step3; 否则, 结束.

3.3 实验结果

文献[25]将CCPD数据集中的Base子集等分为2部分, 每部分约有10万张图片, 其中一部分用作网络训练, 另一部分用于网络评估. 另外, 作者还对除NP以外的6个单一场景子集进行了评估. Base子集中的图片虽然由多种场景混合而成, 但

是每种场景下的图片比例不确定, 所以本文更加有针对性地选择训练图片, 即从除Base和NP以外的7个子集中抽取训练集和验证集, 并将剩下的图片用作测试集. 为了对比方便, 额外测试了10万张Base子集图片.

本文算法对非约束环境下车牌的检测效果如图6所示, 其中外围红色矩形框为车牌粗定位区域, 黄色实线表示车牌4个顶点与粗定位框4个顶点对应的偏移关系, 内部的红色四边形框为车牌精定位区域, 这是通过粗定位框与偏移量计算得到的. 图6分别列出了倾斜角度大、拍摄距离远、模糊和光照过亮或过暗4种复杂的非约束场景下的车牌定位效果. 可以看出, 本文算法对各种复杂场景下的车牌均具有良好的检测效果, 充分证明了所提出的顶点偏移估计方案是合理的.

为进一步说明融合了顶点偏移估计的YOLOv3算法可有效地提升车牌定位的精准程度, 本文对改进前后的算法效果进行对比, 如图7所示. 从图7可以看出, 原始YOLOv3算法检测到的车牌区域中包含部分非车牌冗余信息, 而YOLOv3与顶点偏移估计相结合的算法能更精准地定位车牌区域, 便于后续车牌识别. 这充分说明了本文算法具有一定的优越性.

图6 本文算法对非约束环境场景下车牌图像的检测结果

本文通过检测车牌4个顶点来实现车牌区域精定位, 由于得到的车牌区域为非规则四边形, 因此无法使用传统的通过矩形框计算交并比(intersection over union, IoU). 为保证后续对比实验的公平性, 使用图8中的IoU计算方法, 即计算车牌精定位区域

的外接矩形框与真实车牌外接矩形框的IoU. 同文献[25]中使用的标准一样, 当 $\text{IoU} > 0.70$ 时, 判定当前目标检测正确.

为验证顶点偏移算法的可行性, 采用2种不同的方式对数据进行了训练, 分别是YOLOv3与顶点

a. YOLOv3 车牌检测效果

b. YOLOv3 与顶点偏移估计结合车牌检测效果

图 7 YOLOv3 与本文算法的车牌检测效果

偏移估计相结合的算法, 以及直接使用 YOLOv3 算法进行训练. 同时, 为定量地评估算法的性能, 本文对算法在 Base 子集以及除 NP 外的 7 种单一场景中的车牌定位精度进行了统计, 同时计算了算法在整个测试集上的平均精度 AP.

由表 4 可知, YOLOv3 与顶点偏移估计相结合的算法在各个子集上的精度均优于原始 YOLOv3 算法. 另外, 结合了顶点偏移估计的算法对车牌定位的精准程度更高, 包含的冗余区域要远小于不采用顶点偏移估计的算法. 为了进一步对算法进行评估, 本文还与文献[25]中的相关数据进行了对比. 从表 4 可以看出, 本文算法无论是在 Base 子集还是 7 种单一场景子集上, 其精度均高于文献[25]中测试的 5 个算法, 即表 4 中的前 5 个算法. 同时, 本文算法在 Tilt, Weather 及 Base 子集上的精度达到 99.00% 以上, 证明了本文算法的有效性. 尽管本文算法在 7 种单一场景子集上的精度普遍较高, 但对 DB 和 FN 场景下的车牌定位效果相对较差, 且在其他检测算法中也出现了同样的情况. 因此, 可以重点从 DB 和 FN 这 2 个场景入手改善车牌定位算法的性能. 另外, Base 子集的精度高达 99.98%, 说明相对 7 个单一场景子集而言, Base 子集图片中的车牌定位难度是最小的.

使用 Base 子集作训练数据时不利于提升算法在各种非约束环境下的车牌定位性能. 而本实验

图 8 本文 IoU 计算方式

表 4 不同算法精度对比分析

算法	AP	Base(100k)	Challenge	DB	FN	Rotate	Tilt	Weather	Blur
SSD300 ^[26]	94.40	99.10	93.10	89.20	84.70	95.60	94.90	83.40	
YOLO9000 ^[27]	93.10	98.80	88.60	89.60	77.30	93.30	91.80	84.20	
Faster R-CNN ^[15]	92.90	98.10	83.90	92.10	83.70	91.80	89.40	81.80	
TE2E ^[28]	94.20	98.50	93.10	91.70	83.80	95.10	94.50	83.60	
RPnet ^[25]	94.50	99.30	92.80	89.50	85.30	94.70	93.20	84.10	
YOLOv3 ^[21]	99.13	99.94	97.74	94.35	91.04	97.87	99.00	99.73	98.60
本文	99.32	99.98	98.21	95.35	93.08	98.44	99.27	99.82	98.91

从除 Base 和 NP 外的 7 种单一场景子集中抽取图片用于训练, 有效地避免了各种场景图片比例悬殊的问题。尽管本文使用的训练集图片数目小于文献[25]所使用的训练集, 但是算法精度却高于表 4 中的其他算法, 充分说明了本实验选择训练集方案的合理性。除了精度高外, 本文算法仅对 YOLOv3 的输出层进行扩展, 其他网络层并没有增加额外的计算量, 因此能够实现实时车牌精确定位, 在 GTX TITAN X 硬件测试环境中, 对输入尺寸为 320×320 像素的图片, 所需检测时间为 0.031 s。

为进一步验证顶点偏移估计算法的性能, 本文对算法在不同 IoU 阈值下的精度进行了统计, 如表 5 所示。将表 5 中的结果与表 4 中阈值设定为 0.70 的其他算法比较, 可以看出本文算法在阈值为 0.75 时仍然具有较明显的优势; 当阈值为 0.80 时, 算法在 Base, Rotate, Tilt, Weather 子集上精度

仍然具有一定的竞争力, 且其整体精度能达到 97.15%; 当阈值增加到 0.85, 0.90 时, 车牌定位精度明显下降, 这主要是因为阈值设置较大时, 对算法定位的精准程度要求更高。一般来说, 目标检测训练集标签信息的精准程度也只是相对的, 所以算法检测时允许检测到的目标区域与实际标注区域存在一定的误差, 通常选取 $\text{IoU} > 0.70$ 。而将本文算法在 0.75, 0.80 阈值下与表 4 其他算法在 0.70 阈值下的精度进行对比, 仍然占一定优势。

在利用 IoU 对车牌检测正确性进行评判时, 本文选取了车牌的外接矩形框进行 IoU 计算, 以便与其他算法进行比较。但本文算法的最大优势在于能对车牌 4 个顶点进行定位, 经过简单的几何变换就能得到精准的车牌区域, 便于后续车牌字符识别。而直接使用矩形框的定位结果包含更多冗余信息, 还需要进一步精定位才能进行车牌识别。

表 5 本文算法在不同 IoU 阈值下的精度

IoU	AP	Base(100k)	Challenge	DB	FN	Rotate	Tilt	Weather	Blur
0.75	98.76	99.92	96.18	92.25	88.95	97.89	98.45	99.38	97.20
0.80	97.15	99.57	90.53	85.90	79.64	95.99	95.81	99.12	92.62
0.85	91.16	96.37	75.63	69.09	61.30	90.58	86.77	94.78	78.62
0.90	61.02	66.84	39.87	32.46	33.91	65.37	59.70	67.46	43.04

本文与其他算法最大的不同在于 YOLOv3 与顶点偏移估计相结合的方案能够较为准确地预测车牌的 4 个顶点坐标。尽管精度指标能够从整体上评价算法定位车牌区域的效果好坏, 但是无法衡量预测出的车牌顶点坐标与实际的车牌顶点之间的误差。因此, 通过对全部检测正确的图片计算顶点坐标的预测值与真实值之间的欧几里得度量的平均值, 得到了表 6 中的相关数据。

表 6 坐标预测值与真实值的误差像素

图像尺寸	顶点方位				平均误差
	左上	右上	右下	左下	
原始图像尺寸 1160×720	6.06	6.40	6.42	6.68	6.39
网络输入尺寸 320×320	1.67	1.77	1.77	1.84	1.76

表 6 列出了车牌 4 组顶点坐标分别在原始图像尺寸及网络输入尺寸下的误差, 该误差使用欧几里得度量计算。从表 6 可以看出, 4 个方位顶点在原始图像中的误差均在 6~7 个像素, 其中误差最小和最大的分别为左上与左下 2 个方位的顶点, 其差距仅为 0.62 个像素。虽然各方位顶点的误差不相同, 但将误差转换到网络输入尺寸下以后, 各顶点

偏移误差不超过 2 个像素, 且此时网络在预测各顶点偏移时产生的误差几乎一致。因此, 对于车牌 4 个顶点, 网络预测效果基本相同, 并无明显偏差, 也说明本文算法在车牌精定位上具有良好的性能。

3.4 异常检测情况分析

图 9 所示为本文算法定位车牌有误的结果示例, 主要分为 3 种情况。

(1) 定位结果偏大, 即包含较多冗余的背景信息, 从图 9a 可以看出, 这主要是由于车牌与背景颜色非常相似、光照条件较差和倾斜严重等导致车牌边界信息不明显, 从而无法获得准确的车牌区域。

(2) 定位结果偏小, 即仅得到车牌的部分区域, 从图 9b 可以很明显看出, 定位区域偏小的车牌主要是受到恶劣的光照条件及抖动模糊因素的影响, 导致车牌部分区域的特征极为不明显, 模型无法提取有效的车牌特征。同时, 也可以发现, 当车牌粗定位区域接近或小于车牌外接矩形时, 更容易导致车牌定位不准确, 说明将车牌外接矩形框扩展后作为车牌粗定位区域是十分必要的。

(3) 未检测到车牌, 如图 9c 所示, 这类图像主要是在光照条件极差、车牌受到一定程度遮挡等条件下获取的, 车牌特征受到严重影响而不明显。

a. 定位结果偏大

b. 定位结果偏小

c. 未检测到车牌

图 9 顶点定位有误的车牌图像

由上述分析可知, 本文模型在背景颜色与车牌颜色非常相似、光照条件极差及模糊程度高等极端的非约束环境下的车牌定位效果还有一定的提升空间。

CCPD 数据集中由顶点标签得到的车牌区域如图 10 所示, 其中图 10a 为顶点标签较准确的车牌图像, 图 10b 为标注存在一定误差的图像。在整个 CCPD 数据集中, 标注信息不准确的图像占一

a. 标注准确

b. 标注不准确

图 10 数据集标签可视化示例

定比例, 这会对模型训练产生一定的影响, 使得最终得到的模型权重对某些图像的检测结果较差, 从而影响模型整体精度。因此, 使用更精准的标签信息也能在一定程度上提升模型的检测效果。

4 结 语

针对现有以矩形检测框为输出的目标检测算法无法准确定位非约束环境下车牌区域的问题, 提出一种 YOLOv3 与顶点偏移估计结合的车牌精定位算法, 其在几乎不增加额外计算量的情况下, 通过扩展 YOLOv3 网络输出, 增加车牌顶点相对于矩形边界框角点偏移的预测, 实现车牌精定位。

本文算法虽然在一定程度上提升了非约束环境下车牌定位的精度, 但是仍然有一定的提升空间, 如车牌倾斜角度过大、光照条件极差、车牌与车辆或图片背景十分相似等情况时, 会导致车牌整体特征不明显以及车牌部分特征丢失。因此, 对该类型的车牌区域进行精定位仍然存在一定的困难。在接下来的工作中, 可以通过改进网络结构、使用更准确的数据标签等措施, 提升模型在上述难度较大的场景下的车牌定位性能。

参考文献(References):

- [1] Yang Dingding, Chen Shiqiang, Liu Jingyi. License plate location algorithm based on background and character color feature[J]. Computer Applications and Software, 2018, 35(12): 216-221(in Chinese)
(杨鼎鼎, 陈世强, 刘静漪. 基于车牌背景和字符颜色特征的车牌定位算法[J]. 计算机应用与软件, 2018, 35(12): 216-221)
- [2] Zeng Lihua, Li Chao, Xiong Zhang. Precisely license plate locating algorithm based on edge and color information[J]. Journal of Beijing University of Aeronautics and Astronautics, 2007, 33(9): 1112-1116(in Chinese)
(曾丽华, 李超, 熊璋. 基于边缘与颜色信息的车牌精确定位算法[J]. 北京航空航天大学学报, 2007, 33(9): 1112-1116)
- [3] Pan Yong, Zhao Jiale, Xu Zhigang, et al. An algorithm of license plate location and recognition in low illuminance environment based on texture feature[J]. Journal of Highway and Transportation Research and Development, 2015, 32(7): 140-148(in Chinese)
(潘勇, 赵佳乐, 徐志刚, 等. 基于纹理特征的低照度环境下车牌定位与识别算法[J]. 公路交通科技, 2015, 32(7): 140-148)
- [4] Sun Hong, Guo Kai. License plate location based on characters texture feature and RGB color feature[J]. Opto-Electronic Engineering, 2015, 42(6): 14-19+44(in Chinese)
(孙红, 郭凯. 融合字符纹理特征与 RGB 颜色特征的车牌定位[J]. 光电工程, 2015, 42(6): 14-19+44)

- [5] Fei Jiyou, Xie Jinlu, Li Hua, *et al.* Adaptive license plate correction and extraction based on character feature constraint[J]. Chinese Journal of Scientific Instrument, 2016, 37(3): 632-639(in Chinese)
(费继友, 谢金路, 李花, 等. 基于字符特征约束的自适应车牌校正提取[J]. 仪器仪表学报, 2016, 37(3): 632-639)
- [6] Ha P S, Shakeri M. License plate automatic recognition based on edge detection[C] //Proceedings of the Artificial Intelligence and Robotics. Los Alamitos: IEEE Computer Society Press, 2016: 170-174
- [7] Li Zhixin, Zhou Tao, Zhang Canlong, *et al.* Improve multi-label image classification using adversarial network[J]. Journal of Computer-Aided Design & Computer Graphics, 2020, 32(1): 16-26(in Chinese)
(李志欣, 周韬, 张灿龙, 等. 利用对抗网络改进多标记图像分类[J]. 计算机辅助设计与图形学学报, 2020, 32(1): 16-26)
- [8] Tong G F, Chen H R, Li Y, *et al.* Object detection for panoramic images based on MS-RPN structure in traffic road scenes[J]. IET Computer Vision, 2019, 13(5): 500-506
- [9] Cao Xuehong, Gao Mengfei, Fan Jiulun, *et al.* Cutset-type possibilistic c-mean image segmentation algorithm based on nonlocal spatial information[J]. Journal of Xi'an University of Posts and Telecommunications, 2019, 24(4): 43-49(in Chinese)
(曹雪红, 高梦飞, 范九伦, 等. 基于非局部信息的截集式 PCM 图像分割算法[J]. 西安邮电大学学报, 2019, 24(4): 43-49)
- [10] Liu C S, Chang F L. Hybrid cascade structure for license plate detection in large visual surveillance scenes[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(6): 2122-2135
- [11] Yonetsu S, Iwamoto Y, Chen Y W. Two-stage YOLOv2 for accurate license-plate detection in complex scenes[C] //Proceedings of the IEEE International Conference on Consumer Electronics. Los Alamitos: IEEE Computer Society Press, 2019: 1-4
- [12] Kim S G, Jeon H G, Koo H I. Deep-learning-based license plate detection method using vehicle region extraction[J]. Electronics Letters, 2017, 53(15): 1034-1036
- [13] Zhang J J, Li Y Y, Li T, *et al.* License plate localization in unconstrained scenes using a two-stage CNN-RNN[J]. IEEE Sensors Journal, 2019, 19(13): 5256-5265
- [14] Li Xiangpeng, Min Weidong, Han Qing, *et al.* License plate location and recognition based on deep learning[J]. Journal of Computer-Aided Design & Computer Graphics, 2019, 31(6): 979-987(in Chinese)
(李祥鹏, 闵卫东, 韩清, 等. 基于深度学习的车牌定位和识别方法[J]. 计算机辅助设计与图形学学报, 2019, 31(6): 979-987)
- [15] 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
- [16] Hsu G S, Chen J C, Chung Y Z. Application-oriented license plate recognition[J]. IEEE Transactions on Vehicular Technology, 2013, 62(2): 552-561
- [17] Masood S Z, Shu G, Dehghan A, *et al.* License plate detection and recognition using deeply learned convolutional neural networks[OL]. [2020-08-02]. <https://arxiv.org/abs/1703.07330>
- [18] Li H X, Yang R, Chen X H. License plate detection using convolutional neural network[C] //Proceedings of the IEEE International Conference on Computer and Communications. Los Alamitos: IEEE Computer Society Press, 2017: 1736-1740
- [19] Xie L L, Ahmad T, Jin L W, *et al.* A new CNN-based method for multi-directional car license plate detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(2): 507-517
- [20] Silva S M, Jung C. License plate detection and recognition in unconstrained scenarios[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 593-609
- [21] Redmon J, Farhadi A. YOLOv3: an incremental improvement[OL]. [2020-08-02]. <https://arxiv.org/abs/1804.02767>
- [22] Wei Hongbin, Zhang Ruijin, Du Guangming, *et al.* Vegetable detection algorithm based on improved YOLOv3[J]. Journal of Zhengzhou University: Engineering Science, 2020, 41(2): 7-12(in Chinese)
(魏宏彬, 张瑞金, 杜广明, 等. 基于改进型 YOLOv3 的蔬菜识别算法[J]. 郑州大学学报: 工学版, 2020, 41(2): 7-12)
- [23] Guo Yijing, Huang Siqu, Liu Li, *et al.* Automatic tableware sorting detection based on YOLO v3[J]. Journal of Xiamen University: Natural Science, 2020, 59(4): 572-577(in Chinese)
(郭一晶, 黄斯奇, 刘丽, 等. 基于 YOLOv3 的自动餐具分拣检测[J]. 厦门大学学报: 自然科学版, 2020, 59(4): 572-577)
- [24] Wang Yuning, Pang Zhiheng, Yuan Deming. Vehicle detection based on YOLO in real time[J]. Journal of Wuhan University of Technology, 2016, 38(10): 41-46(in Chinese)
(王宇宁, 庞智恒, 袁德明. 基于 YOLO 算法的车辆实时检测[J]. 武汉理工大学学报, 2016, 38(10): 41-46)
- [25] Xu Z B, Yang W, Meng A J, *et al.* Towards end-to-end license plate detection and recognition: A large dataset and baseline[C] //Proceedings of the European conference on computer vision. Heidelberg: Springer, 2018: 261-277
- [26] Liu W, Anguelov D, Erhan D, *et al.* SSD: single shot multibox detector[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2016: 21-37
- [27] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6517-6525
- [28] Li H, Wang P, Shen C H. Toward end-to-end car license plate detection and recognition with deep neural networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(3): 1126-1136