







图2 Dense-YOLOv3 模型

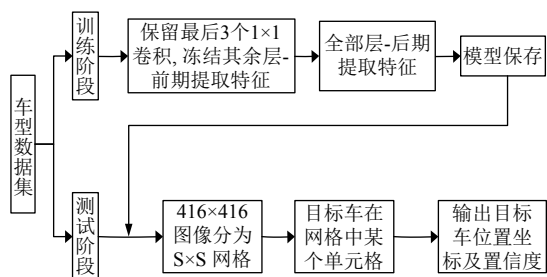


图3 Dense-YOLOv3 模型流程图

算法具体步骤如下。

(1) 数据预处理

输入: BIT-Vehicle 原始数据集

输出: 已标注训练集 (目标物体车的位置、类别)

xml 文件和未标注的测试集

Step 1. 在 BIT-Vehicle 车型数据集 (如图 4) 中挑选具有代表性即不同光照强度和不同拍摄距离的图片, 随机划分训练集和测试集;

Step 2. 将训练集用 labelImg 工具进行标注形成 xml 文件;

Step 3. 利用代码形式训练集划分为两部分即 train 集和 val 集;

Step 4. 保留测试集图片。



图4 车型数据集

(2) 网络训练

输入: 已标注训练集 xml 文件与对应图片

输出: 训练权重模型

Step 1. 配置 cfg 文件, 设置学习率、类别数、batch 和 subdivisions 等超参数;

Step 2. 前期阶段, 冻结部分层进行训练;

Step 3. 一个 xml 文件经过卷积层、最大池化层和一系列密集卷积块与转移层提取车型特征;

Step 4. 将 Step3 中提取特征经过三个尺度的检测框进行特征局部交互;

Step 5. 计算出本文模型输出与真实值  $y\_true$  之间的损失值;

Step 6. 返回 Step 3, 在训练过程中不断调整网络中的参数, 使得损失函数达到一个前期稳定的收敛值, 控制前期迭代次数结束前期训练;

Step 7. 开始后期阶段, 取消冻结层将进行全部层数的训练;

Step 8. 重复 Step 3~ Step 5, 达到迭代次数停止训练.

### (3) 网络测试

输入: 测试集、训练权重模型

输出: 目标车位置及类别

Step 1. 将测试集图片放入网络中;

Step 2. 进行前向传播, 得到预测框与相应的类别;

Step 3. 将网络预测框与真实框使用 IOU 技术得到接近真实框的预测框, 如图 5 所示.



图 5 IOU 技术展示

Step 4. Step 3 中的预测框与相应的类别求积得到置信度;

Step 5. 设置阈值, 过滤到低分的预测框, 对剩余的预测框进行 NMS 处理, 得到最后目标物体车结果;

Step 6. 循环 Step 1~Step 5, 直到预测完测试集即测试结束.

## 2 实验分析与结果

### 2.1 实验环境与数据集正文内容

硬件环境: 在系统类型为 Windows10 64 位操作系统和基于 x64 的处理器、计算机 CPU 型号为 Intel(R)

Xeon(R) W-2102 CPU@ 2.90 GHz、安装 RAM 的内存 16.0 GB, 以及使用 GPU device 0: GeForce RTX 2080 Ti, 设备上 68 个多处理器、全局内存总量为 11 264 MB, 每个线程块共享内存 48 KB、每个线程块最大线程数为 1024 上进行实验的运行.

软件环境: 采用基于 Theano 的深度学习框架 Keras, 是一个高度模块化的神经网络库, 简化了编程复杂度. 该框架有三大特性: 在操作方面建模简单; 配置简单; 相比其他框架, 对于不同模型算法, 微调步骤简单. 另外, 在算力上, 从 CPU 到 GPU 上进行加速不需要任何代码的改动, 且性能没有太多的损耗.

本文用于训练与测试数据集选取了少部分研究学者们经常实验用的 BIT-Vehicle 车型数据集, 一共有 6 种类型的车型. 该库中的图像均来自于实际路段的卡口高清图像, 与实际路况非常吻合, 且受拍摄距离、光照强度以及天气等的影响而拍摄到的多路况的车型图像, 更能突出深度卷积神经网络在检测效率上的优越性. 本次实验将每类车型数据集分为两组, 一组数据用来训练, 另一组用来测试. 在 BIT-Vehicle 车辆数据库中, 用于训练的车型集共有 5404 张, 按照类别 Bus、Microbus、Minivan、Sedan、SUV 和 Truck 划分为 810、700、748、1357、845 和 944 张; 用于模型测试有 3603 张车型集, 将其按类别排列顺序划分为 601、520、511、765、600 和 606 张车型图像. 具体车型数据集分配如表 2 所示.

表 2 实验数据分配

| 类别       | 训练集  | 测试集  | 总数   |
|----------|------|------|------|
| Bus      | 810  | 601  | 1411 |
| Microbus | 700  | 520  | 1220 |
| Minivan  | 748  | 511  | 1259 |
| Sedan    | 1357 | 765  | 2122 |
| SUV      | 845  | 600  | 1445 |
| Truck    | 944  | 606  | 1550 |
| 总数       | 5404 | 3603 | 9007 |

### 2.2 实验分析与对比

为了更好地验证 Dense-YOLOv3 在车型数据集上的网络性能和使用价值, 文中通过采用不同算法定性分析对比与实验数据的定量分析对比两部分进行研究.

#### 2.2.1 定性分析

在本文中, 分别采用文献 [13]、YOLOv3 增加 4 层网络<sup>[1]</sup>和 Dense-YOLOv3 模型在已经准备好的测试集上进行检测, 如表 3 所示, 其中, 选取具有代表性

即拍摄距离较远且相似车辆和光线不同的图片, 分别采用单通道直方图的重合度与差异值哈希法和亮度进行定量分析. 其中 YOLOv3 增加 4 层网络算法是在 Darknet-53 网络基础上再增加 12 层网络即 4 组卷积层+残差单元共同构成的网络结构. 理论上讲, 神经网络层数越大, 其表达能力越强即提取到的车型特征信息越多, 测试结果越显著, 但从表 3 中检测结果可以看出, 文献 [1] 检测效果并没有文献 [13] 那么明显.

从主观角度判断图片中目标物体车距离较远; 从客观的角度判断图片的相似度, 其由颜色和图像指纹相结合, 分别用单通道直方图的重合度与差异值哈希法 (dHash) 来进行表示. dHash 算法都是值越小, 相似度越高, 取值为 0-64, 即汉明距离中, 64 位的 hash 值有多少不同; 单通道直方图的值为 0-1, 值越大, 相似度越高. 两者进行结合, 更能判断不同算法检测效果. 从表 4 中看出, 所选取图片重合度为 0.604、0.798 和 0.602; dHash 值为 16、12 和 14.

图片亮度表示光线暗或者曝光程度如何, 一般规定一张图片的平均亮度为 128, 对于一张正常拍照拍出来的图片平均亮度应该是在 [100, 150]. 在表 4 中选取亮度为 33.99、20.31 和 166.95 的较暗与曝光的 3 张图片进行检测研究.

从表 3 可知, 对于检测同为距离较远的车, 本文模型相比于文献 [13]、文献 [1] 模型检测准确, 且检测出正确车型概率分别为 0.96、0.99 和 0.93, 而文献 [13]、文献 [1] 模型虽然检测出有目标物体存在, 但是未能正确检测出车型或者准确率比本文模型低. 光线较暗和曝光较强车型图片, 本文模型正确检测出车型且准确率高, 而文献 [13] 模型, 虽然正确检测出了车型, 但是准确率没有本文模型高, 从表 3 中看到, 检测出多余框, 准确效率差; 文献 [1] 模型虽然在网络层数加大, 但是相较于文献 [13] 模型车型检测错误. 对于车型相似的图片, 文献 [13] 与文献 [1] 模型将 Minivan 错识别为 Truck, 而本文模型识别正确且准确率高. 总体来说, 本文模型不论在拍摄距离较远和光线较暗、较强情况下, 都能正确检测出目标物体车的类型且准确率高, 整体检测效果明显, 出现漏检的情况极少, 达到了预期的效果.

### 2.2.2 定量对比

为了进一步测试车型图片在本文模型中检测效果明显, 本文又分别利用文献 [13]、文献 [1] 和 Dense-

YOLOv3 网络模型在车型数据集上进行定量分析比较, 如表 4 所示.

在表 4 中通过采用性能指标平均精度  $AP$ 、多类别平均精度  $mAP$ 、准确率  $Precision$  和召回率  $Recall$ , 这些数值更能直观的反应出网络的性能如何. 准确率是指预测样本中的实际正样本数在所有正样本中的占比; 召回率是指预测样本中的实际正样本数占预测样本数的多少; 平均精度是由 Precision-Recall 曲线所围成的面积, 其公式为:

$$\int_0^1 p(r)dr = \sum_{k=1}^N P(k)\Delta r(k) \quad (2)$$

且本文算法各类的  $Recall$  与  $Precision$  之间的关系图如图 6 所示. 在式 (3) 中, 积分中  $p$  为准确率、 $r$  为召回率,  $N$  代表此次实验所用到的测试集数即为 3603; 多类别平均精度  $mAP$  是将所有类的  $AP$  值进行平均, 公式为:

$$mAP = \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (3)$$

在式 (4) 中,  $Q$  代表实验数据集的类别数, 即为 6 类.

图 7 为表 4 中各个算法的  $AP$  与  $mAP$  直观图.

为了更直观综合评价算法好坏, 将精确率、召回率和  $F1$  值这 3 个指标在不同算法上的对比图如图 8 所示. 从图 7 中的召回率和准确度可知, 本文算法相对于文献 [13]、文献 [1] 找到正确的 6 种车型图片要多, 并且准确率也相当高, 比传统的 YOLOv3 高达 3.8%, 说明本文模型综合利用密集卷积网络提取到更多的车型基本结构特征; 另外, 本文算法平均检测速度达到了 28 fps, 在与 DenseNet 网络集成后也保证了实时检测效果; 总体来看, 本文算法在不同类别的车型情况下, 各类的  $AP$  相对于其他文献 [13]、文献 [1] 两种算法占很大的优势, 且多类平均值高出 6.45%. 因此, 这些数据很好的验证了本文模型优越性和使用价值.

## 3 结论与展望

文中设计的 Dense-YOLOv3 模型通过将密集卷积神经网络作为主要车型特征的提取, 结合 YOLOv3 中的三尺度检测框进行局部信息的交互来实现目标车的

表3 不同算法的识别过程

|              | Microbus与SUV<br>直方图重合度: 0.604 dHash:16 | Microbus与Sedan<br>直方图重合度: 0.798 dHash:12 | Sedan与SUV<br>直方图重合度: 0.602 dHash:14 |
|--------------|--|--|-------------------------------------|
| 示例图<br>(相似度) |  |  |                                     |
| 文献[13]       |  |  |                                     |
| 文献[1]        |  |  |                                     |
| 本文算法         |  |  |                                     |
|              | 33.99                                  | 20.31                                    | 166.95                              |
| 示例图<br>(亮度)  |  |  |                                     |
| 文献[13]       |  |  |                                     |
| 文献[1]        |  |  |                                     |
| 本文算法         |  |  |                                     |

表 4 算法对比

| 算法     | AP(%) |          |         |       |       |       | mAP(%) | Recall(%) | Precision(%) | FPS(fps) |
|--------|-------|----------|---------|-------|-------|-------|--------|-----------|--------------|----------|
|        | Bus   | Microbus | Minivan | Sedan | SUV   | Truck |        |           |              |          |
| 文献[13] | 98.61 | 81.93    | 91.58   | 96.42 | 79.61 | 92.59 | 90.12  | 76.72     | 93.41        | 35       |
| 文献[1]  | 97.24 | 78.42    | 64.31   | 94.71 | 89.34 | 97.47 | 86.91  | 76.83     | 90.02        | 29       |
| 本文算法   | 98.88 | 95.77    | 95.72   | 95.80 | 95.51 | 97.75 | 96.57  | 93.30     | 97.21        | 28       |

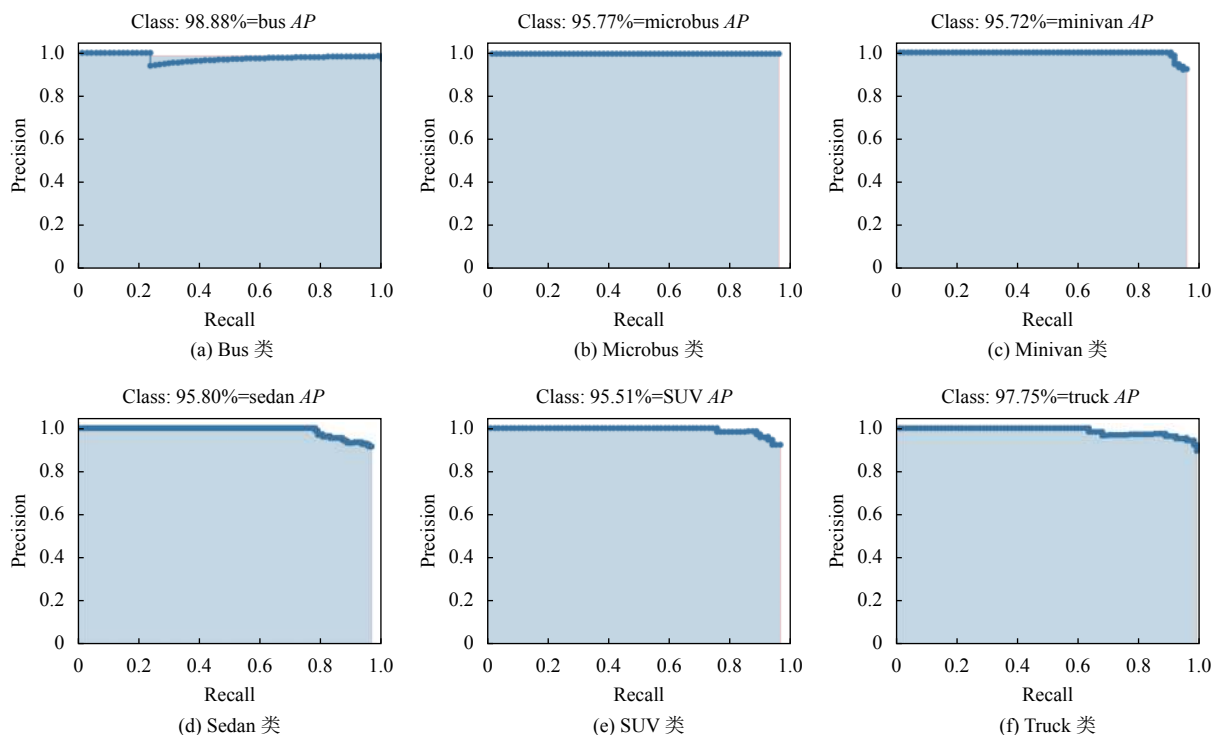


图 6 Precision-Recall 曲线

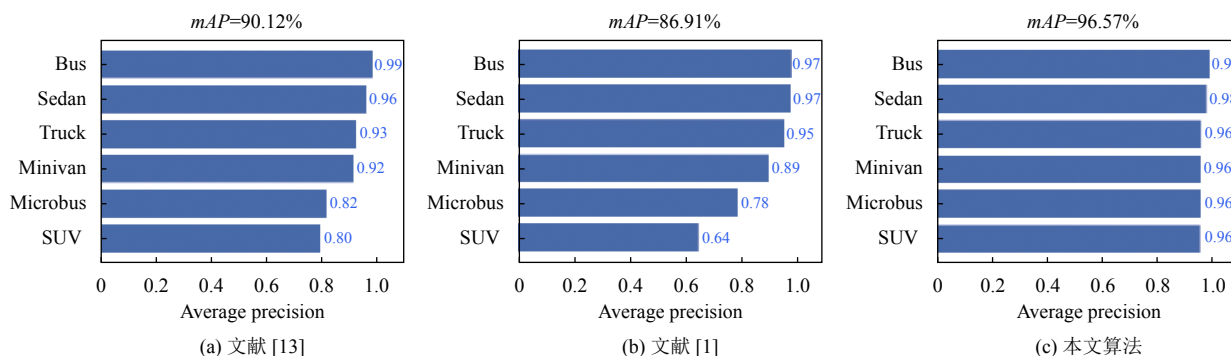


图 7 各算法 AP 与 mAP 图

检测,综合利用密集卷积网络的优势即在车辆存在异常环境下将提取到的车型特征信息进行重复利用,以此来提高了车型准确率,并且此模型在 BIT-Vehicle 车型数据集进行了实验验证.先采用 3 种不同的算法对拍摄距离较远和光线不同图片的检测过程进行定性分析;然后在车型数据集上利用目标检测指标性能参数

进行定量分析对比.通过定性和定量的实验分析,本文模型有较强的学习、抗过拟合能力,在实际应用中有很好的检测效果并且有一定的使用价值.但是,实验中的车型数据集有一些不足:车型数据集的种类较少、场景较为单一等的缺点.因此,在后续工作中将寻找更多样的、不同场景下车型数据集进行实验研究.



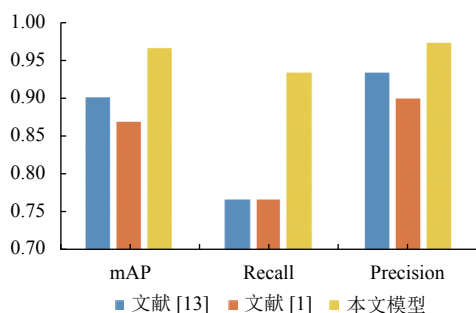


图8 各算法的准确率、召回率、F1值对比

## 参考文献

- 张富凯, 杨峰, 李策. 基于改进 YOLOv3 的快速车辆检测方法. 计算机工程与应用, 2019, 55(2): 12–20. [doi: 10.3778/j.issn.1002-8331.1810-0333]
- 石磊, 王亚敏, 曹仰杰, 等. 基于深度卷积神经网络的车型识别. 计算机科学, 2018, 45(5): 280–284.
- 何霞, 汤一平, 袁公萍, 等. 基于级联多任务深度学习的卡口识别引擎研究. 计算机科学, 2019, 46(1): 303–308. [doi: 10.11896/j.issn.1002-137X.2019.01.047]
- 王思元, 王俊杰. 基于改进 YOLOv3 算法的高密度人群目标实时检测方法研究. 安全与环境工程, 2019, 26(5): 194–200.
- Girshick R, Donahue J, Darrell T, *et al.* Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014. 580–587.
- He KM, Zhang XY, Ren SQ, *et al.* Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 37(9): 1904–1916.
- Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448.
- Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149.
- He KM, Gkioxari G, Dollár P, *et al.* Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386–397.
- Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 6517–6525.
- Liu W, Anguelov D, Erhan D, *et al.* SSD: Single shot MultiBox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam, the Netherlands. 2016. 21–37.
- Redmon J, Divvala S, Girshick R, *et al.* You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788.
- Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv: 1804.02767, 2018.
- 徐诚极, 王晓峰, 杨亚东. Attention-YOLO: 引入注意力机制的 YOLO 检测算法. 计算机工程与应用, 2019, 55(6): 13–23. [doi: 10.3778/j.issn.1002-8331.1812-0010]
- 王琳, 卫晨, 李伟山, 等. 结合金字塔池化模块的 YOLOv2 的井下行人检测. 计算机工程与应用, 2019, 55(3): 133–139. [doi: 10.3778/j.issn.1002-8331.1710-0236]
- He KM, Zhang XY, Ren SQ, *et al.* Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778.
- Huang G, Liu Z, van der Maaten L, *et al.* Densely connected convolutional networks. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 2261–2269.