**人工智能综合设计课程设计报告**

|  |  |
| --- | --- |
| **作业名称：** | **基于YOLOv3/4的混合模态行人检测算法** |
| **学 院：** | 计算机学院 |
| **班 级：** | 10052001 |
| **学 号：** | 2020300400 |
| **姓 名：** | 王涛 |
| **团队组成：** | 王涛 |

**西北工业大学**

**2023年7月25日**

**1、任务说明（描述所要完成的任务和问题）**

|  |
| --- |
| 使用Kaist行人检测数据集做行人目标检测  要求：  可以使用原始的标注或者增强后的标注  必须使用RGB和Thermal双模态数据  考虑到训练所需资源，可以选择部分Train和对应的Test做评测  完成可执行的代码项目，可参考网络开源代码和项目  榜单中的评测精度仅供参考，作业成绩会参考精度  最后需要提交源代码和实验报告 |

**2、开发工具（列出所使用的机器学习平台和相关库）**

|  |
| --- |
| Python 3.9.6  Requirements:  numpy  torch  tensorboard  scipy  opencv\_python  pycocotools  matplotlib  tqdm |

**3、设计内容（整体软硬件设计框架）**

|  |
| --- |
| **一种基于YOLOv3/4的双流混合模态道路行人检测方法：将可见光和红外光模态图像分别送入双流并行主干提取网络并通过特征共享学习网络进行混合模态特征融合，最后进入FPN/PANet进行特征聚合加强，最后预测层进行目标检测输出边界框回归参数、置信度和类别分数。同时考虑到网络的实时性和轻量化.**  **kaist_dyolov3_fshare.drawio**  **主要工作包括：**  **- 在YOLOv4 Mosaic数据增广方法的基础上引入雪花变换、CLAHE以及伪目标嵌入等方法，提高网络泛化能力。**  **- 使用双流并行的CSPDarknet53网络从可见光和红外光图像中提取出各自的特征图。**  **- 设计了基于特征共享网络FSNet的混合模态特征融合方法，通过相互指导的方式实现红外光图像特征与可见光图像特征图的有效融合。**  **kaist_dyolov4.drawio**  **二、数据集说明**  **本文采用的是KAIST数据集，该数据集是于2015年由Hwang等人所构建的多光谱检测数据集，主要目的就是解决夜间环境下行人检测数据缺乏的问题。数据集本身分为12个子集，其中set00~set05为训练数据（set00~set02为白天场景，set03~set05为夜间场景），set06~set11为测试数据（set06~set08为白天场景，set09~set11为夜间场景），图像分辨率大小640x512，总共包含95328张图片，每张图片都包含RGB彩色图像和红外光图像两个版本。数据集分别在白天和夜间捕获了包括校园、街道以及乡下等多个常规交通场景，并含有103108个密集标注，其中较好区分的行人个体被标注为person，不太好分辨的多个个体则被标注为people，骑行的人则被标注为cyclist。**  **但考虑到标注的people包含多个行人目标，对于网络模型的学习可能会产生不良的影响；且骑行者cyclist本身也是分辨良好的人体，因此在实际的数据清洗中会忽略people标注，并将cyclist替换成person。除此之外，数据集中的图片由于都是从视频上截取下来，编号相近的图片所记录的画面几乎含有相同的信息，因此在数据清洗的时候会以一定步距抽样方式对有效使用的数据进行筛选，防止模型重复学习。**  **实际实验时，采用上面指定的部分数据集作为训练数据集和验证集，两者随机按照3：1分配，最后剩下的数据集作为测试集（由白天数据集和夜间数据集组成）。所有的下面实验得到的性能数据都是在测试数据集上得到的。** |

**4、算法5、程序实现（核心程序片段）**

|  |
| --- |
| **double-yolov4-kaist.drawio**  **论文工作.drawio**  **Conv2d层+BN层+激活层组成的模块，可用于实现CBL、CBM等结构**    **残差模块Residual Block x n，默认使用Mish激活函数**    **Inception结构，共有4条分支，分别为1x1卷积、3x3卷积、5x5卷积和一个池化分支**    **通道注意力机制**    **SE-Inception Fusion混合模态特征融合模块**    **深度可分离卷积可以用来替代卷积核大小为3x3的普通卷积，减少网络参数量和计算量**    **混合深度卷积（MixConv），即在一个卷积中自然的混合多个不同大小的卷积核。用这种简单的嵌入式替换普通的深度卷积，**  **MixConv可以提高了MobileNets在ImageNet的分类任务和COCO的目标检测任务的准确率和效率。**    **雪花变换**    使用前需要下载上述Kaist数据集到Kaist目录下，解压并通过trans\_kaist2voc.py转换成VOC格式的Kaist数据集（之所以不直接转换到YOLO格式是因为前期的一些工作是在VOC格式的数据集上做的），然后使用trans\_kaistvoc2yolo.py将VOC格式的数据集转换成YOLO格式的数据集，接着使用dataset\_calculate.py从转换得到的数据集中获取相关数据写入到data目录下的文件中。然后从U版的YOLOv3、WongKinYiu版的YOLOv4中下载对应的yolov3.pt和yolov4.weights权重文件，方便通过迁移学习的方式加载到自己的网络中进行训练。 |

**6、调试过程（算法训练测试过程描述，以及中间结果图表）**

|  |
| --- |
| 训练Training：  # 以训练kaist-visible-yolov3为例  python train.py --epochs 300 --batch-size 16 --cfg config/kaist\_yolov3.cfg --weights weights/pretrained\_yolov3.pt --name kaist\_yolov3 --freeze-layers -1 --anchor-cluster  推理Detecting：  # 以使用kaist-visible-yolov3推理为例  python detect.py --model-name Visible-YOLov3 --cfg config/kaist\_yolov3.cfg --weights weights/kaist\_yolov3\_best.pt --src imgs/ori/I0020\_lwir.jpg --save imgs/res# 使用时只要给出一张可见光或者红外光图像的路径它就会自动的去找另一张图像的  使用测试集获取模型性能指标Evaluating：  # 以获取kaist-visible-yolov3在夜间测试集上的性能为例，并将结果保存在rec-prec.fppi-mr.npy文件中  python evaluate.py --cfg config/kaist\_yolov3.cfg --weights weights/kaist\_yolov3\_best.pt --batch-size 64 --test-  参数调整： |

**7、结果分析（运行测试结果图表和分析，包括失败例子的分析）**

|  |
| --- |
| 表格分析：  基于yolov3的模型表现效果尚可，其中未优化的普通yolov3模型的全天候测试集准确率既达到了82%左右，在经过雪花变换、CLAHE以及伪目标嵌入等方法优化后准确率有了显著的提升，达到了89%左右。使用基于特征共享网络FSNet的混合模态特征融合方法后，比较明显的是检测帧数的提升。    上述实验中较好算法模型得到的P-R曲线和FPPI-MR曲线如下图所示：  yolov3.pr-fm-4  在yolov4模型的工作中，SL、CSE和Fshare等优化方式也具有显著的准确率提升的效果。但更令人值得关注的是进行尝试使用MobileNet来轻量化目标检测网络的参数量和计算耗时性的优化后，得出的效果显著：在使用MobileNetv2-Fshare-Global-CSE3后，准确率上升7%左右，而检测帧数也未降低到一个较低的水平，反而要比yolov4高10帧左右。    yolov4.pr-fm-8  所有模型性能对比图：  yolov3-4.pr-fm-12  I00304  I00972  I01039  I01136  I01949  I03307  此检测图像（下图）是一个典型的因遮挡和重叠在图像上只有部分人体而未检测出的典例，在居中图像上，一位因同行人的前后遮挡只有右半个身体露出而未被检测出，一位因车辆遮挡没有全部的热成像图而未被检测出。这两个例子体现了yolov3、v4模型系列对部分目标的预测性能不足的缺点。  I01566 |

**8、程序开发总结（简要叙述编写本作业的收获与思考）**

|  |
| --- |
| 本次作业历时近两个月，完成并实现了基于yolov3v4的多模态行人检测任务，深入学习了pytorch与paddlepaddle框架，加深了对目标检测任务的理解，极大增强了我对目标检测算法的理解。在工作过程中我查阅了许多资料与相关文献，对目标检测领域的发展与前沿技术有更深刻的理解。  （上述实验得到的网络权重下载地址：  https://pan.baidu.com/s/1byPQq2gXpL0ZLnuBIer\_Tg?pwd=edut 提取码：edut）  **##** 五、参考资料  - WongKinYiu版YOLOv4 Pytorch实现：https://github.com/WongKinYiu/PyTorch\_YOLOv4  - Ultralytics版YOLOv3 Pytorch实现：https://github.com/ultralytics/yolov3  - Kaist数据集所属项目：https://github.com/SoonminHwang/rgbt-ped-detection  - 多类别平均准确度计算：https://github.com/Cartucho/mAP  - 《动手学深度学习》：https://zh.d2l.ai/ |