《基于深度学习的目标检测实验报告》

目录

（一）目标检测经典算法介绍 2

（二）实验内容、目标和动机 4

（三）yolov5原理介绍 4

（四）yolov5的基本架构 5

（五）yolov5核心架构的具体实现 6

（六）网络训练和推理过程及说明 8

（七）实验结果比对和分析 9

（八）总结和讨论 11

（九）参考文献 11

**一、目标检测经典算法介绍**

**（一）R-CNN系列**

R-CNN(全称Regions with CNN features) 是R-CNN系列的第一代算法，没有过多的使用“深度学习”思想，而是将“深度学习”和传统的“计算机视觉”的知识相结合。[1]

**1、R-CNN**

（1）流程

①预训练模型。选择一个预训练神经网络（如AlexNet、VGG）。

②重新训练全连接层。使用需要检测的目标重新训练最后全连接层。

③提取 proposals并计算CNN 特征。利用选择性搜索算法提取所有proposals（大约2000幅images），调整它们成固定大小，以满足 CNN输入要求（因为全连接层的限制），然后将feature map 保存到本地磁盘。

④训练SVM。利用feature map 训练SVM来对目标和背景进行分类。

⑤边界框回归。训练将输出一些校正因子的线性回归分类器。

（2）效果

R-CNN在VOC 2007测试集上mAP达到58.5%，打败当时所有的目标检测算法。

（3）缺点

①重复计算，每个region proposal，都需要经过一个AlexNet特征提取，为所有的RoI（region of interest）提取特征大约花费47秒，占用空间。

②selective search方法生成region proposal，对一帧图像，需要花费2秒。

③三个模块（提取、分类、回归）是分别训练的，并且在训练时候，对于存储空间消耗较大。[2]

**2、SPPNet**

（1）简介

SPPNet[3]提出了一种空间金字塔池化层(Spatial Pyramid Pooling Layer, SPP)。它的主要思路是对于一副图像分成若干尺度的图像块(比如一副图像分成1份，4份，8份等)，然后对每一块提取的特征融合在一起，从而兼顾多个尺度的特征。

（2）性能

相比于RCNN算法，SPPNet在Pascal-07数据集上不牺牲检测精度(VOC-07, mAP=59.2%)的情况下，推理速度提高了20多倍。

（3）不足

和RCNN一样，SPP也需要训练CNN提取特征，然后训练SVM分类这些特征，这需要巨大的存储空间，并且多阶段训练的流程也很繁杂。除此之外，SPPNet只对全连接层进行微调，而忽略了网络其它层的参数。

（二）YOLO系列

**1、YOLOv1**

（1）简介

YOLO [4]的核心思想就是把目标检测转变成一个回归问题，利用整张图作为网络的输入，仅仅经过一个神经网络，得到bounding box的位置及其所属的类别。

（2）优点

①YOLO检测速度非常快。标准版本的YOLO可以每秒处理 45 张图像；YOLO的极速版本每秒可以处理150帧图像。这就意味着 YOLO 可以以小于 25 毫秒延迟，实时地处理视频。对于欠实时系统，在准确率保证的情况下，YOLO速度快于其他方法。

②YOLO 实时检测的平均精度是其他实时监测系统的两倍。[5]

③迁移能力强，能运用到其他的新的领域（比如艺术品目标检测）。

（3）局限：

①YOLO对相互靠近的物体，以及很小的群体检测效果不好，这是因为一个网格只预测了2个框，并且都只属于同一类。

②由于损失函数的问题，定位误差是影响检测效果的主要原因，尤其是大小物体的处理上，还有待加强。（因为对于小的bounding boxes，small error影响更大）

③YOLO对不常见的角度的目标泛化性能偏弱。[6]

**2、YOLOv2**

相比于YOLOv1是利用全连接层直接预测Bounding Box的坐标，YOLOv2借鉴了Faster R-CNN的思想，引入Anchor机制。利用K-means[7]聚类的方法在训练集中聚类计算出更好的Anchor模板，大大提高了算法的召回率。同时结合图像细粒度特征，将浅层特征与深层特征相连，有助于对小尺寸目标的检测。

**3、YOLOv3**

2018年，作者 Redmon 又在 YOLOv2 的基础上做了一些改进。[8]特征提取部分采用darknet-53[9]网络结构代替原来的darknet-19，利用特征金字塔网络结构实现了多尺度检测，分类方法使用逻辑回归代替了softmax，在兼顾实时性的同时保证了目标检测的准确性。

在YOLOv3中，作者不仅提供了darknet-53，还提供了轻量级的tiny-darknet[10]。如果你想检测精度与速度兼具，可以选择darknet-53作为backbone；如果你希望达到更快的检测速度，精度方面可以妥协，那么tiny-darknet是你很好的选择。

**二、实验内容、目标和动机**

**1.实验内容**

(1) 下载COCO数据集

训练集（Train）：118,287张图像，包含80种类别的物体。

验证集（Validation）：5,000张图像。

测试集（Test）：40,670张图像。

(2) 构建yolov5模型并进行训练，将COCO数据集中图像中的物体进行检测。

模型输入：用COCO数据集中的训练图像作为模型训练输入。

模型输出：检测图像中的物体，并标注物体的类别和位置。

**2.实验目标和动机**

（1）评估和验证yolov5算法在不同场景下的检测性能，包括检测速度、准确率和鲁棒性。

（2）通过实验，期望能够深入理解yolov5的工作原理和应用效果，并为进一步优化和改进目标检测算法提供数据支持和技术参考。

（3）将实验结果和几种经典的目标检测算法进行对比，综合评价yolov5目标检测算法的优缺点。

**三、yolov5原理介绍**

yolov5[11]是单阶段目标检测算法的一种，采用端到端的方式直接在图像上进行物体检测。其核心原理是将输入图像划分为S×S的网格，每个网格预测一个或多个边界框及其所属的类别概率。yolov5通过卷积神经网络提取图像特征，并在预测头中输出目标的边界框坐标和类别信息。相比于多阶段的检测方法，yolov5具备更高的检测速度和较好的准确率，同时通过改进的损失函数、数据增强方法和模型架构，进一步提升了检测性能和鲁棒性，适用于实时性要求较高的应用场景。

**四、yolov5的基本架构**

yolov5的基本架构主要分为四个部分：输入、骨干网络、颈部和检测头。

**1.输入（Input）**

包括数据预处理和数据增强。输入图像在进入网络前进行归一化、尺寸调整和数据增强，以提高模型的泛化能力和鲁棒性。

**2.骨干网络（Backbone）**

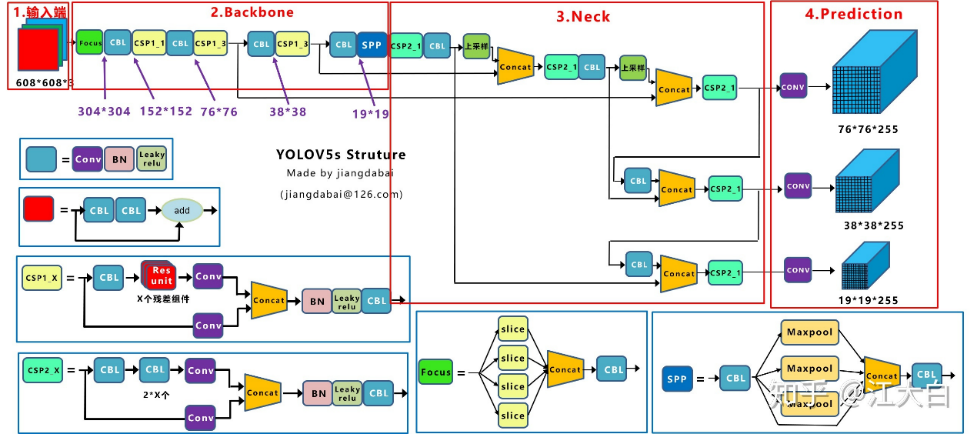
主要负责特征提取。YOLOv5使用CSPDarknet作为骨干网络，通过一系列卷积层、残差连接和CSP（Cross Stage Partial）模块来提取不同尺度的特征。CSP模块有助于减少计算量并提高模型效率。

**3.颈部（Neck）**

负责特征融合和进一步的特征提取。yolov5采用FPN（Feature Pyramid Network）[12]和PANet（Path Aggregation Network）[13]结构，结合多层次的特征图，提高了检测小目标的能力。FPN用于自顶向下的特征融合，而PANet用于自底向上的特征整合。

**4.检测头（Head）**

负责输出目标的边界框和类别信息。检测头在多个尺度的特征图上进行预测，以便检测不同尺寸的目标。每个尺度的特征图输出目标的边界框坐标、置信度和类别概率。

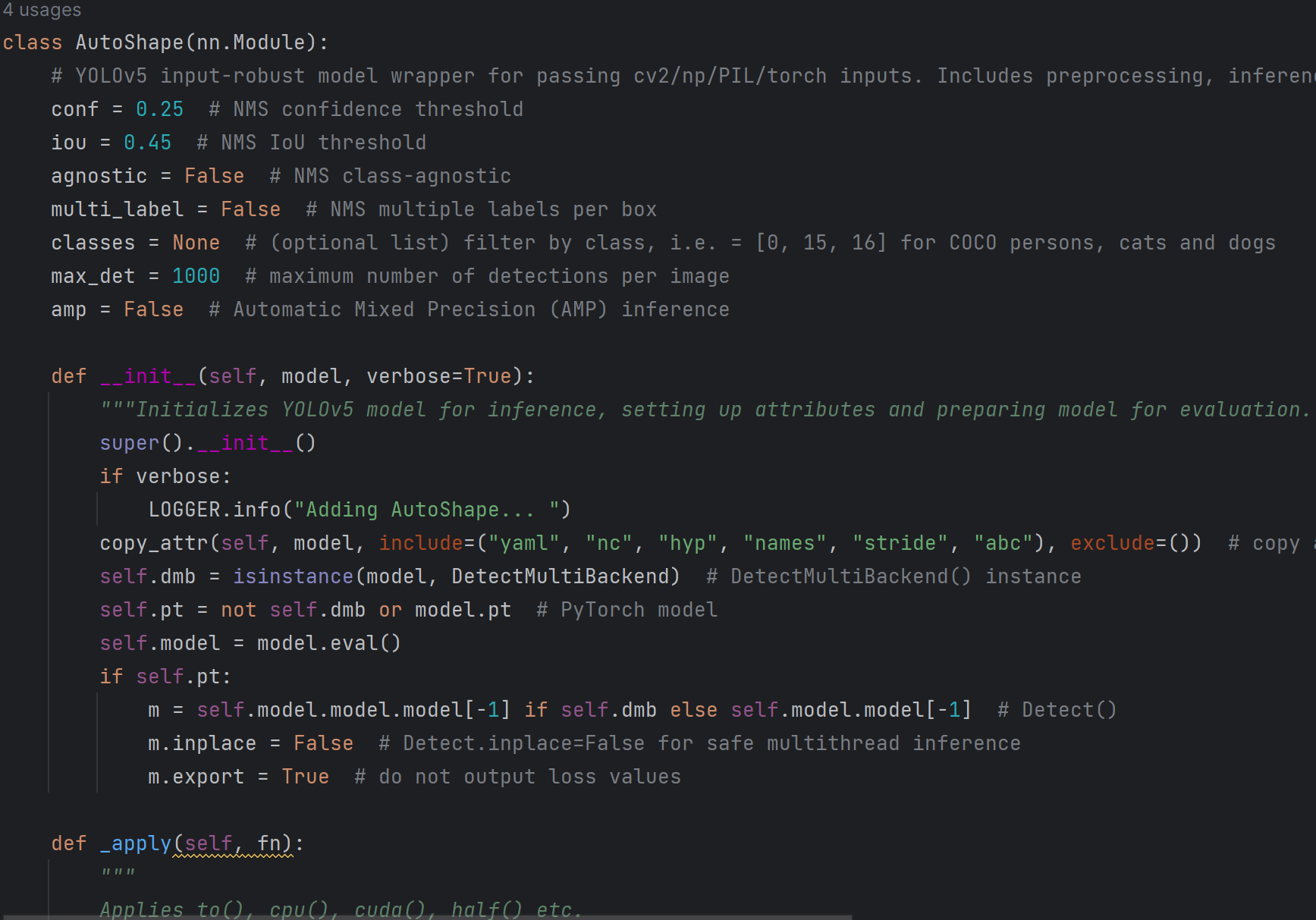


**五、yolov5核心架构的具体实现**

**1.Input（输入）**

（1）AutoShape模块

功能：用于处理输入数据并进行预处理、推断和非极大值抑制（NMS）操作。



**2. Backbone（骨干网络）**

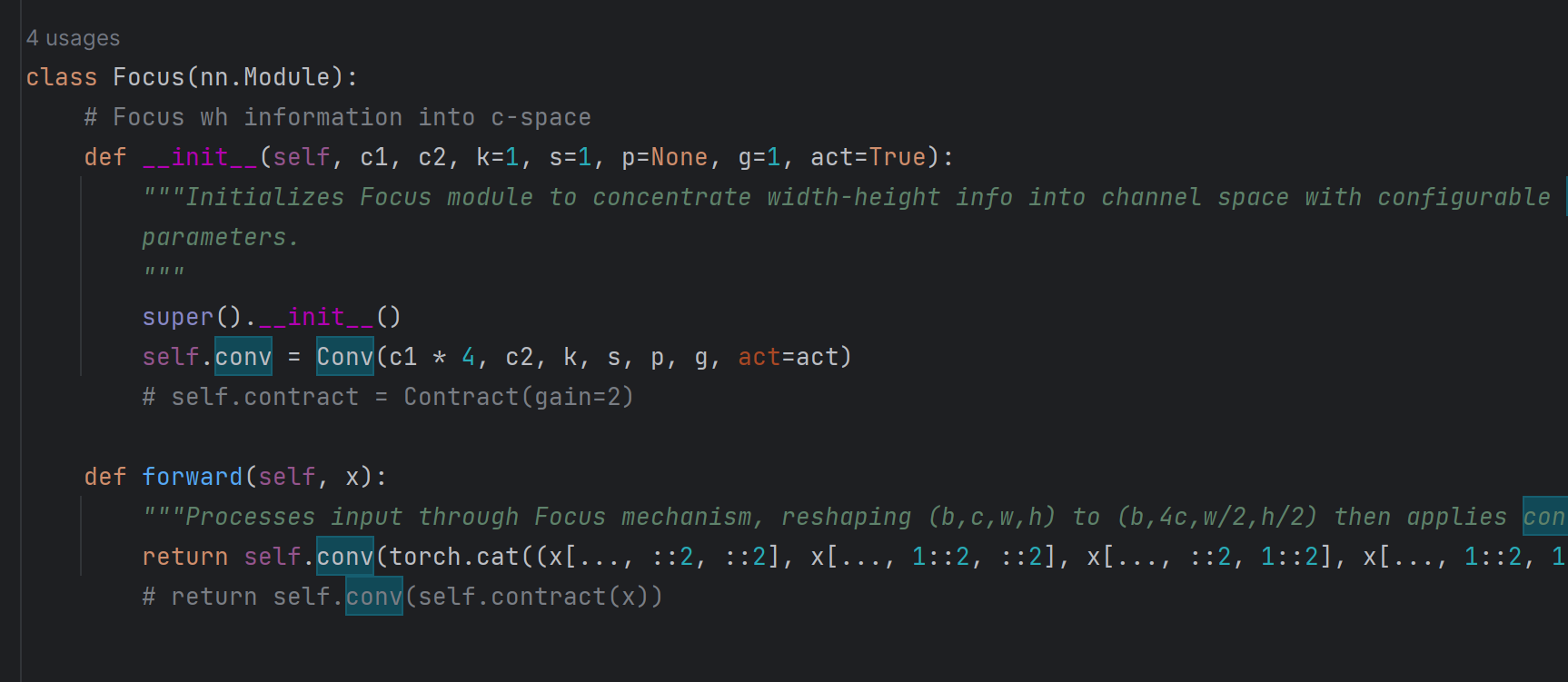
（1）包含三个卷积层的bottleneckCSP模块

功能：增强网络的表示能力，并通过跨阶段连接促进信息流动，提高网络的性能和效率。



（2）Focus模块

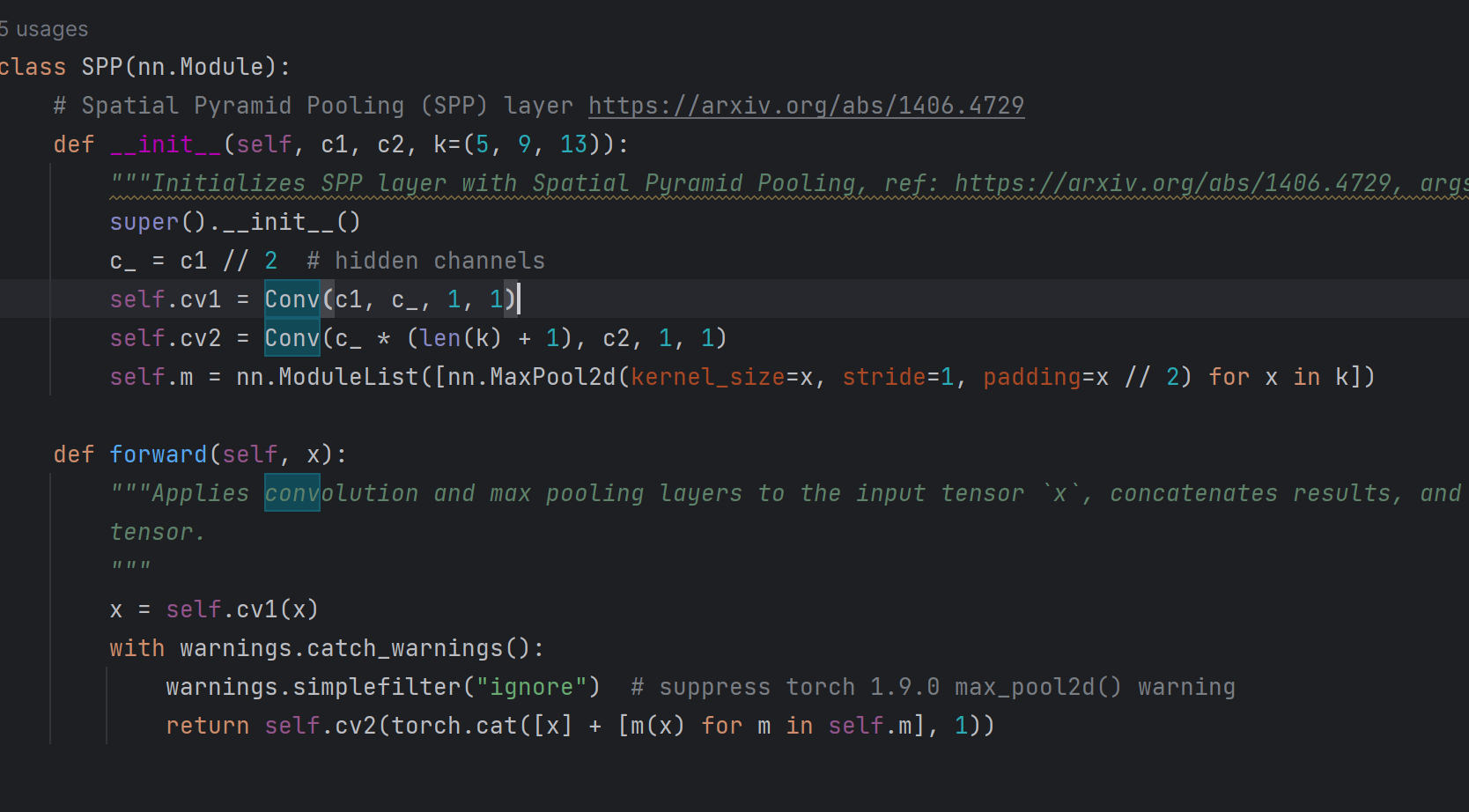
功能：用于集中（focus）宽度和高度信息到通道空间，旨在增强网络对特征的表达能力和感知能力。



**3.Neck（颈部）**

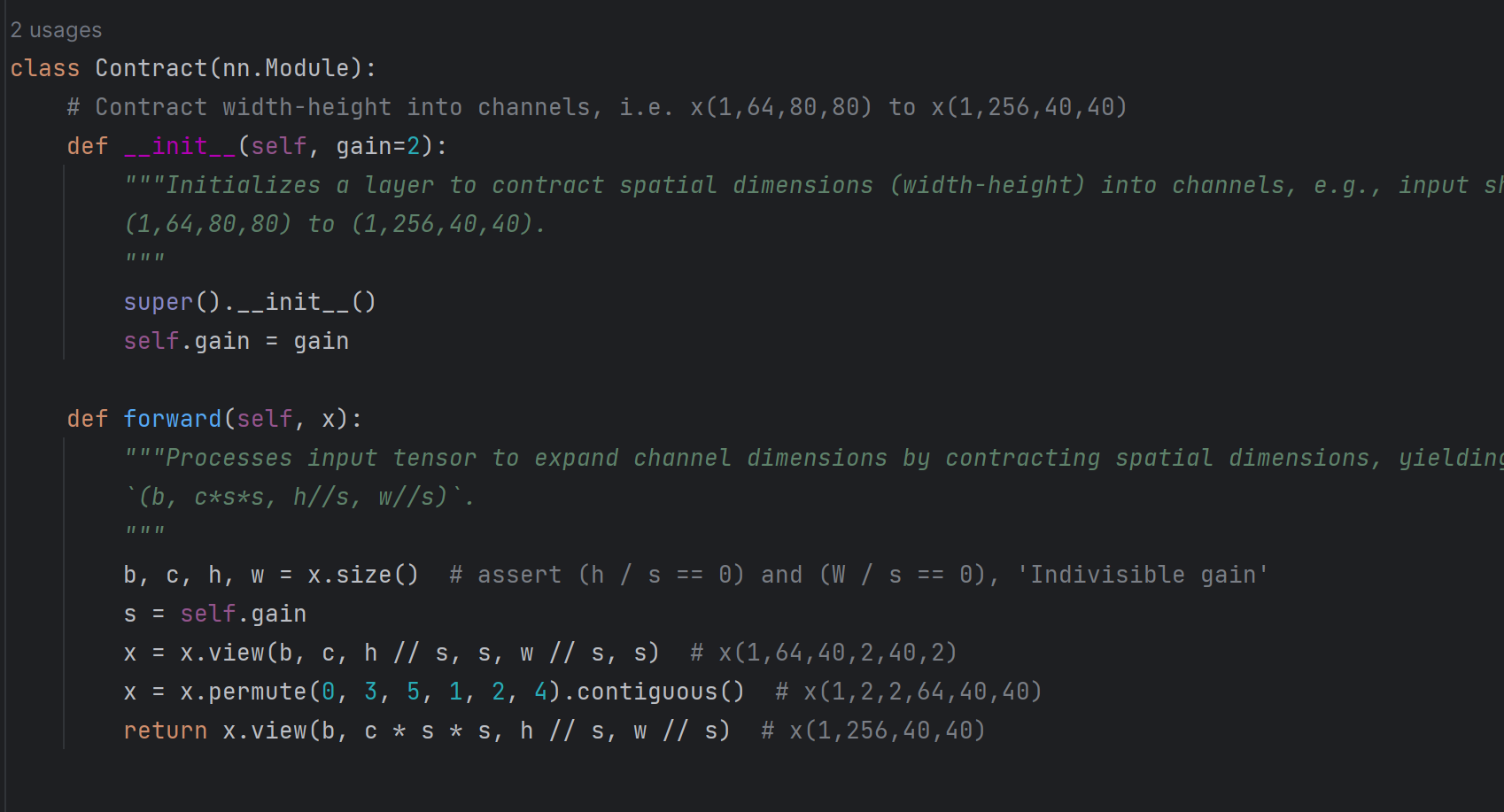
（1）SPP模块

功能：用于增强神经网络对输入特征的空间信息的感知能力



（2）Contract模块

功能：用于实现特征金字塔网络或者其变体的功能



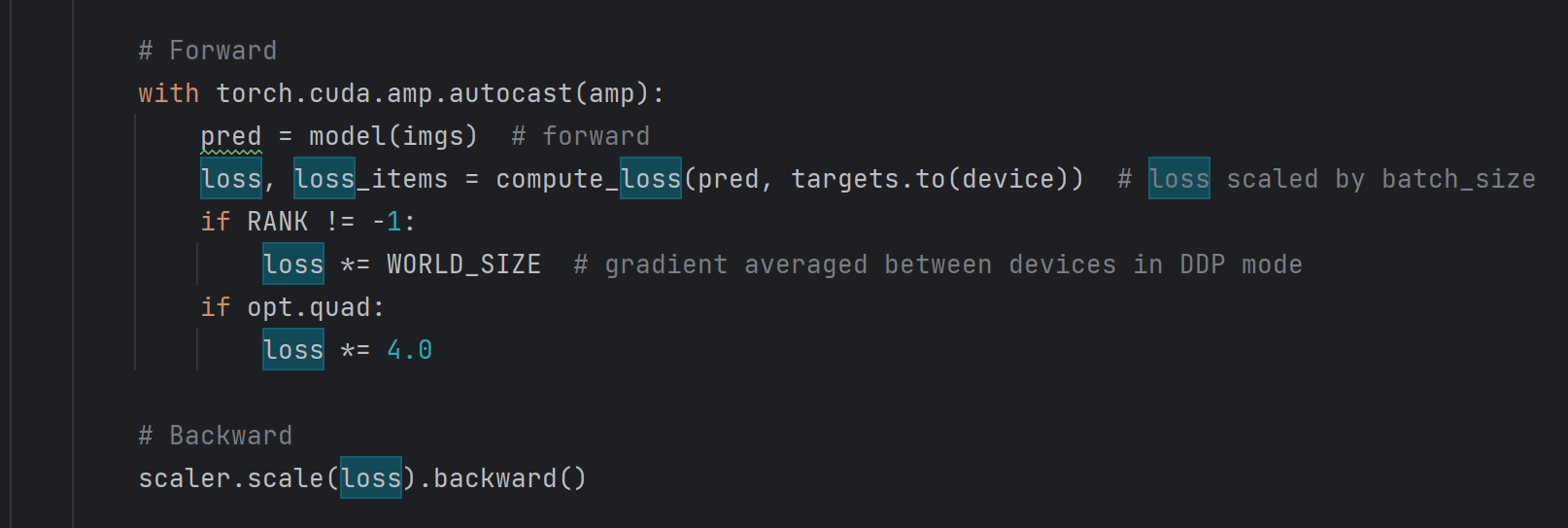
**4. Head（检测头）**

（1）Classify模块

功能：用于预测物体的类别



（2）损失计算



**六、网络训练和推理过程及说明**

**1.网络训练过程**

1. 初始化训练参数和辅助工具，如计算损失、优化器、学习率调度器等。
2. 设置训练时的一些参数，如warmup的迭代次数、损失函数权重、梯度累积等。
3. 进入训练循环，逐个epoch进行训练。每个epoch包括以下步骤：
   1. 设置模型为训练模式，并根据需要进行图片尺寸、权重更新等操作。
   2. 遍历训练数据加载器中的每个批次，进行前向传播、计算损失、反向传播、优化器更新等操作。
   3. 在训练过程中记录损失和性能指标，并在适当的时候进行模型保存和评估。
   4. 在每个epoch结束时进行验证，计算模型的性能指标如mAP。
   5. 在DDP模式下进行早停检查，并在满足条件时结束训练循环。

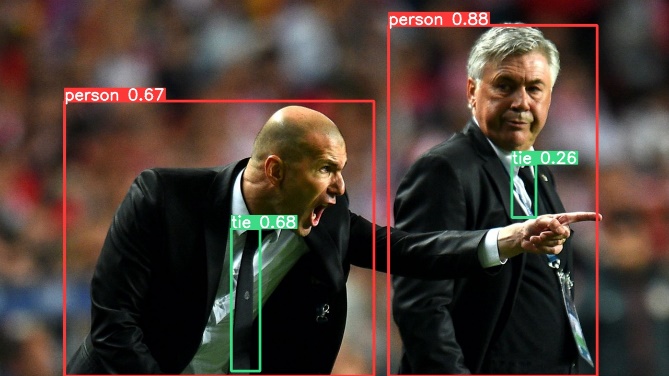
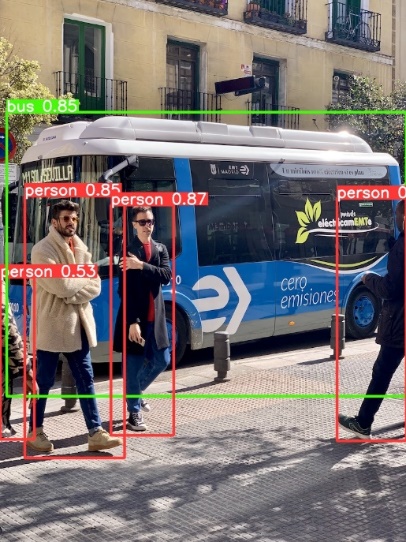
**2.网络推理过程**

1. 配置模型为评估模式，加载数据，并创建数据加载器。
2. 对数据进行推理，计算推理结果和相关指标，如准确率、召回率、mAP等。
3. 保存推理结果到文件，如文本文件、JSON文件等。
4. 可以使用回调函数来扩展功能，如在推理开始前、每个批次开始、每个批次结束等时刻执行特定操作。
5. 进行推理并计算损失（如果需要计算损失）。
6. 对预测结果进行非极大值抑制（NMS）处理。
7. 计算指标，如精确率（P）、召回率（R）、mAP等。
8. 可选地保存推理结果到文本文件或JSON文件，并绘制相关图表。
9. 如果需要，对结果进行打印和保存，并返回相关指标和评估结果。

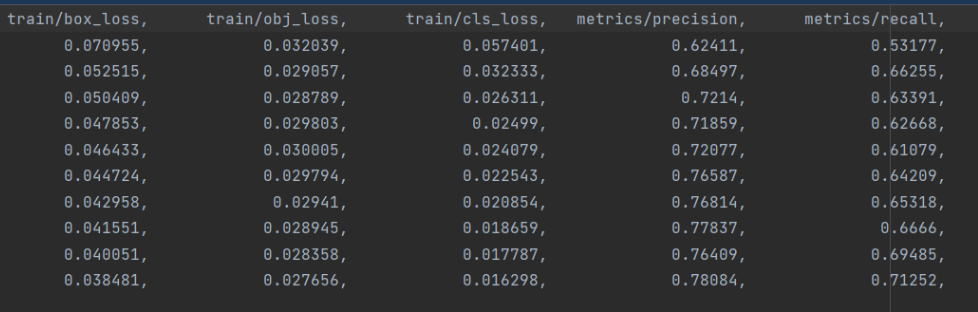
**七、实验结果比对和分析**

**1.实验结果分析**

（1）模型在不同类别上的表现存在差异，有些类别（如人、车）表现较好，有些类别（如领带）表现相对较差。

****

（2）整体的精确率、召回率较高，说明模型在目标检测任务中具有较好的性能。

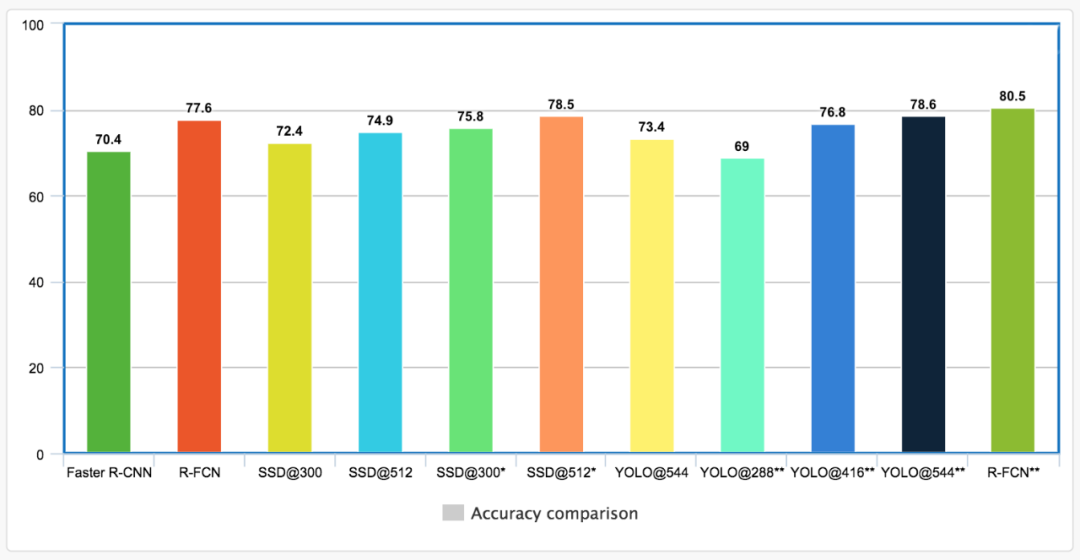


（3）推理速度较快，适合实时或近实时的应用场景。

**2.实验对比分析**

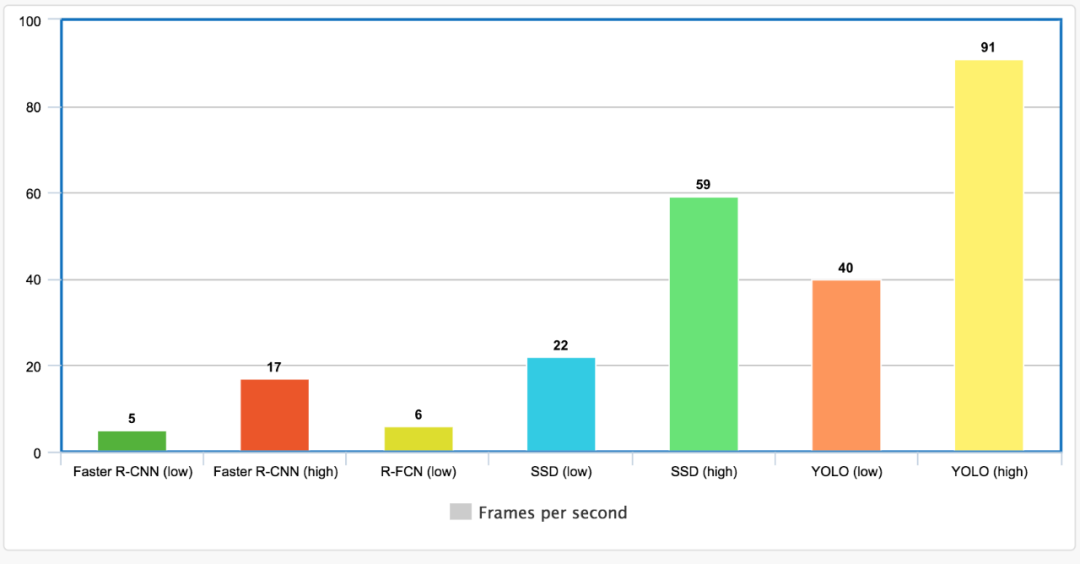
**（1）准确率比较**

将yolov5与Fast R-CNN, R-FCN等目标检测算法进行比较，发现R-FCN准确率最高。通过查阅资料发现，R-FCN模型使用区域候选网络生成一组候选区域，然后在这些候选区域上进行分类和回归，而且RPN还可以生成较高质量的候选框，从而提高目标检测的准确性。



**（2）推理速度比较**

从图中可以明显看到，yolov5算法在目标检测任务的速度性能最好。这是因为YOLOv5是一个单阶段检测器，能够直接在输入图像上进行目标检测，大大简化了计算过程。Yolov5还使用CSPDarknet53作为其主干网络，能够在保证准确率的同时显著减少计算量。此外，YOLOv5提供了预训练权重，使得在新数据集上的微调更为高效，yolov5综合以上优势，从而大大提高了检测速度。



**八、总结和讨论**

**1.总结**

本次实验对YOLOv5模型进行了全面的评估，主要针对其在目标检测任务中的性能表现进行了测试。结果表明YOLOv5在检测速度和准确度方面均表现良好，尤其是在实时应用场景中，其速度优势尤为明显。

**2.讨论**

（1）虽然YOLOv5的轻量级结构带来了速度优势，但在处理复杂背景或密集目标时，其检测精度可能不如一些更复杂的模型（如R-FCN）。因此未来的研究或许可以探讨在保持轻量级特性的同时，进一步提高模型在复杂场景中的检测能力。

（2）本次实验中，数据增强技术对模型性能的提升起到了关键作用。除了现有的增强方法，还可以尝试引入新的数据增强策略，如Regmixup，它可以作为损失函数的正则化器来优化损失函数，提高检测OOD（分布外）样本的精确度，从而提高模型性能。

（3）yolov5在目标检测方面表现出较好的效果，而如今已经推出了yolov8系列，或许可以探索它在中国前沿领域（如生物工程与基因编辑、空间探索与航天技术和自动驾驶与智能交通等）中的应用效果，助力中国前沿领域的发展。

**九、参考文献**

[1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[2]He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In European conference on computer vision (pp. 346-361). Springer, Cham.

[3] 王鹏飞, 刘成名, 张鹏, 等. (2014). 基于深度学习的目标检测研究综述. *计算机应用研究*, 31(10), 2931-2935.

[4] 李建华, 杨旭东, 韩宗沅, 等. (2015). 深度学习在目标检测中的应用与研究进展. *计算机科学*, 42(4), 20-26.

[5]Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[6] 张宇杰, 刘瑞祺, 赵琦, 等. (2016). 基于YOLO算法的实时目标检测. *计算机工程与应用*, 52(20), 137-141.

[7]Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[8]Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[9]Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[10]Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[11]Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[12]刘辉, 李鑫, 李成. (2017). YOLOv2改进算法及其在交通场景中的应用. *电子与信息学报*, 39(9), 2346-2351.

[13]Huang, Z., & Liang, Y. (2021). YOLOv5: Model enhancement and real-time detection of human action. arXiv preprint arXiv:2104.08378.