

# 目标检测实验

---

本次共准备了两个实验，大家选择其一完成即可。本次提交的实验报告请以 **实验x\_实验名\_姓名** 进行命名（实验一or实验二）。

## 目标检测实验

### 实验一：YOLOv3

实验介绍

yolo模型介绍

运行方法

实验要求

### 实验二：faster-rcnn

实验描述与要求

RCNN模型介绍

faster-rcnn模型介绍

模型对比

## 实验一：YOLOv3

### 实验介绍

实验来源：<https://github.com/dilipnair/YOLOv3-Object-Detection-with-OpenCV-master>

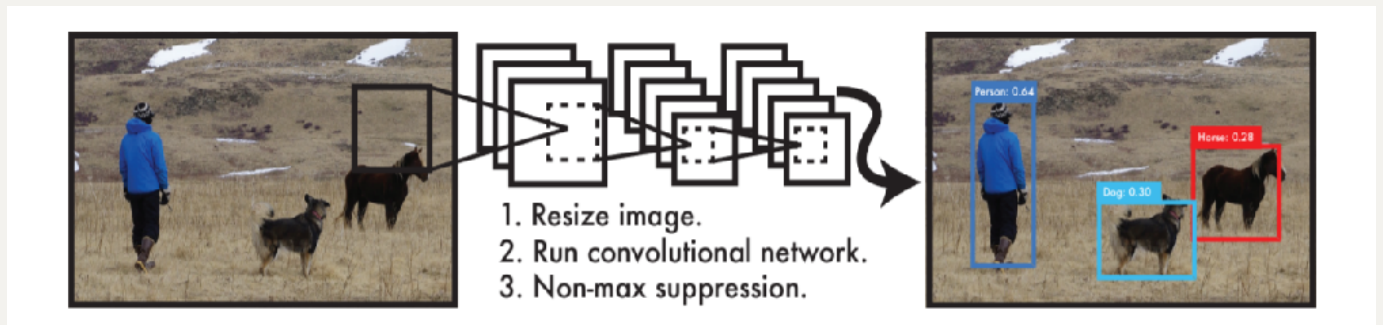
项目介绍：<https://pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>

本次实验旨在了解使用 yoloV3 进行对象检测。该项目使用预训练的 yolov3 模型实现了图像和视频对象检测分类器。yolov3 模型取自 2018 年发布的官方 yolov3 论文。yolov3 实现来自 [darknet](#)。此外，该项目还实现了使用网络摄像头实时执行分类的选项。

### yolo模型介绍

YOLO模型是用来进行目标检测的，也就是在一副图像中检测到物体并识别出物体在图像中的位置。

- 输入：图片/视频
- 输出：目标的位置信息（比如左上角和右下角的坐标）、目标的预测类别、目标的预测置信度（confidence）。



YOLOv3 (You Only Look Once, Version 3) 是一种实时目标检测模型，由 Joseph Redmon 和 Ali Farhadi 在 2018 年提出。它是 YOLO 系列模型的第三代版本，融合了速度和准确性的平衡，非常适合应用于需要实时处理的大规模目标检测任务中。YOLOv3 很强大，速度上比 R-CNN 快 1000 倍，比 Fast R-CNN 快 100 倍。检测准确率上，它不是最准的：YOLOv3-608 比 DSSD 更高，接近 FPN。但是它的速度不到后二者的 1/3。

#### <u>模型特点：</u>

1. 单阶段检测器：YOLOv3 使用单一卷积神经网络预测图像中目标的类别和边界框，不像 Faster R-CNN 等方法需要额外的候选区域生成阶段，因此具有较快的检测速度。
2. 多尺度预测：YOLOv3 在网络的不同层次进行目标检测，能够更好地捕捉大目标和小目标的信息。这种特性使其在复杂场景中的检测效果更加鲁棒。
3. 分类器改进：YOLOv3 使用了 Darknet-53 作为特征提取的骨干网络，相较于之前的 Darknet-19，模型深度更大，特征表达能力更强，并引入了残差连接以缓解梯度消失问题。
4. 无目标分类的 **Softmax**：YOLOv3 使用了二元交叉熵损失 (Binary Cross-Entropy Loss) 替代 Softmax，从而能够更好地处理多标签分类问题。
5. 锚框机制 (**Anchor Boxes**)：模型采用预定义的锚框预测目标边界框的偏移量，这种机制能够提升模型对多种目标尺寸和形状的适应能力。
6. 快速且高效：YOLOv3 在主流 GPU（如 GTX 1080 Ti）上能以较高的帧率运行，同时保持较高的准确性，非常适合实时任务。

优势和局限性:

- **优势:** 与其他目标检测方法相比, YOLOv3 的检测速度非常快, 适合实时应用; 多尺度预测和改进的分类器提升了模型在复杂场景中的表现。
- **局限性:** 与两阶段检测器相比 (如 Faster R-CNN), YOLOv3 的检测精度可能在某些情况下略有不足, 尤其是对小目标或遮挡目标的检测。

YOLOv3 广泛应用于自动驾驶、安防监控、医疗影像分析等领域, 并为后续版本 (如 YOLOv4 和 YOLOv5) 奠定了重要基础。

网址: <https://pjreddie.com/darknet/yolo/>

## 运行方法

1. 根据本地机器上存储的图像进行推断

```
python3 yolo.py --image-path='/path/to/image/'
```

2. 根据本地机器上存储的视频进行推断

```
python3 yolo.py --video-path='/path/to/video/'
```

3. 通过网络摄像头进行实时推断

```
python3 yolo.py
```

注意: 如果权重文件不在 yolov3-coco 目录中, 则在调用时修改路径。

```
yolo.py --help
```

## 实验要求

基本要求: 完成基于 yolo V3 模型的图像目标检测任务, 报告中需要包含详细的 YOLO 模型结构解释 (可任选 YOLO 版本,  $\geq 3$ )、实验结果。

优化建议（根据自己的实际情况完成即可）：

- 本次实验提供的代码是检测单张图片/视频的效果，可以自主选择合适的数据集与评估指标，修改代码，使用数据集对模型效果进行检测。
- 使用其他的yolo版本模型（可以选择开源已训练好的模型进行测试，也可以自主训练）完成本实验，对比效果。

## 实验二：faster-rcnn

### 实验描述与要求

本实验使用faster-rcnn网络进行检测。提供的代码已经包含模型训练过程。可以阅读并尝试运行本实验代码，完成训练及测试任务，并撰写实验手册。本实验主要为了理解faster-rcnn的主要结构和流程。

### RCNN模型介绍

R-CNN（Regions with Convolutional Neural Network Features）是目标检测领域的一种经典模型，由 Ross Girshick 等人在 2014 年提出。R-CNN 是目标检测迈向深度学习时代的重要里程碑，它利用卷积神经网络（CNN）提取图像特征，从而显著提升了目标检测的准确性。

模型特点：

1. 候选区域生成：R-CNN 使用选择性搜索（Selective Search）方法从图像中生成大约 2000 个候选区域（Region Proposals），这些区域可能包含目标。候选区域的生成阶段独立于神经网络。
2. CNN特征提取：对于每个候选区域，R-CNN 将其调整为固定大小（例如  $224 \times 224$ ），然后输入一个预训练的 CNN（如 AlexNet）提取特征。这些特征被用于后续的分类和边界框回归。
3. 分类和回归：
  - 分类：提取的特征通过一个支持向量机（SVM）分类器，用于判定目标类别或背景。
  - 边界框回归：通过线性回归修正候选区域的边界框坐标，使其更准确地贴合目标。

4. 分阶段处理：R-CNN 的处理是分阶段的，包含候选区域生成、特征提取、分类和回归多个步骤。

模型优势：

- 高精度：R-CNN 利用 CNN 提取深度特征，显著提升了目标检测的性能，相较于传统方法（如 HOG + SVM）效果更优。
- 可扩展性：模型可以通过更换 CNN 主干（如 VGG 或 ResNet）进一步提高检测性能。

模型局限性：

1. 计算效率低下：每个候选区域都需要独立地通过 CNN 提取特征，导致检测速度非常慢。
2. 存储开销大：由于需要对每个候选区域单独存储特征，模型的存储需求很高。
3. 不适合实时检测：多阶段处理的设计使得 R-CNN 难以满足实时应用的需求。

## faster-rcnn模型介绍

Faster RCNN是由 Shaoqing Ren 等人于2015年提出的一种高效的目标检测模型，它在 RCNN和Fast RCNN的基础上引入了区域提议网络（Region Proposal Network, RPN），通过端到端的方式实现了更快速、精确的目标检测。

Faster RCNN的主要结构和流程如下：

### 1. 特征提取

整张图像通过一个共享的卷积神经网络（通常为VGG或ResNet）提取特征图，这种共享机制避免了RCNN对每个候选区域单独提取特征的冗余计算。

### 2. 区域提议网络（RPN）

RPN直接在特征图上生成候选区域（region proposals）。通过滑动窗口机制，RPN在每个位置生成一组预定义的候选框（称为anchors），并通过分类和回归来筛选并调整这些候选框。

- 分类：判断每个候选框是否包含目标。
- 回归：调整候选框的位置和大小，使其更贴近真实目标。

### 3. ROI池化

通过RPN生成的候选区域被映射到特征图上，利用ROI（Region of Interest）池化操作将每个候选区域统一调整为固定大小，便于后续处理。

### 4. 目标分类与边界框回归

调整后的特征通过全连接层，分别进行目标分类和边界框的精确回归，输出最终的检测结果。

优势：

- **高效性**  
通过RPN替代传统的候选区域生成方法（如Selective Search），将候选区域生成和特征提取整合在一个网络中，极大提高了检测速度。
- **端到端训练**  
Faster RCNN实现了从图像到检测结果的全流程端到端训练，优化更加高效。
- **通用性**  
RPN模块可以与各种卷积神经网络结合（如VGG、ResNet），适配性强。

不足：

- **实时性不足**  
虽然相比RCNN和Fast RCNN速度有了显著提升，但在实时应用（如视频检测）中仍显得偏慢。
- **多阶段训练复杂**  
Faster RCNN虽然实现了端到端训练，但RPN和后续检测头的训练依然需要分阶段调整。

应用：

Faster RCNN是目标检测领域的里程碑式算法，其结构奠定了现代目标检测框架的基础。许多后续改进模型（如Mask RCNN、Cascade RCNN）均基于Faster RCNN进行扩展，被广泛应用于自动驾驶、安防监控、医疗影像分析等领域。

## 模型对比



Faster RCNN 和 YOLO 都是经典的目标检测算法，它们的目标都是从输入图像中检测出目标位置（Bounding Box）并分类。然而，这两种模型的设计理念和实现方式有显著不同。

1. 设计思路

• **Faster RCNN**

Faster RCNN属于两阶段检测器（**Two-Stage Detector**），其核心思想是先生成候选区域，再对这些区域进行精细分类和回归。具体来说：

- a. 第一阶段：通过RPN（区域提议网络）生成一组候选框。
- b. 第二阶段：对每个候选框进行分类和边界框回归。

这种分阶段的处理方式使得Faster RCNN在目标检测的精度上表现非常强大，特别适合对检测精度要求高的任务。

• **YOLO（You Only Look Once）**

YOLO是单阶段检测器（**One-Stage Detector**）的代表，其核心思想是将目标检测任务转化为一个单一的回归问题。具体来说：

- a. 整个图像被划分为固定的网格（grid）。
- b. 每个网格直接预测一组固定数量的边界框和类别概率。

YOLO通过一次前向传播直接输出检测结果，无需额外的候选区域生成步骤，因此检测速度非常快，适合实时应用。

2. 性能对比

特性	FASTER RCNN	YOLO
速度	较慢（毫秒级，适合离线应用）	较快（实时检测，适合在线应用）
精度	高精度，特别适合复杂背景和小目标检测	精度略低，容易漏检小目标
计算复杂度	复杂，包含RPN和ROI池化等模块	简单，高度优化的一步检测
灵活性	更适合精细检测任务	更适合实时检测任务

3. 适用场景

• **Faster RCNN**

更适合对精度要求高的应用场景，如：

- 医学影像分析
- 高分辨率遥感图像检测

- 精细化安防监控
- **YOLO**  
更适合对速度要求高的应用场景，如：
  - 实时交通监控
  - 自动驾驶
  - 嵌入式设备目标检测

#### 4.技术传承与互补关系

- **Faster RCNN**代表了传统两阶段检测器的精细化路线，追求高检测精度。
- **YOLO**推动了单阶段检测器的实时性发展，强调速度和端到端的优化。

注意：两个实验选择一个完成即可。