

实验题目	实验 6-物体检测		实验日期	2022. 6. 3	
班级		学号		姓名	郑晟赫、朱永葵、林煜鹏

CS32262 模式识别与深度学习实验

实验报告

一、实验目的（介绍实验目的）

1. 实验选择完成物体检测任务

二、实验环境（介绍实验使用的硬件设备、软件系统、开发工具等）

1. 硬件设备：CPU: i7-9750H; GPU: NVIDIA GeForce GTX 1650; 16G RAM; 512GHD Disk
2. 软件系统：Windows 10; Python 3.7.10; PyTorch 1.9.0
3. 开发工具：PyCharm

三、实验内容（介绍实验过程、设计方案、实现方法、实验结果等）

1. 任务描述

实验六选择完成物体检测任务，该任务意在判断一幅图像上是否存在感兴趣物体，并给出物体分类及位置等。研究角度来看，物体检测是计算机视觉根本问题之一，是很多高层视觉任务的基础（目标跟踪、动作识别、行为描述等）；从应用角度来看，在目前诸如视频监控、自动驾驶等领域，物体检测也表现出了广泛的应用需求。一个物体检测样例如图 1。

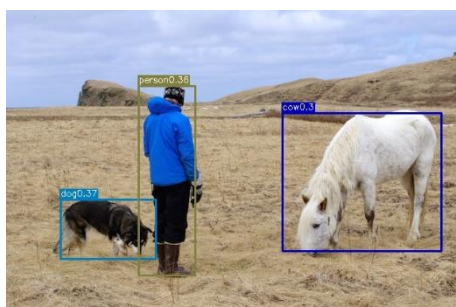


图 1 物体检测任务示例

2. 数据集分析

2.1 数据集描述

由于算力限制，因此实验选用 VOC2012 数据集。该数据集可用于图像分类、目标检测、目标分割、行为识别等任务，本实验使用其中的目标检测部分。该数据集分为 20 个目标类别，图 1 展示了所有类别的目标及其所属类别。

Table 1 The VOC classes

Vehicles	Household	Animals	Other
Aeroplane	Bottle	Bird	Person
Bicycle	Chair	Cat	
Boat	Dining table	Cow	
Bus	Potted plant	Dog	
Car	Sofa	Horse	
Motorbike	TV/Monitor	Sheep	
Train			

图 2 数据集描述

实验题目	实验 6-物体检测		实验日期	2022. 6. 3	
班级		学号		姓名	郑晟赫、朱永燊、林煜鹏

2.2 预处理

在 VOC 文件中对于每张图片有一个 xml 格式描述文件，其中记录了在这张图中出现的所有物体种类和对应的物体选择框的坐标。其详细信息如以下代码片段所示：

```

1. <annotation>
2.     <folder>VOC2012</folder>
3.     <filename>2007_000027.jpg</filename>
4.     <source>
5.         <database>The VOC2007 Database</database>
6.         <annotation>PASCAL VOC2007</annotation>
7.         <image>flickr</image>
8.     </source>
9.     <size>
10.        <width>486</width>
11.        <height>500</height>
12.        <depth>3</depth>
13.    </size>
14.    <segmented>0</segmented>
15.    <object>
16.        <name>person</name>
17.        <pose>Unspecified</pose>
18.        <truncated>0</truncated>
19.        <difficult>0</difficult>
20.        <bndbox>
21.            <xmin>174</xmin>
22.            <ymin>101</ymin>
23.            <xmax>349</xmax>
24.            <ymax>351</ymax>
25.        </bndbox>

```

在处理过程中，为了方便读取数据，因此只选取每个标识文件中的框信息，存储在一个 txt 文件中，txt 文件中每行存储图片名称与其中的框信息，一个图片中可能有多个物体没有多个框，则 xml 文件中描述的顺序存储。由于数据集已经分好训练集与测试集，因此不再对数据集进行划分。

3. 方案设计

实验主要使用的是 YOLO 系列的算法，其中重点复现并训练了 YOLO V1, YOLO V3 与 YOLO V5 均使用官方预训练好的权重文件测试物体检测的性能。以下主要介绍 YOLO V1 与 YOLO V3。

3.1 YOLO V1

首先明确物体检测可以分为两个步骤：检测目标位置；对目标物体进行分类。而物体检测主流的算法框架大致分为 one-stage 与 two-stage。two-stage 算法代表有 R-CNN 系列，one-stage 算法代表有 Yolo 系列。One-stage 算法将步骤一与步骤二同时执行，输入图像只经过一个网络，生成的结果中同时包含位置与类别信息，与 two-stage 相比，one-stage 算

实验题目	实验 6-物体检测		实验日期	2022. 6. 3
班级	学号		姓名	郑晟赫、朱永葵、林煜鹏

本实验中其只作为简单对照使用，因此不展开介绍。

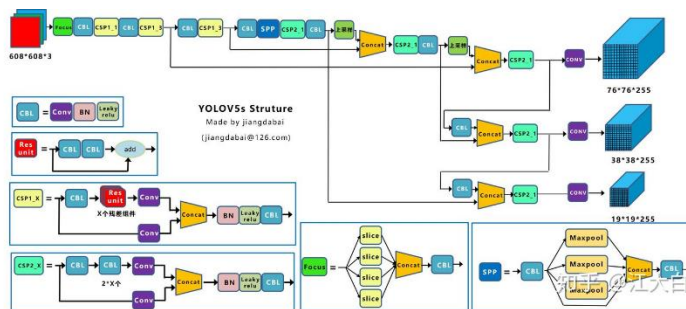


图 5 YOLO V5

3.4 损失函数

由于 YOLO 系列算法的损失函数设计的思路基本一致，因此在此只介绍 YOLO V1 使用的损失函数。损失函数计算的是预测的边框与对象类别与真实结果的差别。损失函数如下：

$$\begin{aligned}
 \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} & \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] && \text{边框中心点误差} \\
 + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} & \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] && \text{边框宽度、高度误差} \\
 + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} & (C_i - \hat{C}_i)^2 && \text{置信度误差(边框内有对象)} \\
 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} & (C_i - \hat{C}_i)^2 && \text{置信度误差(边框内无对象)} \\
 + \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} & \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 && \text{对象分类误差}
 \end{aligned}$$

图 6 YOLO 损失函数

公式中 I_i^{obj} 表示网格 i 中存在对象

公式中 I_{ij}^{obj} 表示网格 i 的第 j 个 bounding box 中存在对象

公式中 I_{ij}^{noobj} 表示网格 i 的第 j 个 bounding box 中不存在对象

总的来说，就是用网络输出与样本标签的各项内容的误差平方和作为一个样本的整体误差。损失函数中的几个项是与输出的 30 维向量中的内容相对应的。

① 对象分类的误差

公式第 5 行，注意 I_i^{obj} 意味着存在对象的网格才计入误差。

② bounding box 的位置误差

公式第 1 行和第 2 行。

a) 都带有 I_{ij}^{obj} 意味着只有“负责”（IOU 比较大）预测的那个 bounding box 的数据才会计入误差。

b) 第 2 行宽度和高度先取了平方根，因为如果直接取差值的话，大的对象对差值的敏感度较低，小的对象对差值的敏感度较高，所以取平方根可以降低这种敏感度的差异，使得较大的对象和较小的对象在尺寸误差上有相似的权重。

c) 乘以 λ_{coord} 调节 bounding box 位置误差的权重（相对分类误差和置信度误差）。YOLO 设置 $\lambda_{coord} = 5$ ，即调高位置误差的权重。

实验题目	实验 6-物体检测		实验日期	2022. 6. 3	
班级		学号		姓名	郑晟赫、朱永燊、林煜鹏

③ bounding box 的置信度误差

公式第 3 行和第 4 行。

a) 第 3 行是存在对象的 bounding box 的置信度误差。带有 I_{ij}^{obj} 意味着只有"负责" (IOU 比较大) 预测的那个 bounding box 的置信度才会计入误差。

b) 第 4 行是不存在对象的 bounding box 的置信度误差。因为不存在对象的 bounding box 应该老老实实的说"我这里没有对象", 也就是输出尽量低的置信度。如果它不恰当的输出较高的置信度, 会与真正"负责"该对象预测的那个 bounding box 产生混淆。其实就像对象分类一样, 正确的对象概率最好是 1, 所有其它对象的概率最好是 0。

c) 第 4 行会乘以 λ_{noobj} 调节不存在对象的 bounding box 的置信度的权重 (相对其它误差)。YOLO 设置 $\lambda_{noobj} = 0.5$, 即调低不存在对象的 bounding box 的置信度误差的权重。

4. 实验分析

实现网络过程中, YOLO V1 实现后使用 VOC2012 数据集进行训练之后, 在 VOC2007 测试集上测试结果。YOLO V3 复现后采用其在 COCO 数据及上预训练好的权重在 VOC2007 测试集上测试。其中 YOLO V1 的训练曲线如图 7 所示。



图 7 YOLO V1 训练曲线

YOLO V1 和 YOLO V3 测试数据结果如下:

Model	Backbone	map@voc2007test
YOLO V1	ResNet50	66.5%
YOLO V3	DarkNet53	74.48%

采用一张测试图片展示 YOLO V1、V3、V5 的性能差异:



图 8 YOLO V1 生成结果

实验题目	实验 6-物体检测		实验日期	2022. 6. 3	
班级	学号		姓名	郑晟赫、朱永燊、林煜鹏	

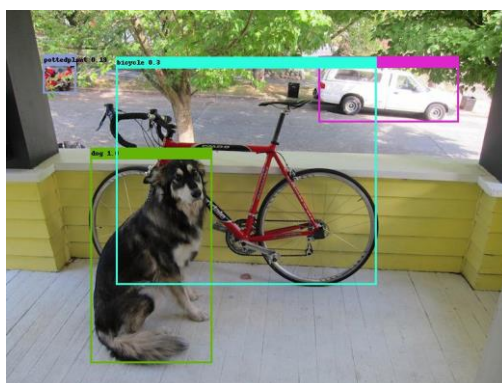


图 9 YOLO V3 生成结果

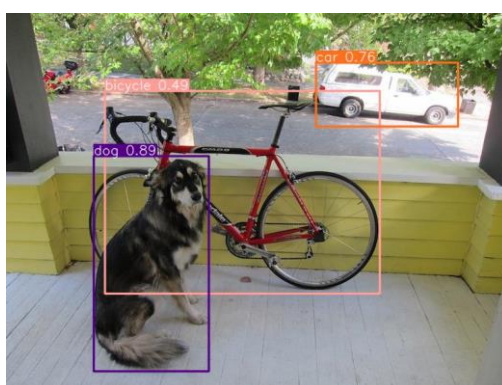


图 10 YOLO V5 生成结果

从图 8~10 可以发现，YOLO V5 的物体类别分类和预测框的生成性能在这几种网络中是最高的，这也和在整体测试集上测试的结果相同。

5. 方案评价

本实验采用了 one-stage 的代表算法 YOLO 中的三种方法在 VOC 数据集上进行测试，在目标检测算法中，YOLO 算法计算量较小，速度较快，且准确度相对也较高，且整体网络结构也并不复杂，较适合初学者入门。而由于设备限制，因此实验的测试只在较小的 VOC 数据集上测试，事实上在较新、较大的 COCO 数据集上测试更能反应三种网络的性能差异。

四、小组分工

1190300321 郑晟赫：代码完成、写报告
1190200927 朱永燊：选择数据集、写报告
1190200913 林煜鹏：题目确定、写报告