

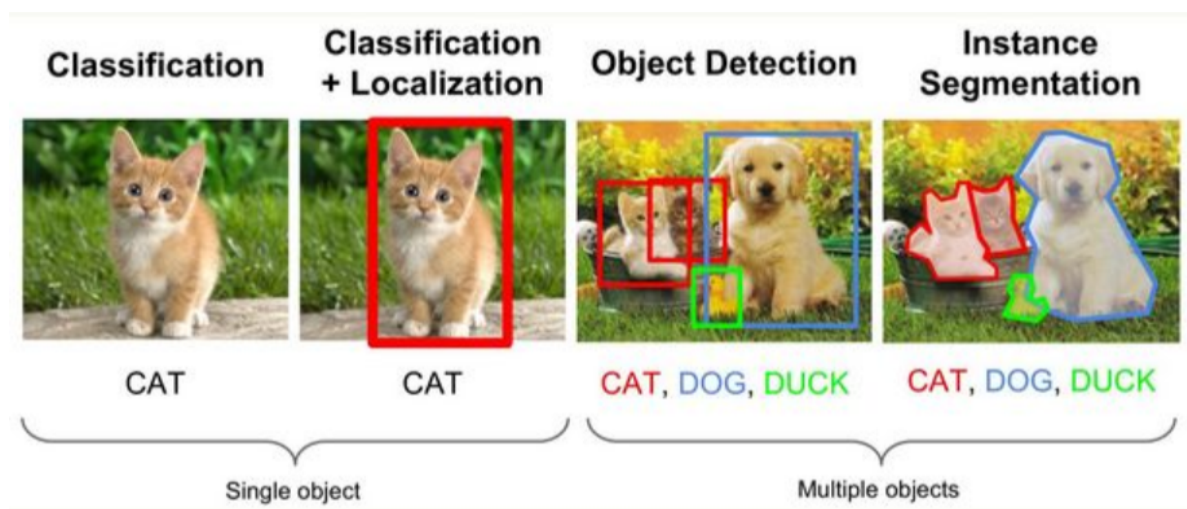
# 目标检测

## 内容介绍

- 单目标检测
- 回归问题、标签归一化、正负样本
- 数据集制作、模型、训练
- 预测效果评估：iou

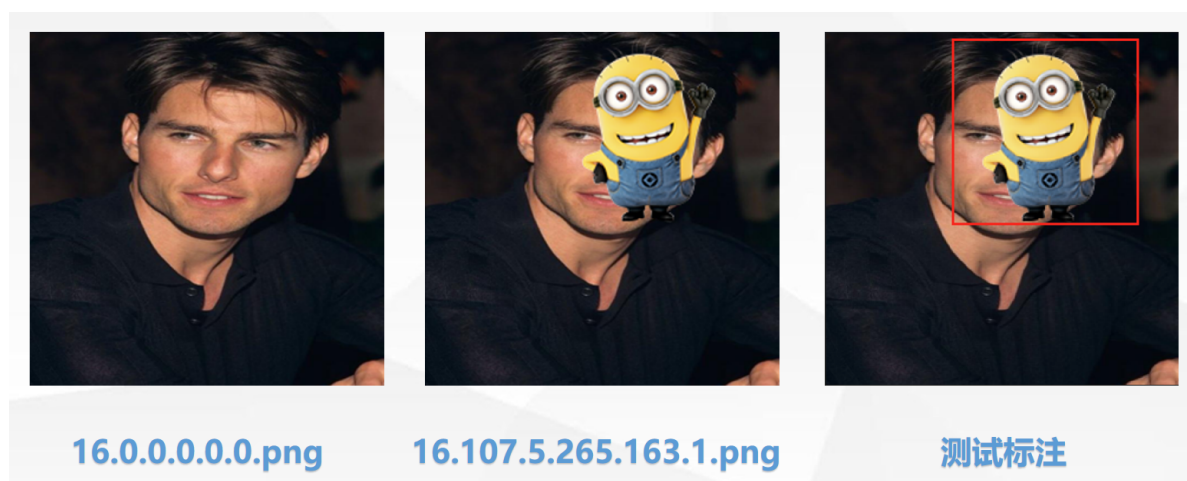
## 1 单目标检测

### 1.1 任务介绍



### 1.2 数据准备

负样本、正样本、检测效果



1. **提高模型的泛化能力：**目标检测任务通常需要先检测出图像中是否存在目标对象，然后再对目标对象进行定位。为了训练目标检测模型，需要一个分类器来判断图像中是否包含目标对象。正样本用于训练分类器以识别目标对象，而负样本则用于训练分类器以区分目标对象和背景。

只使用正样本训练的模型可能会将与目标物体相似的背景或干扰物误判为目标。通过引入负样本，模型可以学会区分目标和非目标，从而减少误判。

2. **平衡训练数据**：正负样本的存在有助于平衡训练数据集，使得模型更好地学习目标对象和背景之间的差异。如果只使用正样本进行训练，模型可能会倾向于过度拟合正样本，而无法很好地泛化到新的数据。通过引入负样本，可以帮助模型更全面地学习目标对象的特征，并提高其泛化能力。

## 数据合成

- 检测目标：小黄人
- 背景：人物
- 生成正样本、负样本



## 1.3 回归、分类问题

- 分类问题：输出变量为有限个离散变量，当个数为 2 时即为最简单的二分类问题

分类相信大家都不会陌生，生活中会见到很多的应用，比如垃圾邮件识别、信用卡发放等等，就是基于数据集，作出二分类或者多分类的选择。（可理解为离散变量预测）

- 回归问题：它处理的是连续值预测问题，即预测一个数值的输出。

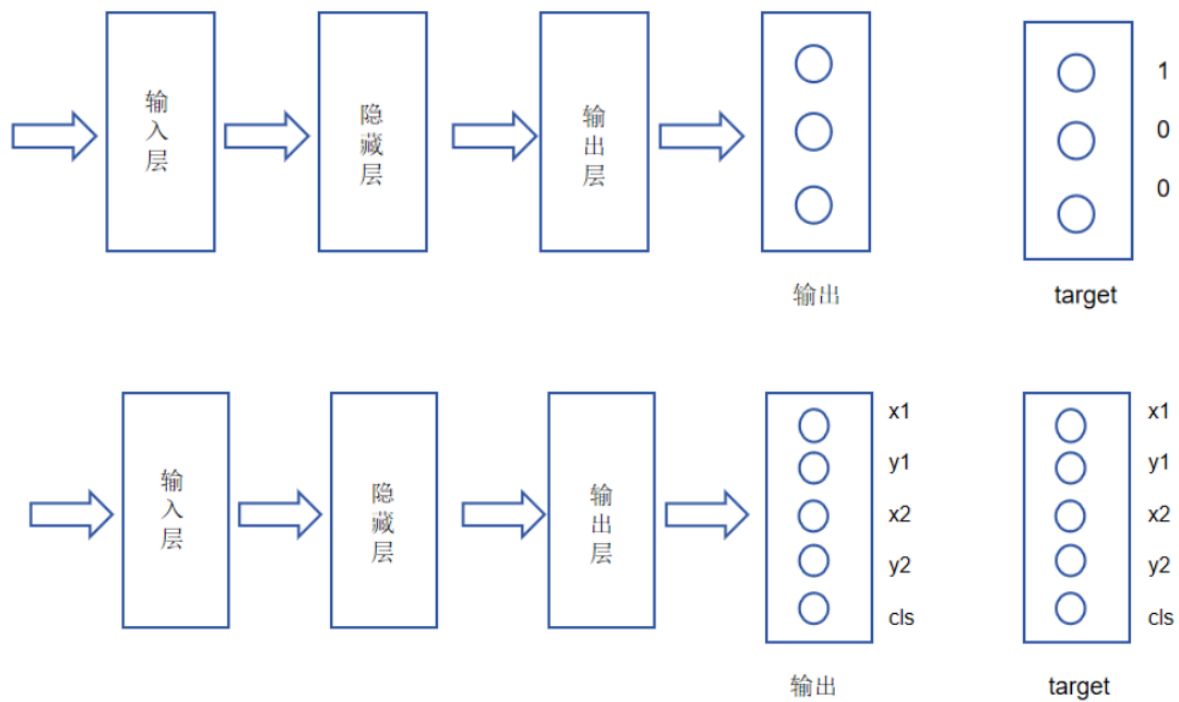
回归问题通常是用来预测一个值，如预测房价、未来的天气情况等等，例如一个产品的实际价格为 500 元，通过回归分析预测值为 499 元，回归是对真实值的一种逼近预测。（可理解为连续变量预测如之前学习的回归模型）

例如房价的数据，根据位置、周边、配套等等这些维度，给出一个房价的预测。

## 3) 训练模型

损失函数：MSELoss()、二分类交叉熵损失函数

## 2 实现方案



$x1, y1, x2, y2, conf$

### 3 代码实现


#### 3.1 数据处理

#### 3.2 网络搭建之输出使用全连接

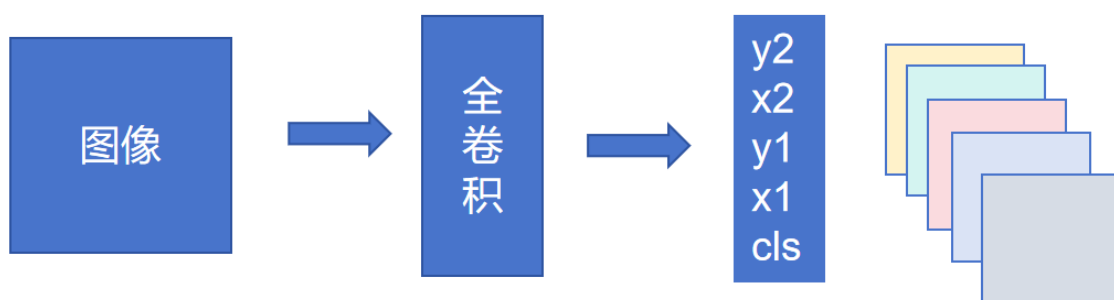
#### 3.3 模型训练

#### 3.4 iou

Intersection over Union是一种测量在特定数据集中检测相应物体准确度的一个标准

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


#### 4 代码实现2之全卷积



##### 特点:

**提高计算效率:** 卷积层在输出时通常使用1x1的卷积核，这样可以大大减少参数数量，提高计算效率。

**灵活性:** 卷积层可以接受任意大小的输入，因此可以处理不同尺寸的图像，而不需要进行额外的调整。

#### 4.1 网络修改

#### 4.2 训练代码修改

### 4.3 测试代码修改

#### 作业布置

作业布置

08:49

≡ □ 🖼

- 完成上课的练习
  - 输出使用的是全连接
    - 两个输出
      - 分类完成回归和分类任务
    - 1个输出
      - 5个值: 其中4个完成的是回归任务, 剩余1个完成分类任务
- 改成全卷积结构

The diagram illustrates a fully convolutional architecture. It starts with an input image (图像) represented by a blue square. An arrow points to a vertical blue rectangle labeled '全卷积' (Fully Convolutional). Another arrow points to a vertical blue rectangle containing the labels 'y2', 'x2', 'y1', 'x1', and 'cls'. To the right of this vector is a stack of four overlapping squares in yellow, cyan, pink, and light blue, representing feature maps.

-