上海师范大学
Shanghai Normal University

YOLO算法

姓名：端木慧娉
学号：212503015

2021年9月25日

提 纲

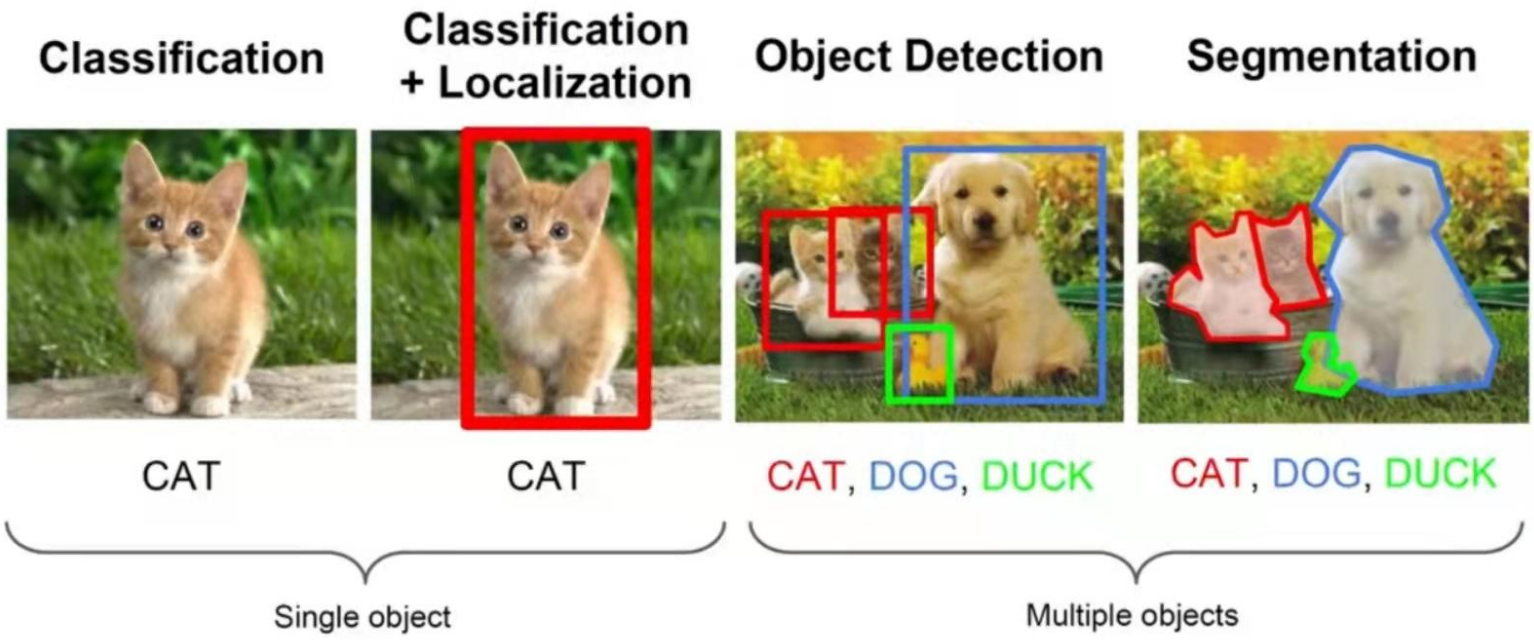
1 计算机视觉

2 yolov1算法

3 yolov5算法的实际应用

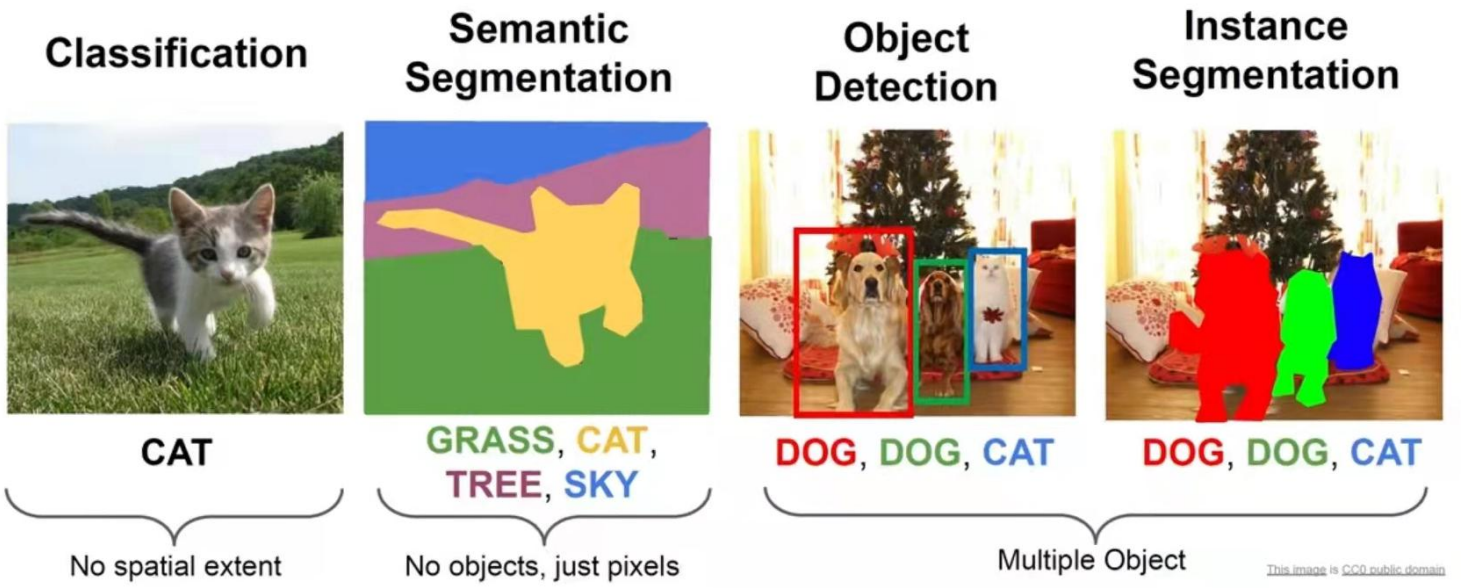
计算机视觉任务

计算机视觉可以用于解决以下问题---分类、检测、分割



计算机视觉任务

分割分为：语义分割和实例分割



提 纲

1 计算机视觉

2 yolov1算法

3 yolov5算法的实际应用

yolov1算法

传统两阶段目标检测模型 VS yolo单阶段目标检测模型

- 传统两阶段目标检测

传统的目标检测主要是使用分类器、滑动窗口、暴力遍历。

特点：

1. 提取潜在的候选框(region proposal)；
2. 使用分类器逐一筛选每个候选框。

yolov1算法

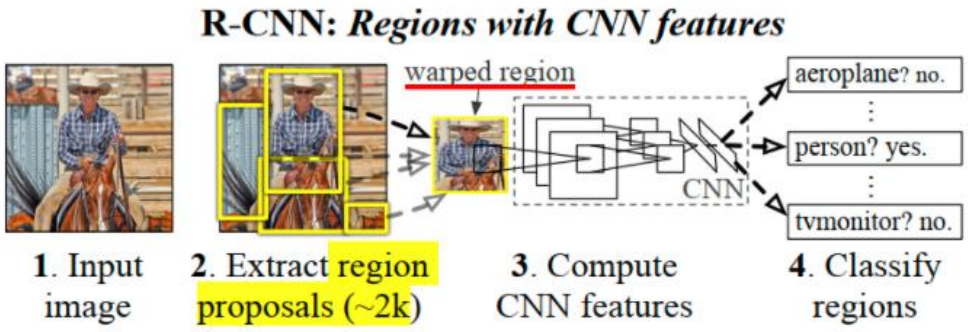
传统两阶段目标检测模型 VS yolo单阶段目标检测模型

- 传统两阶段目标检测

R-CNN： 先提取出候选框region proposals，然后用分类器对这些候选框逐一进行处理。

优点： 准确度高

缺点： 计算量大工作慢（40S处理一张图片）；难以优化（因为不是端到端，而是两阶段分开训练）



yolov1算法

传统两阶段目标检测模型 VS yolo单阶段目标检测模型

- yolo单阶段目标检测模型

无需提取候选框，直接一次向前推断得到bounding boxs定位及分类结果。

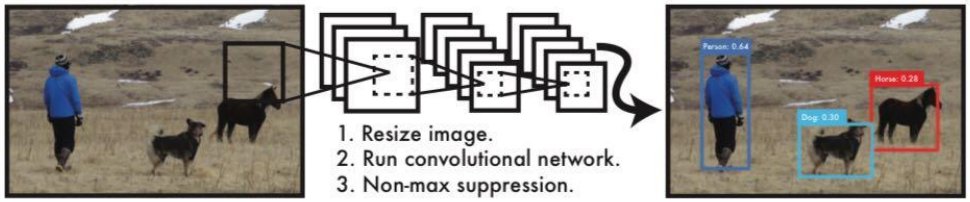


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

yolov1算法

- YOU ONLY LOOK ONCE

主要思想：yolo将目标检测设计成了一个回归问题，利用单阶段神经网络，直接从一副完整的图像中得到bounding boxs和对应的类别及其置信度。

由于它是使用单阶段神经网络，所以可以对其检测性能进行端到端的优化。

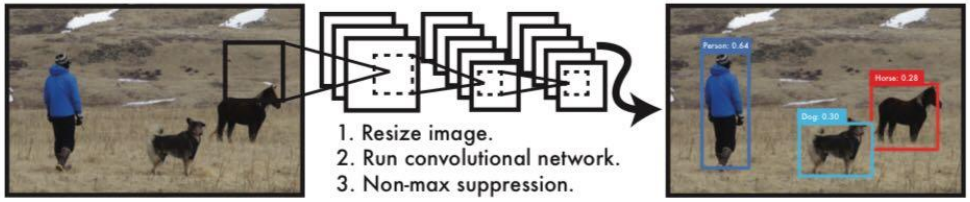


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

yolov1算法

- YOLO目标检测整体框架

对于一整张的输入图片，yolo将其划分为S*S(7*7)个grid cell，每一个grid cell复负责预测一个物体。

我们标注数据集的那个矩形框(ground truth)的中心点(x,y)如果落到了哪一个grid cell里面，就由哪一个grid cell去负责预测该物体。

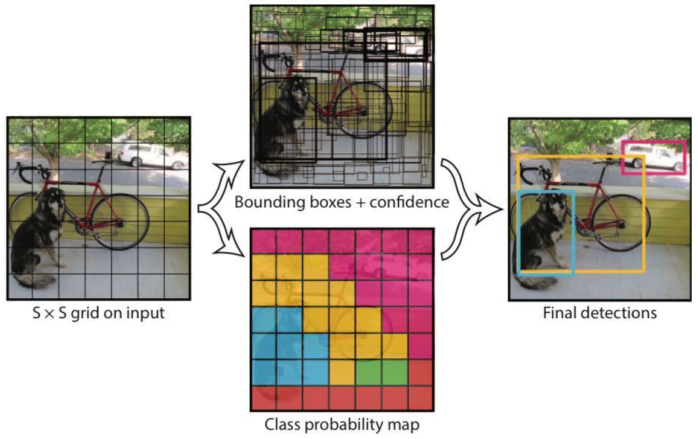


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

每一个grid cell会预测B(2)个bounding boxes。

yolov1算法

- YOLO目标检测整体框架

每一个bounding box包含5个预测值x,y,w,h和confidence。(x,y)代表的是该bounding boxes的中心点相对于其所在的grid cell的位置（归一化），w和h代表该bounding boxes相对于整幅图像的宽和高，confidence就是bounding boxes与ground truth的IOU。

每一个bounding boxes还会预测C个条件类别概率$Pr(Class_i|Object)$

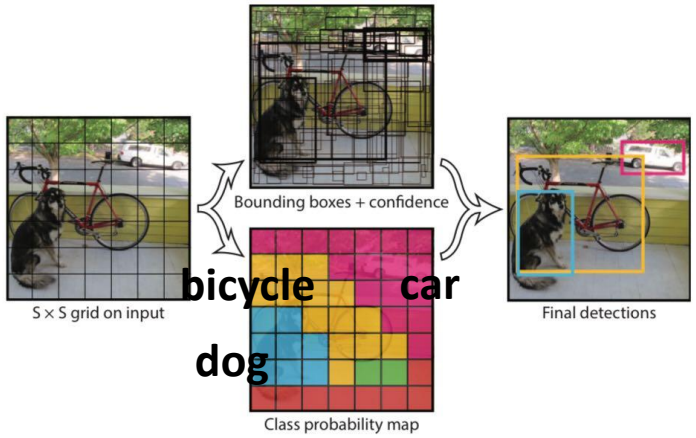


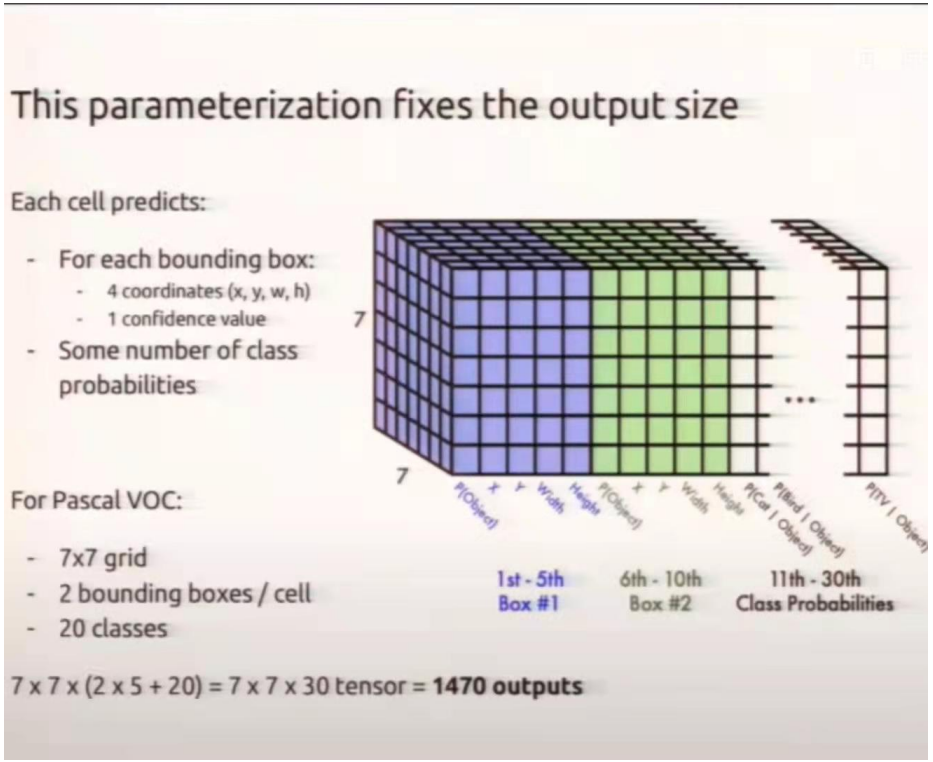
Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

yolov1算法

- YOLO目标检测整体框架

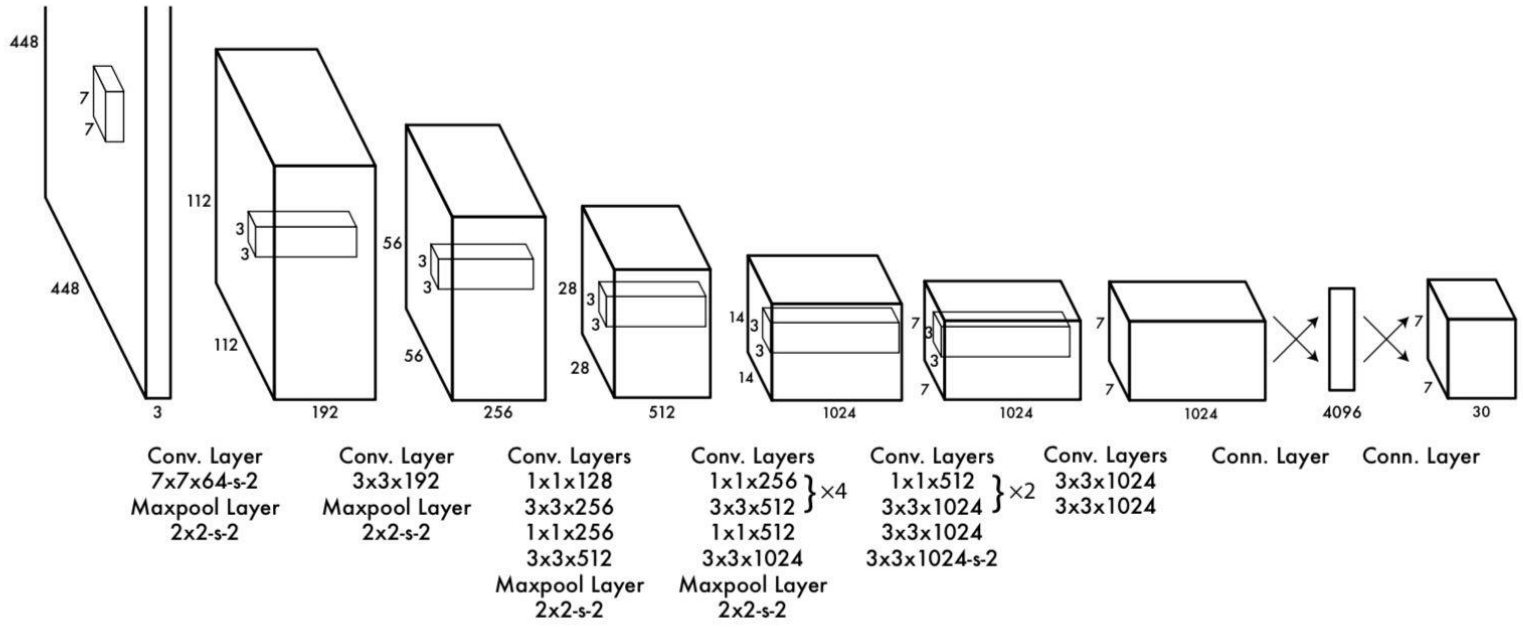
在PASCAL VOC数据集上对yolo的检测性能进行评估；
每次预测产生S*S*(C+B*5)的tensor

7*7*(20+2*5)



yolov1算法

- 网络设计



24层卷积层用来提取图像特征

2层全链接层得到7*7*30的tensor

yolov1算法

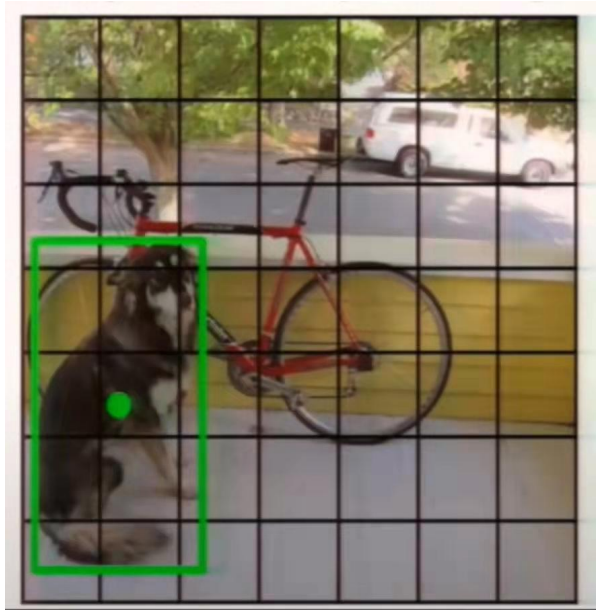
YOLO模型的训练与测试过程

- YOLO模型训练阶段

深度学习的训练主要是根据梯度下降和反向传播，迭代的微调神经元中的权重从而使损失函数最小化的一个过程。

数据集在开始训练之前需要进行标注。标注就是用一个矩形框框出目标物体。称矩形框为ground truth。对于ground truth我们通过x,y,w,h四个值对它的位置进行记录。

以上是训练之前的准备工作。

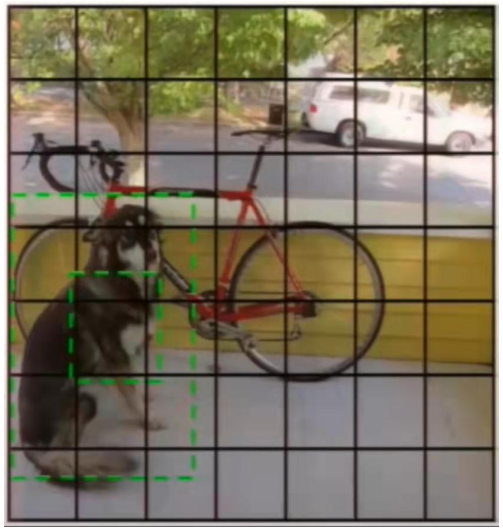


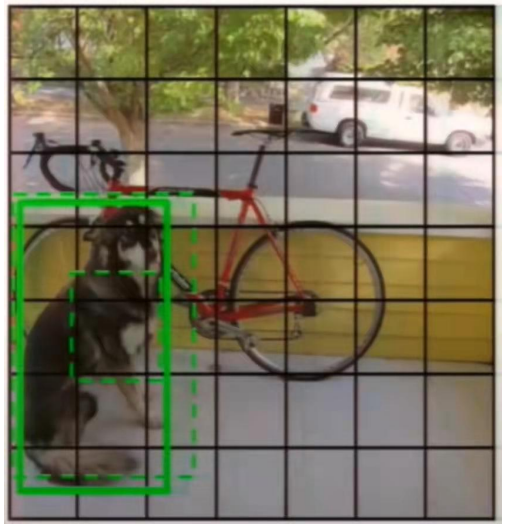
yolov1算法

YOLO模型的训练与测试过程

- YOLO模型训练阶段

中心点(x,y)落在那个grid cell里，就应该由哪个grid cell来负责预测这个物体；每一个grid cell可以生成2个bounging boxes，选择IOU较大的bounging boxes进行拟合。



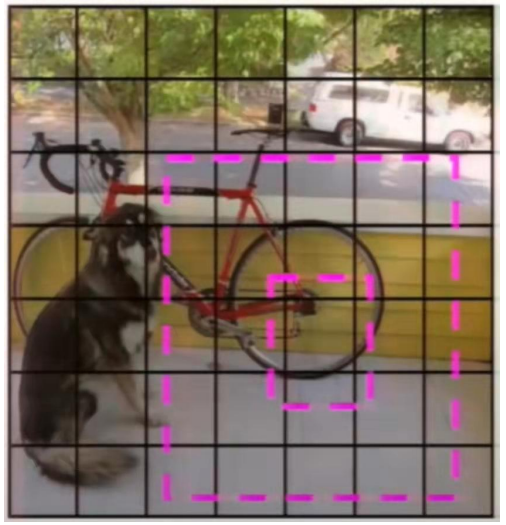


yolov1算法

YOLO模型的训练与测试过程

- YOLO模型训练阶段

对于没有中心点落在其中的grid cell，其生成的两个bounding boxes均被舍去。



由此，我们可以得到，yolo算法的损失函数。

yolov1算法

- 损失函数

loss function:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

负责检测物体的bbox的中心点定位误差

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

负责检测物体的bbox的w, h误差

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

负责检测物体的bbox的置信度误差

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

不负责检测物体的bbox的置信度误差

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 \quad (3)$$

负责检测物体的bbox的分类误差

yolov1算法

YOLO模型的训练与预测过程

- YOLO模型预测阶段

模型已训练成功。预测是将一张陌生的图片输入到训练好的神经网络里，看其是否能够达到较好的分类效果。

不需要训练和反向传播，只需要一次向前推算运行模型即可进行预测。

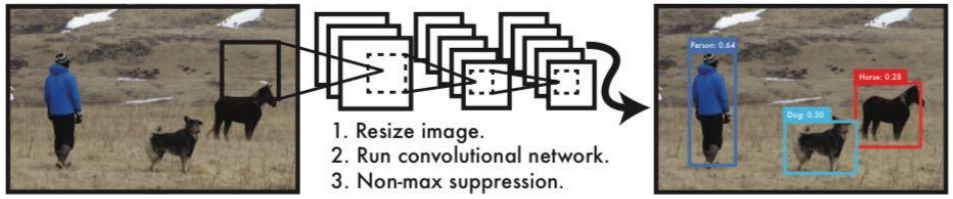
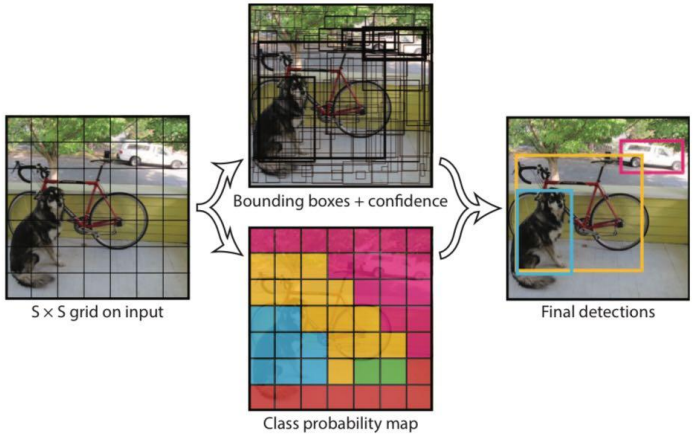


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

yolov1算法

YOLO模型的训练与测试过程

- YOLO模型预测阶段



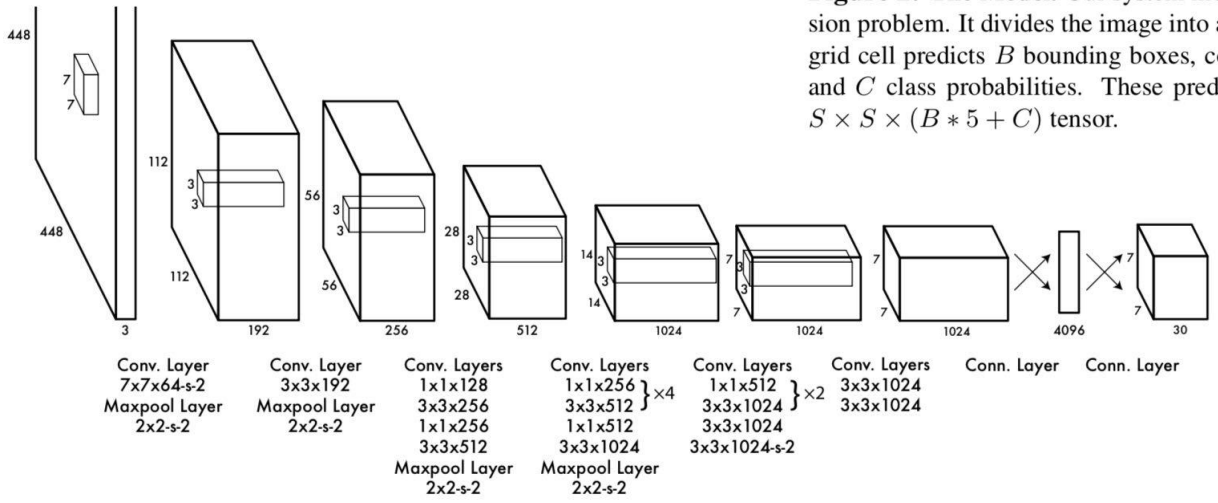


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

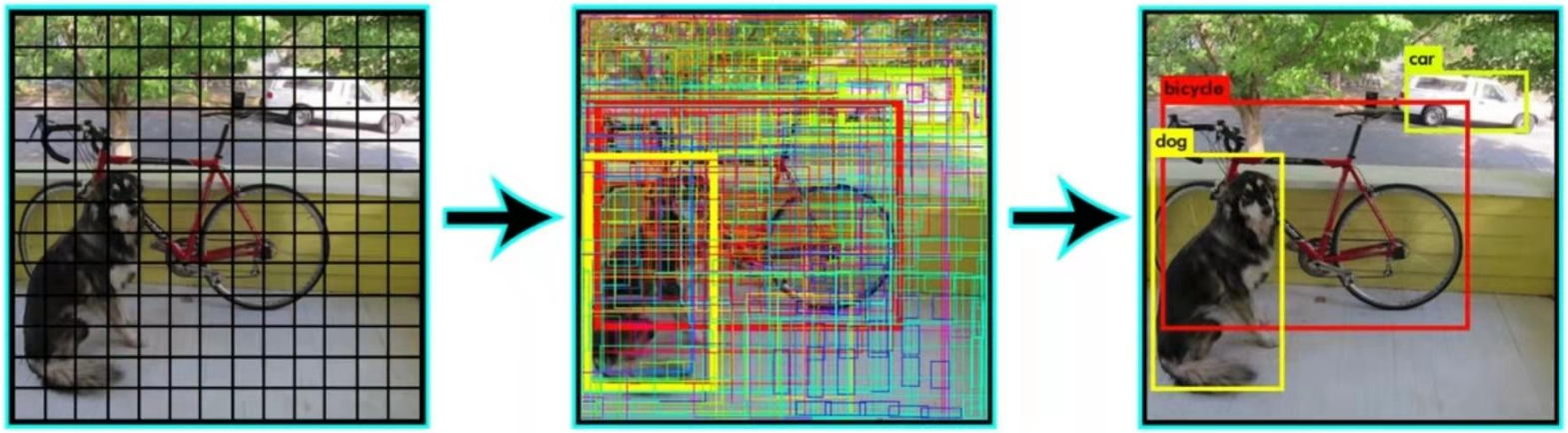
yolov1算法

YOLO模型的训练与测试过程

- YOLO模型预测阶段 后处理

后处理就是将得到的7*7*31的tensor进行处理，得到最终的检测结果。采用的方式是：置信度过滤和非极大值抑制。

后处理的主要任务：



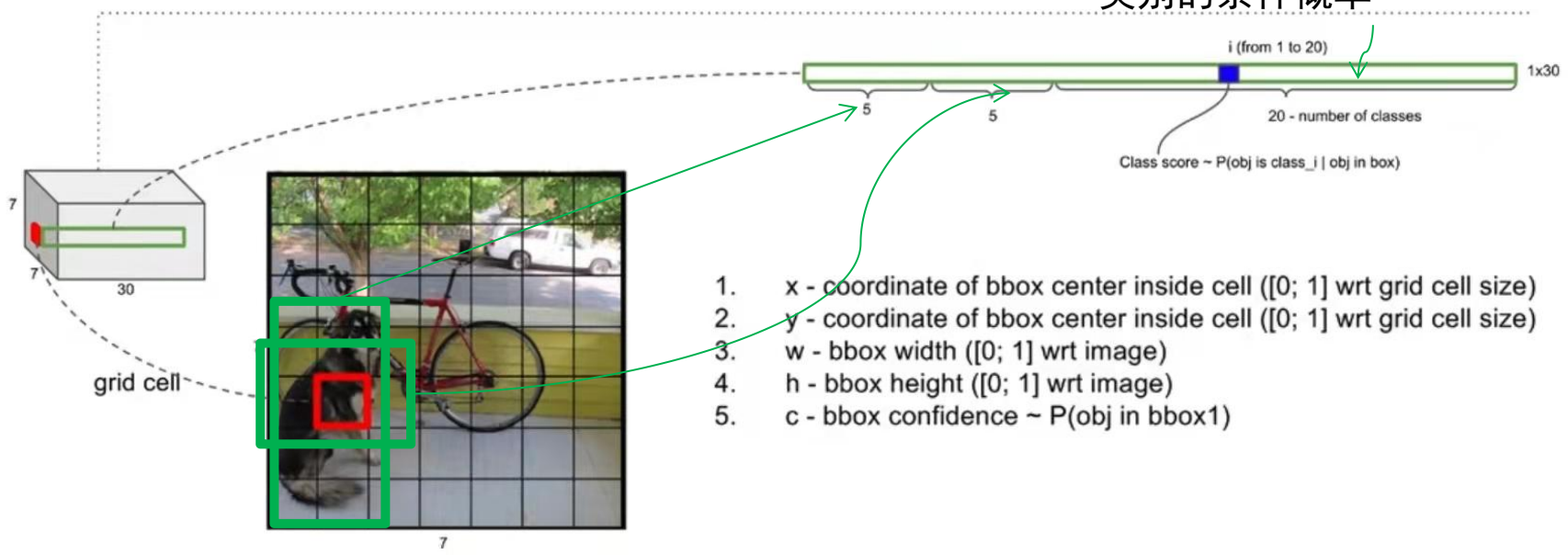
yolov1算法

YOLO模型的训练与测试过程

- YOLO模型预测阶段 后处理

7*7*30的tensor结构

该框属于其中20个类别中的某一类别的条件概率



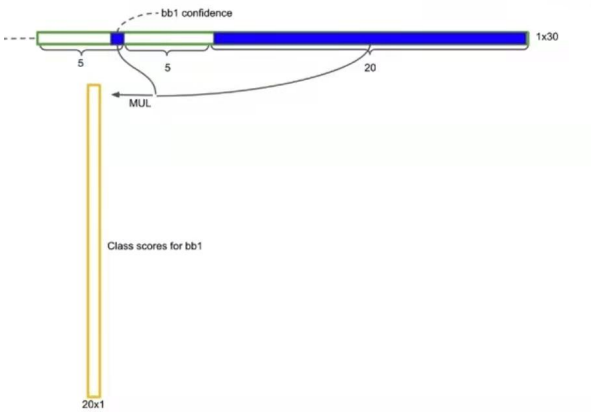
1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox confidence ~ P(obj in bbox1)

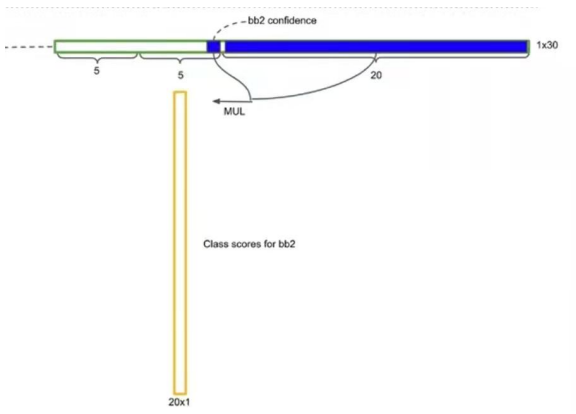
yolov1算法

YOLO模型的训练与测试过程

- YOLO模型预测阶段 后处理

得到某个框属于某类别的全概率，是一个20维的tensor。



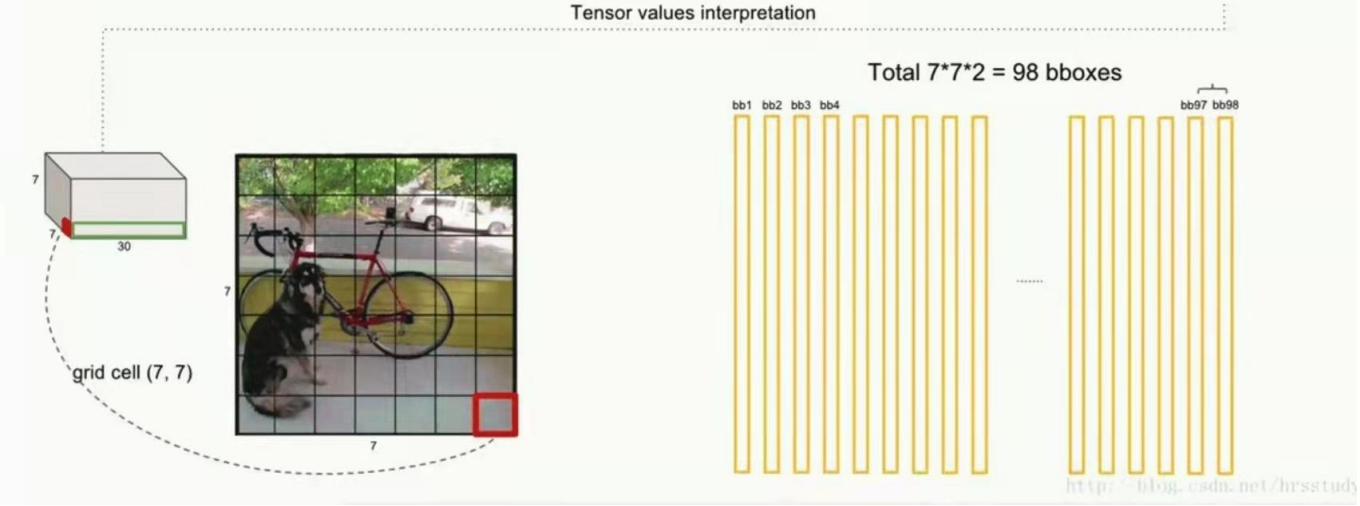


yolov1算法

YOLO模型的训练与测试过程

- YOLO模型预测阶段 后处理

对得到的20维全类别概率进行置信度过滤和非极大值抑制处理。



yolov1算法

YOLO算法优缺点

- 优点

1. 速度快。

使用GPU运算，在不使用批处理的情况下，处理速度45fps。它还有一个更快的版本faster yolo，甚至可以达到150fps。这意味着它可以对视频流进行实时处理。相对与其他的实时监测系统，yolo有他们两倍的Map（准确度性能）

2. yolo在背景误判方面比fast-CNN要少将近一半。

yolo在预测时，是对整张图片的上下文信息进行捕获。fast R-CNN,由于他是滑动窗口式的，所以可能会将背景误判为目标，因为它不能看到更大的coentxt 。

yolov1算法

YOLO算法优缺点

- 优点

3. 迁移和泛化能力较好。

当你训练的是实际生活中的景物，而检测的是艺术作品的时候，yolo相比于DMP和RCNN能更好的检测出来。所以说当它应用与一个新或者未知的领域的时候，掉性能的可能性会更小。

yolov1算法

YOLO算法优缺点

- yolo缺点

1. 在准确度上yolo相较于最好的模型性能较差。
2. 可识别的种类有限，对于较小的物体检测性能较差
3. 可识别的类别有限制

提 纲

1 计算机视觉

2 yolov1算法

3 yolov5算法实际应用

yolov5算法实际应用

- 准备工作

1.在电脑上搭建好深度学习的环境。

Ubuntu、显卡、cuda、cudnn、anaconda

2. 从github上下载YOLOV5工程文件，按照工程文件requirements.txt的要求安装python，torch，torchvision以及相关的库文件，即装配好代码需要运行的环境。

requirements - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

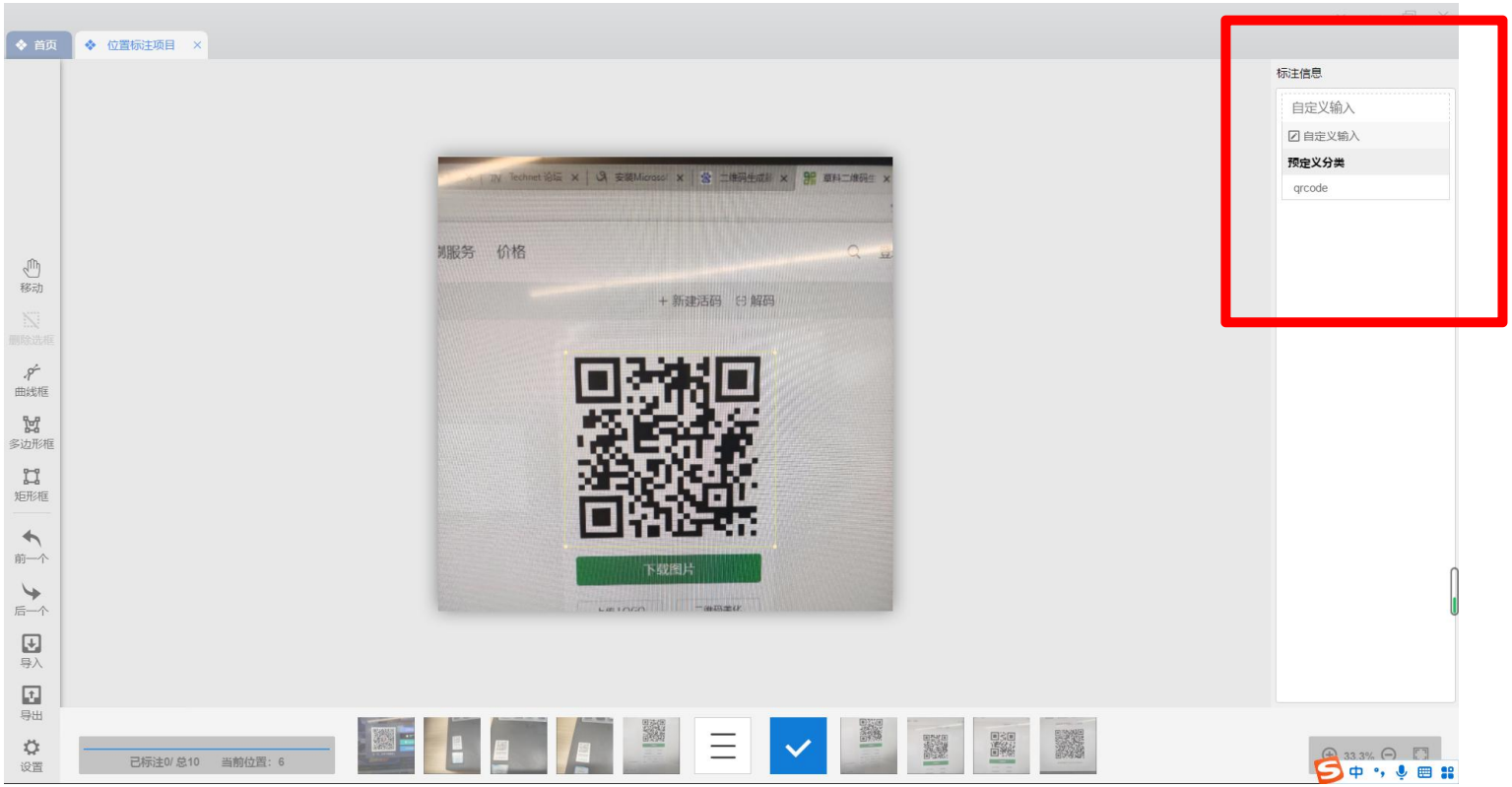
```
# pip install -r requirements.txt

# base ----------------------------------------
matplotlib>=3.2.2
numpy>=1.18.5
opencv-python>=4.1.2
Pillow
PyYAML>=5.3.1
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.41.0

# logging -------------------------------------
tensorboard>=2.4.1
# wandb

# plotting ------------------------------------
seaborn>=0.11.0
pandas

# export --------------------------------------
# coremltools>=4.1
# onnx>=1.9.0
# scikit-learn==0.19.2  # for coreml quantization

# extras --------------------------------------
# Cython  # for pycocotools https://github.com/cocodat
# pycocotools>=2.0  # COCO mAP
# albumentations>=1.0.3
thop  # FLOPs computation
```

yolov5算法实际应用

- 数据集标注

精灵标记助手

imglabel



yolov5算法实际应用

- 自制训练集

在data目录下新建四个文件夹：Annotations，images，ImageSets，labels

Annotations	存放标记后生成的xml文件
images	存放原始的图片数据集
labels	存放保存标记内容的txt
ImageSets	存放训练数据集和测试数据集的分类情况

在ImageSets文件夹中，生成4个文件：train.txt，val.txt，trainval.txt，test.txt

train.txt	用于训练的图片名称
val.txt	用于验证的图片名称
trainval.txt	train与val的合集
test.txt	用于测试的图片名称

yolov5算法实际应用

- 构建数据集

makeTxt.py文件：对图片数据集进行随机分类，以8：1：1的比例分为训练数据集，验证数据集和测试数据集，运行后在ImageSets文件夹中会出现四个文件

Vol_label.py文件：会将图片数据集标注后的xml文件中的标注信息读取出来并写入txt文件，运行后将出现所有图片的数据集的标注信息

yolov5算法实际应用

- 自制数据集的训练

1、修改数据集方面的yaml文件

```
train: data/train.txt  # 118k images
val: data/val.txt  # 5k images
test: data/test.txt  # 20k images for submi

# number of classes
nc: 2

# class names
names: ['Gingerbread', 'Coconut-milk']
```

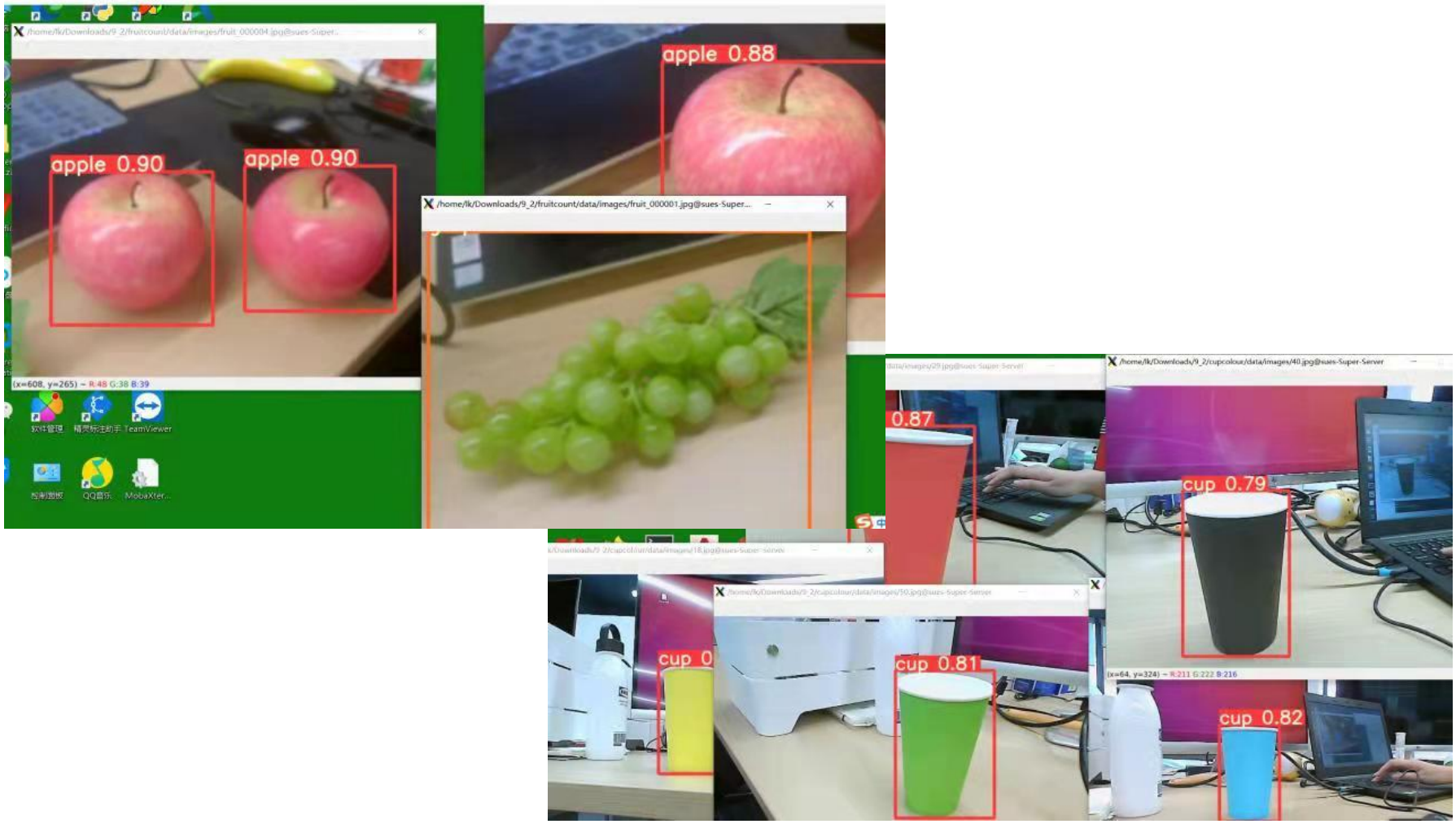
2、修改网络参数方面的yaml文件

```
1  # parameters
2  nc: 2  # number of classes
3  depth_multiple: 0.33  # model depth multiple
4  width_multiple: 0.50  # layer channel multiple
```

3、运行python train.py –-img 640 --batch 4 –epoch 300 --data ./data/xxx.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --workers 0

yolov5算法实际应用

- 训练效果



上海师范大学

谢谢！