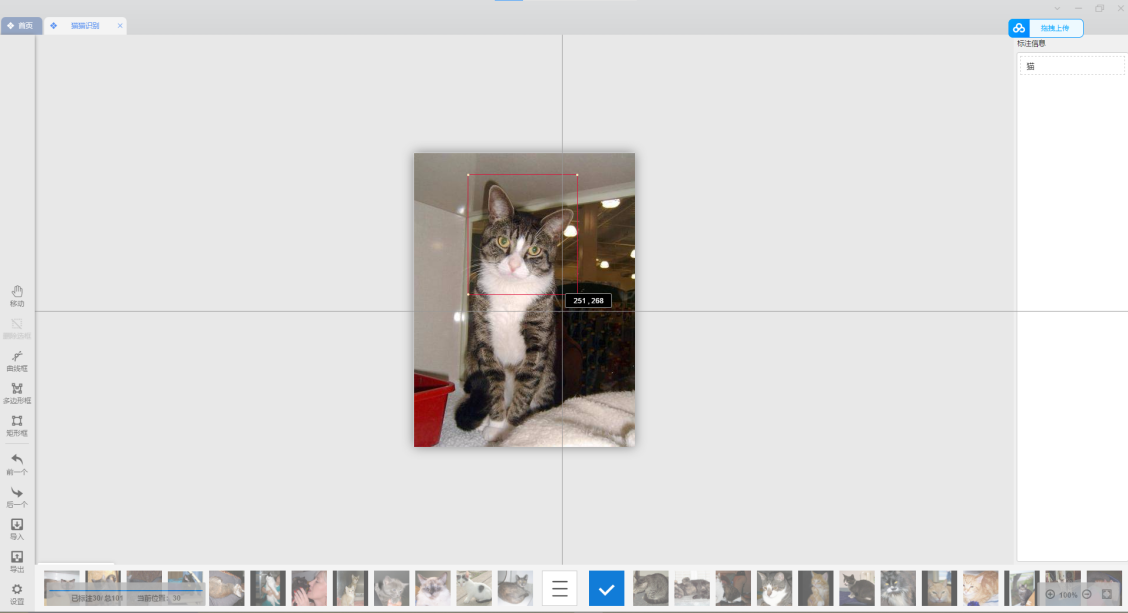
**移动通信与编程作业三——Yolov5实现猫脸识别**

**孙永鑫 520021911106**

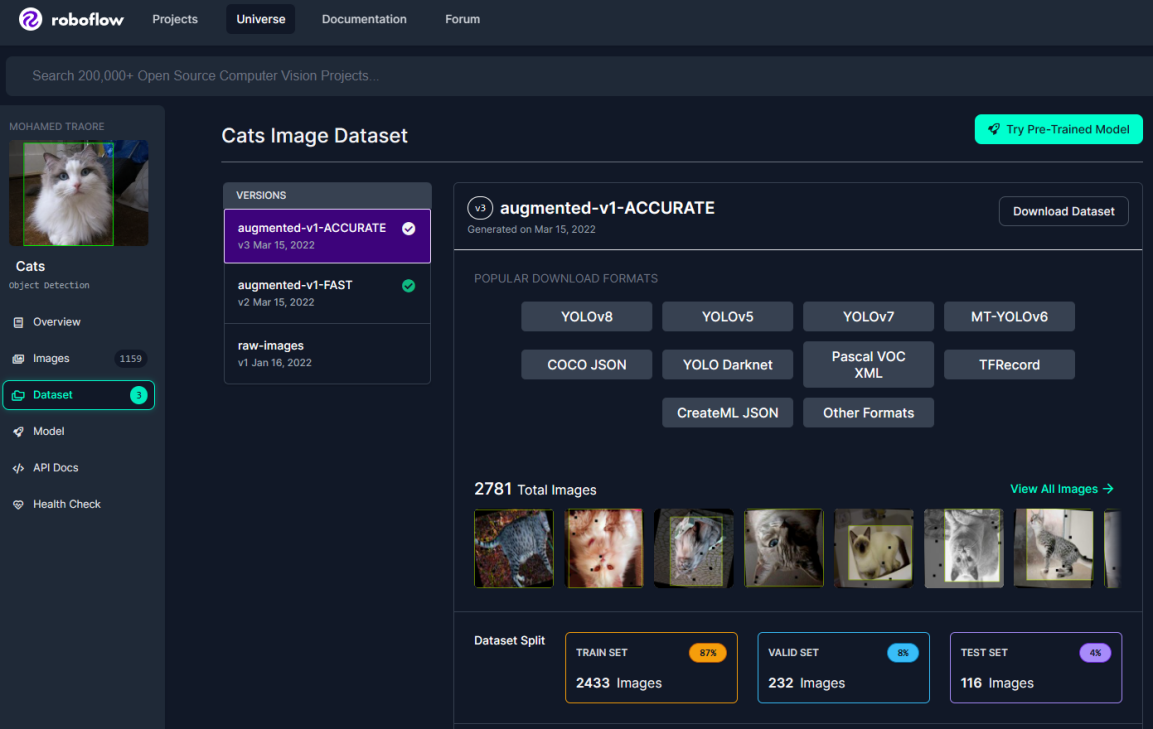
1. **准备猫脸识别数据集**

方案一：自行准备图片，使用精灵标记助手标记数据集，如下图所示：



方案二：直接下载已有猫脸识别数据集 网址：

<https://universe.roboflow.com/mohamed-traore-2ekkp/cats-n9b87/dataset/3>



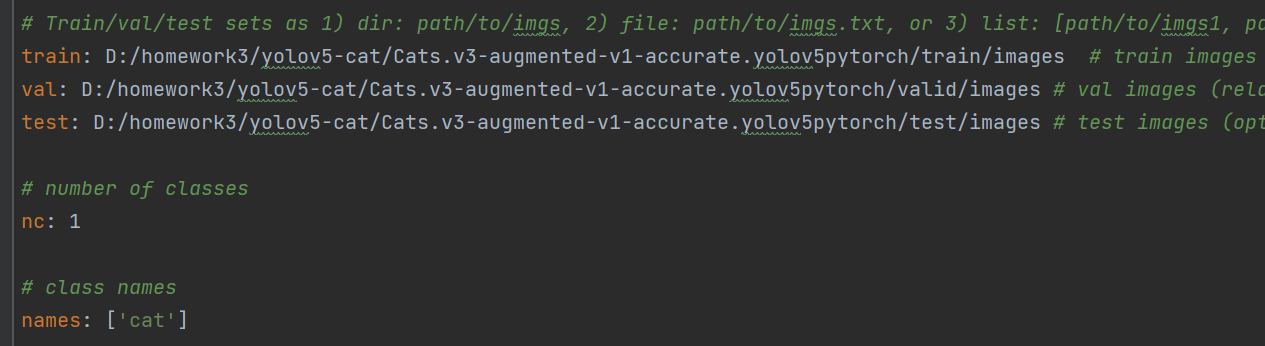
本次实验中我们选择从上述网址直接下载数据集

1. **Clone Yolov5代码**

下载Yolov5原模型 网址：<https://github.com/ultralytics/yolov5>

首先，在/data子文件夹中新建猫脸识别cat.yaml文件，修改里面的train, val, test的路径，并将分类设置成一类：cat。

需要注意的是，这里注释说明可用的路径类型可以是imgs图片文件夹，也可以是.txt样本列表文件。本次实验中我们使用的是imgs图片文件夹。



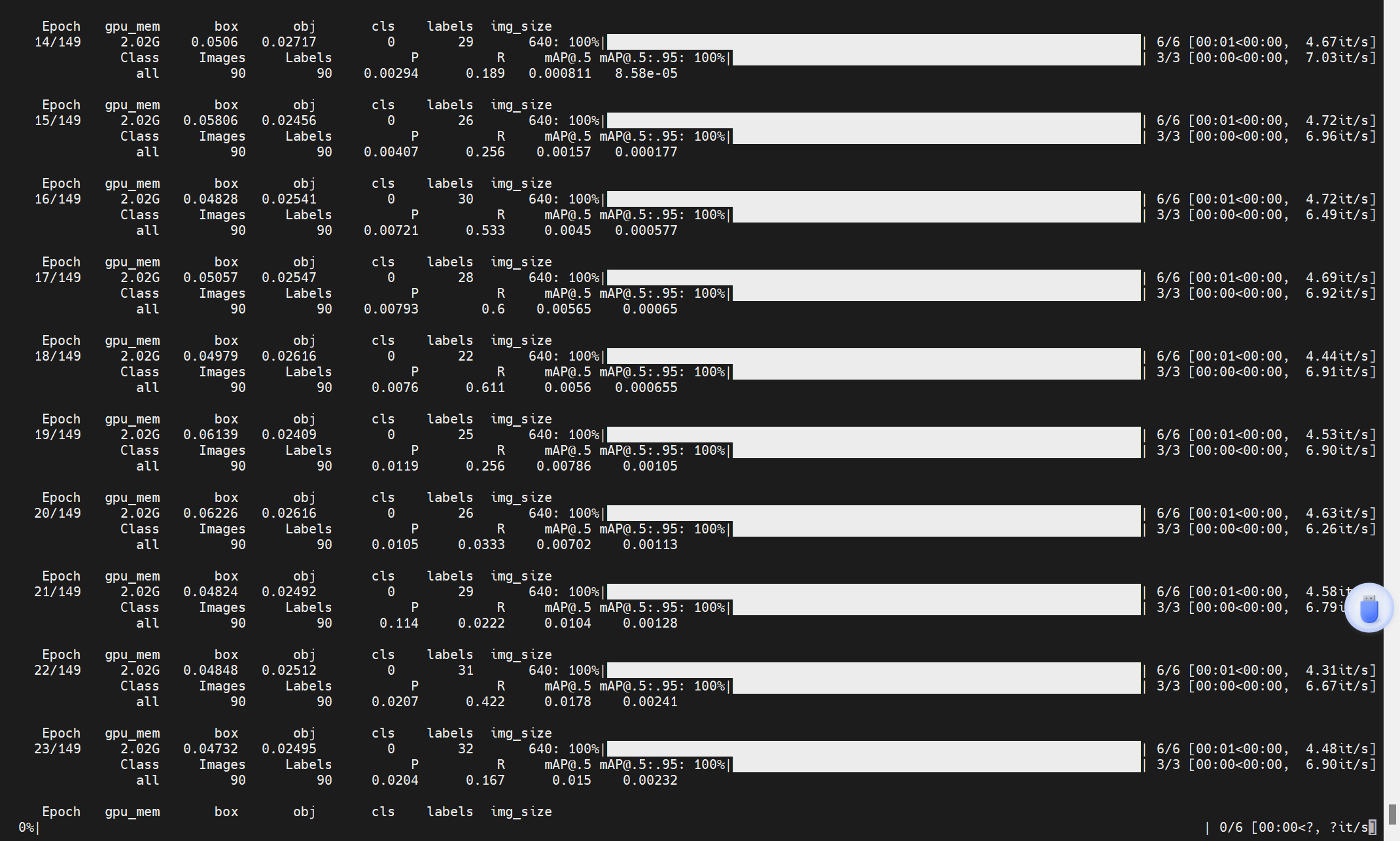
其次，修改/models文件夹下的yolov5s.yaml和yolov5x.yaml文件，将分类数改为1。该项目中使用了最小的yolov5s模型和最大的yolov5x模型作对比。



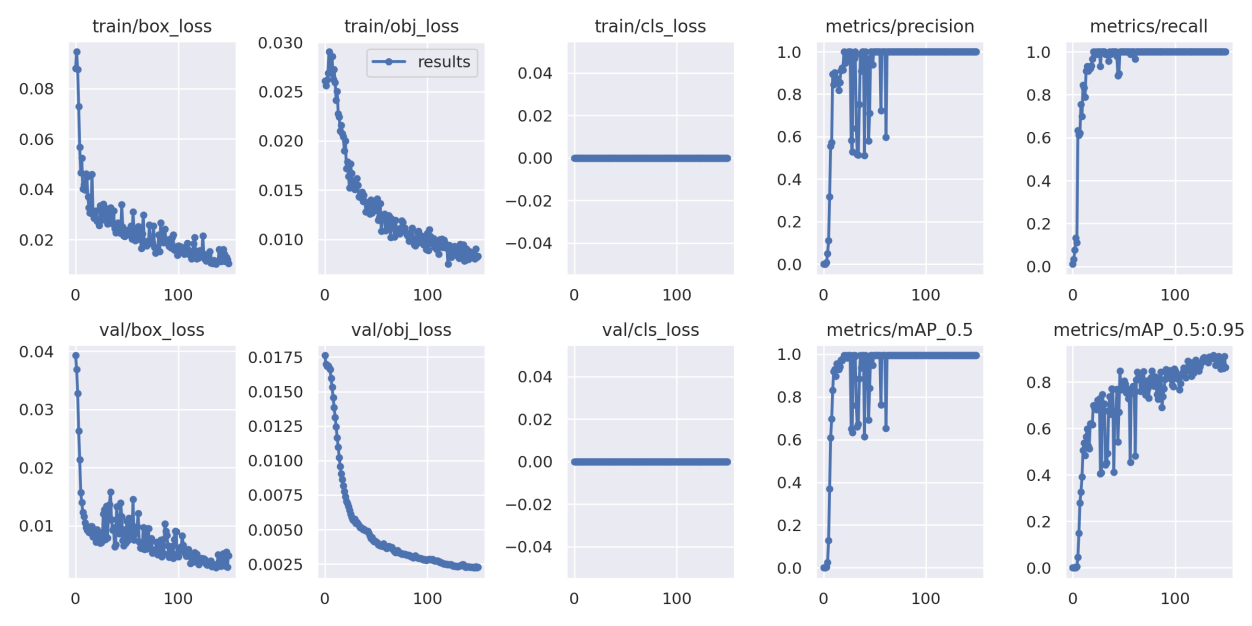
之后，我们只需要执行以下脚本即可开始训练：

**python train.py --data cat.yaml --cfg yolov5x.yaml --batch-size 16 --epochs 150**

训练中：



训练结束后，训练精度结果可以在runs/train/exp文件夹中查看：



其中：

1. 定位损失box\_loss：预测框与标定框之间的误差，越小定位得越准
2. 置信度损失obj\_loss：计算网络的置信度，越小判定为目标的能力越准
3. 分类损失cls\_loss：计算锚框与对应的标定分类是否正确，越小分类越准

可见在epoch数取150时，各类损失函数基本收敛。

1. **校园猫咪拍摄识别结果**

训练完毕后，在./runs/train/exp/weights/best.pt生成通过验证集测试的最好结果的模型。同时可以执行模型预测，初步评估模型的效果：

**python detect.py --source data/test\_images**

**--weights ./runs/train/exp/weights/best.pt --conf 1**

评估结果如下：

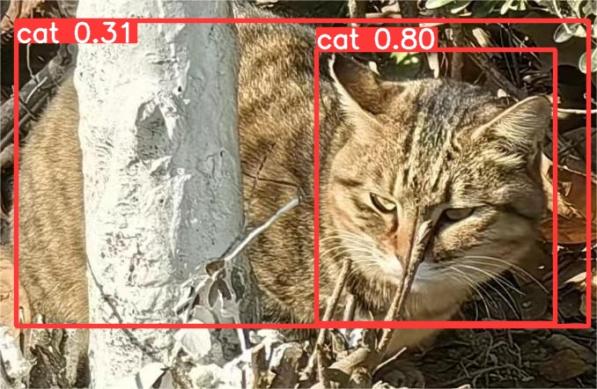
1. (b)

**图1 猫咪正面照识别结果(a)YOLOv5s (b)YOLOv5x**

1. (b)

**图2 猫咪侧面照识别结果(a)YOLOv5s (b)YOLOv5x**



1. (b)

**图3 含有遮挡物的猫咪照片识别结果(a)YOLOv5s (b)YOLOv5x**

对比图1,图2，我们可以发现猫咪正面照的识别效果更好，原因在于训练集中的猫咪正面照居多，YOLOv5学习得到的特征更适用于正面照识别。

对比两种复杂度不同的模型YOLOv5s (小模型)和YOLOv5x(大模型)，发现复杂度更高的YOLOv5x具有更好的识别效果。此外，在识别含有障碍物的猫咪图片时，我们还发现YOLOv5s会将猫咪识别成两部分，而YOLOv5x能将被障碍物遮挡的猫咪识别成一个整体，具有更好的识别精确度。