**基于深度学习的目标检测实验报告**

**实验目标：**

1. 理解常见的目标检测、语义分割方法原理，调通相应程序，并完成目标检测实践任务

**实验内容：**

在本实验中，我们将微调预训练的Mask宾夕法尼亚大学-复旦大学的R-CNN行人检测模型和分段。它包含170张图像，其中包含345个行人实例，我们将使用它来演示如何使用 TorchVision中的新功能进行训练自定义数据集上的对象检测和实例分段模型。

首先下载数据集（PennFudan数据集），然后解压缩zip文件至 'PennFudanPed' 目录。



图1 解压缩数据集文件

以下是一对图像和分割蒙版的一个示例，见图11。

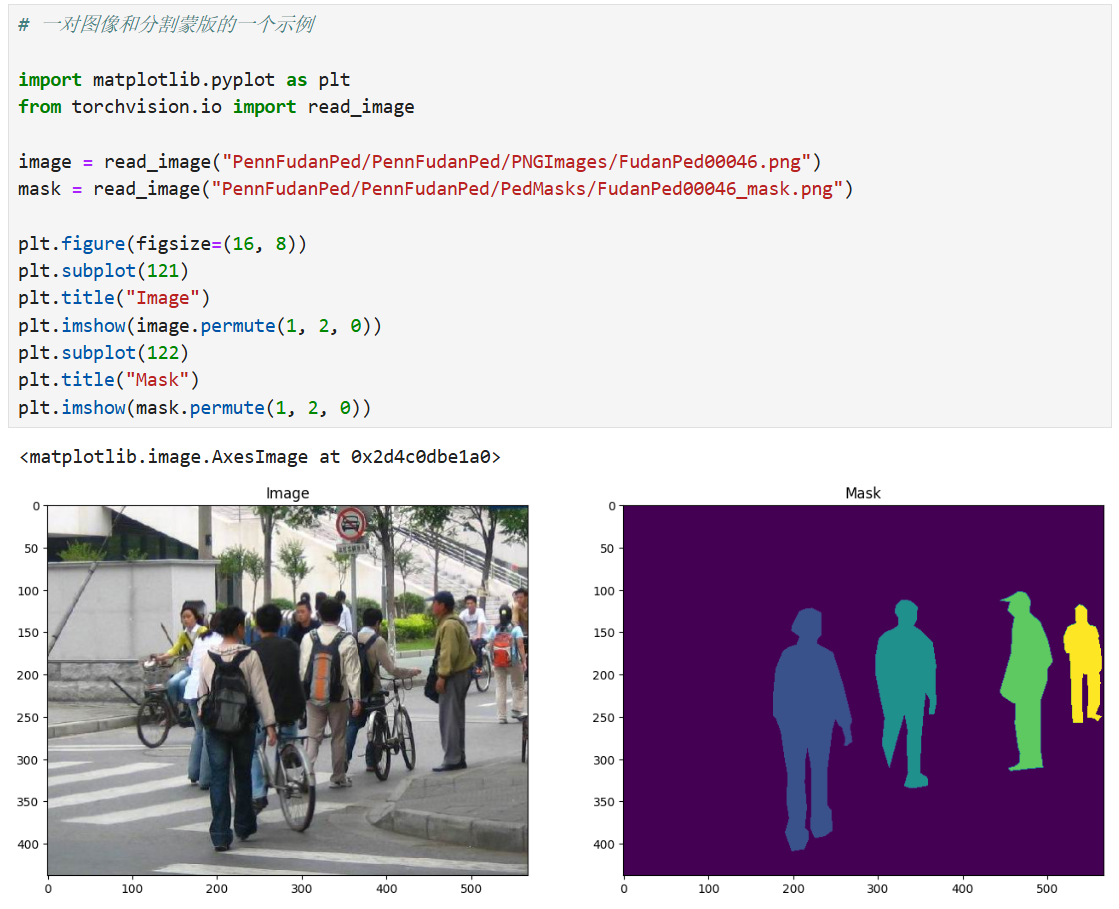


图2 图像和分割蒙版示例

因此，每幅图像都有一个相应的分割掩码（Segmentation Mask），其中每种颜色对应一个不同的实例。让我们为这个数据集编写一个 torch.utils.data.Dataset 类。

1、图像张量将由 torchvision.tv\_tensors.Image 封装；

2、边界框将由 torchvision.tv\_tensors.BoundingBoxes 封装；

3、遮罩将由 torchvision.tv\_tensors.Mask 封装。

在本教程中，我们将使用 [Mask R-CNN](https://arxiv.org/abs/1703.06870)，它基于 [Faster R-CNN](https://arxiv.org/abs/1506.01497) 之上。

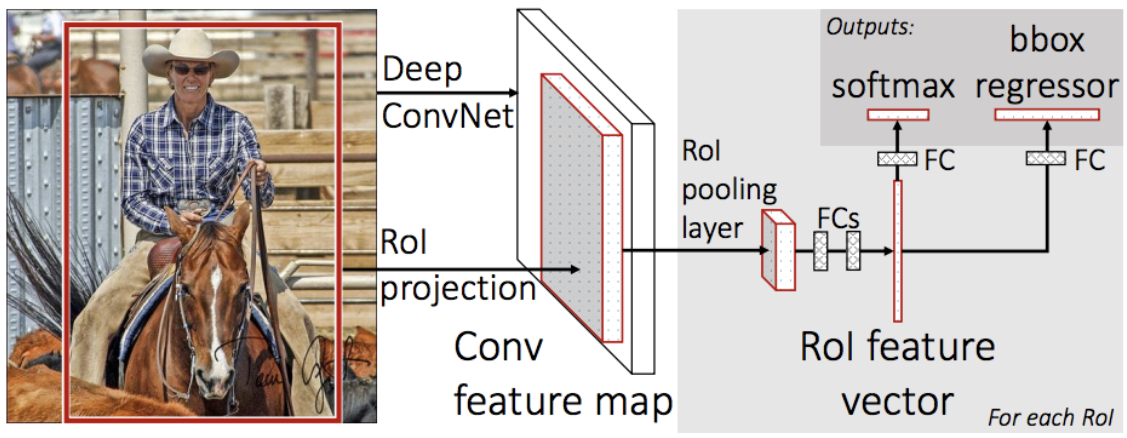


图3 Faster R-CNN示例

Mask R-CNN 添加了一个额外的分支即全卷积网络和掩码预测，并将 Faster R-CNN中的兴趣区域池化层替换为了兴趣区域对齐层，使用双线性插值来保留特征图上的空间信息，从而更适用于像素级预测。因此如果训练集中标注了每个目标在图像上的像素级位置，那么Mask R-CNN能够有效地利用这些详尽的标注信息进一步提升目标检测的精度。

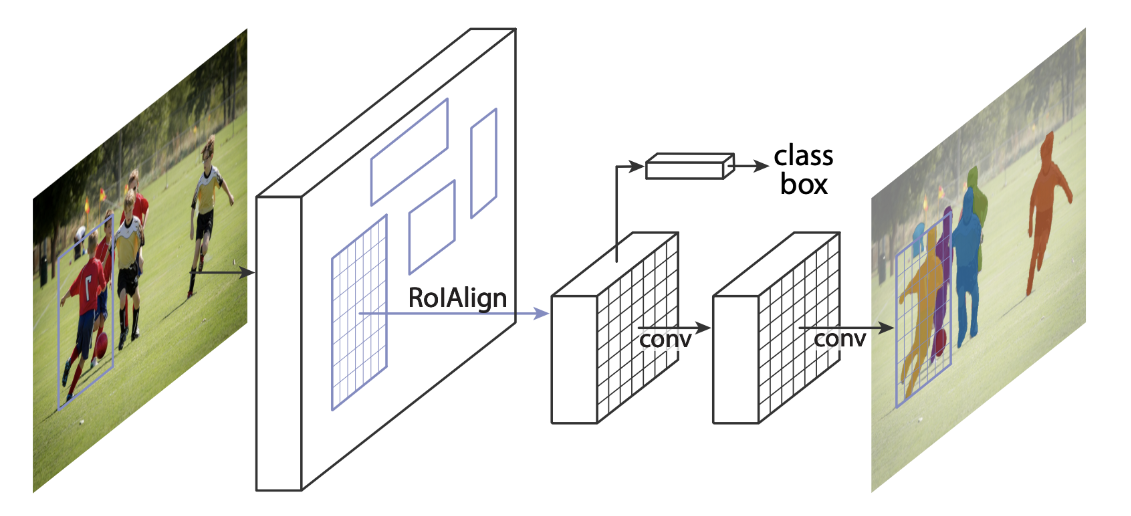


图4 Mask R-CNN示例

有两种常见的情况当人们可能想要修改 TorchVision Model Zoo 中的可用模型之一。第一种是当我们想要从预先训练的模型开始，然后微调最后一层。另一种是当我们想替换模型主干为其他模型（例如：为了更快地进行预测）。

假设您想从COCO（目标检测上的经典数据集，类似于图像分类中的ImageNet）上预训练的模型开始并希望针对您的特定类对其进行微调。这是一个可能的方法：

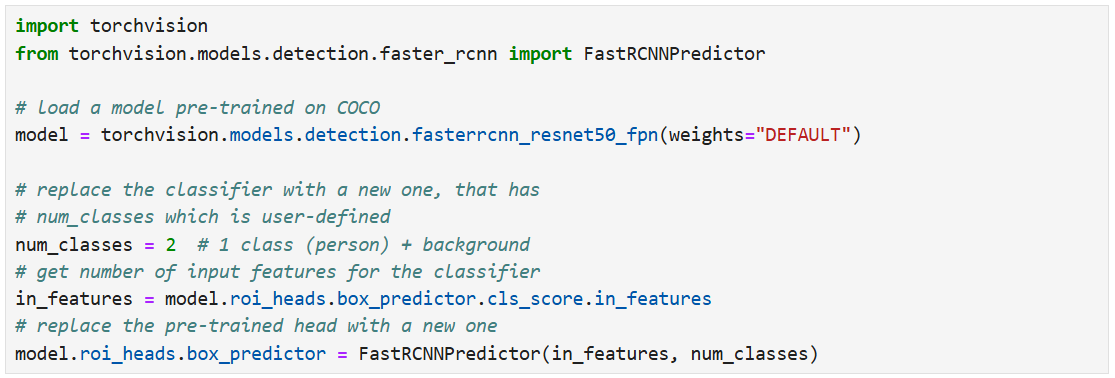


图5 使用COCO预训练模型

修改模型以添加不同的主干（部分修改代码）。

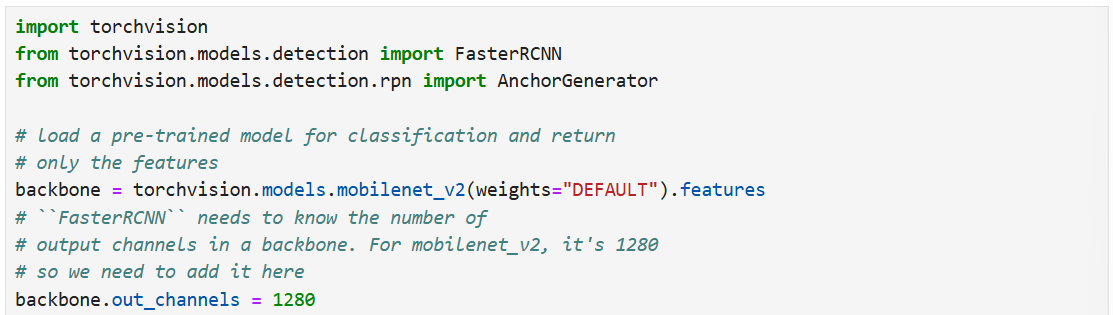


图6 添加不同的主干

然后我们进行 PennFudan 数据集的目标检测和实例分割模型，在我们的例子中，我们希望从预训练模型进行微调，因为我们的数据集非常小，因此我们将遵循第1种方法。

在这里，我们还想计算实例分段掩码，因此我们将使用Mask R-CNN：



图7 使用Mask R-CNN

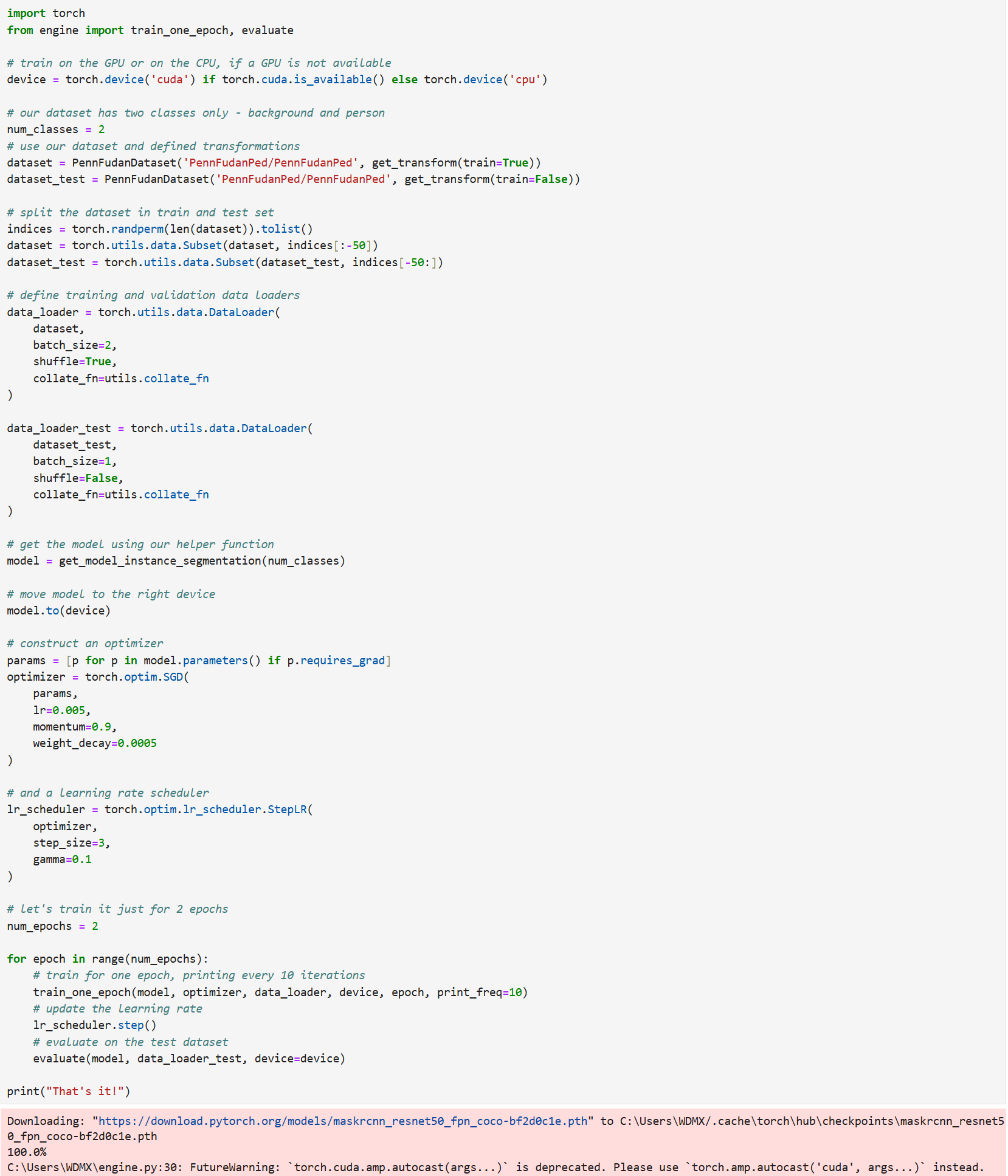
在 references/detection/ 中，我们有许多辅助函数来简化检测模型的训练和评估。在这里，我们将使用 references/detection/engine.py 和 references/detection/utils.py 。只需将下的所有内容下载到您的文件夹中并在此处使用它们（我个人觉得使用Python的request库更具可移植性和跨平台性，因此将代码进行了适当修改）。

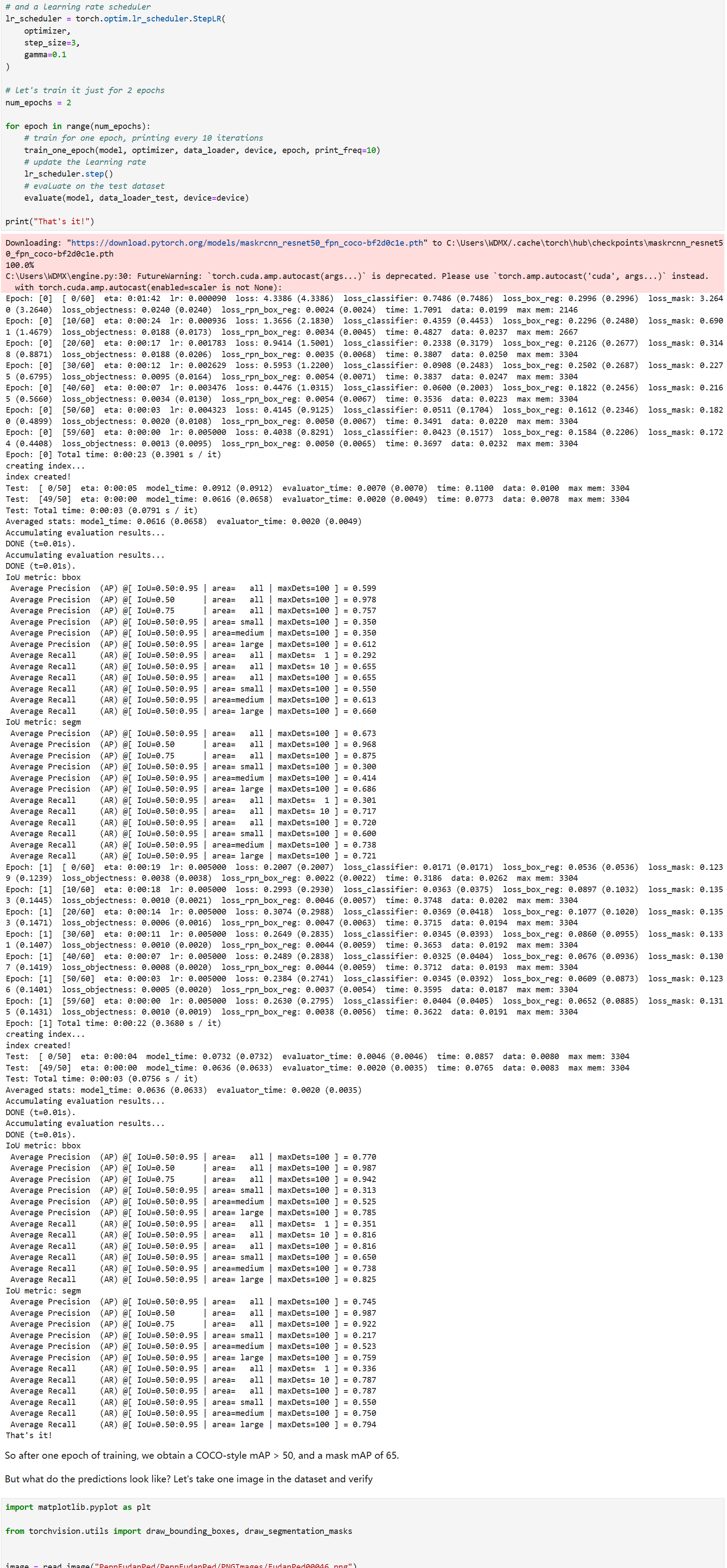


图8 下载并写入辅助函数

从 v0.15.0 开始，torchvision 提供了[新的 Transforms API](https://pytorch.org/vision/stable/transforms.html)，可以轻松地为对象检测和分割任务编写数据增强管道。我们编写一些用于数据增强的辅助函/转型（get\_transfrom）。

现在让我们编写执行训练的 main 函数和验证，并通过输出来展示模型训练效果：





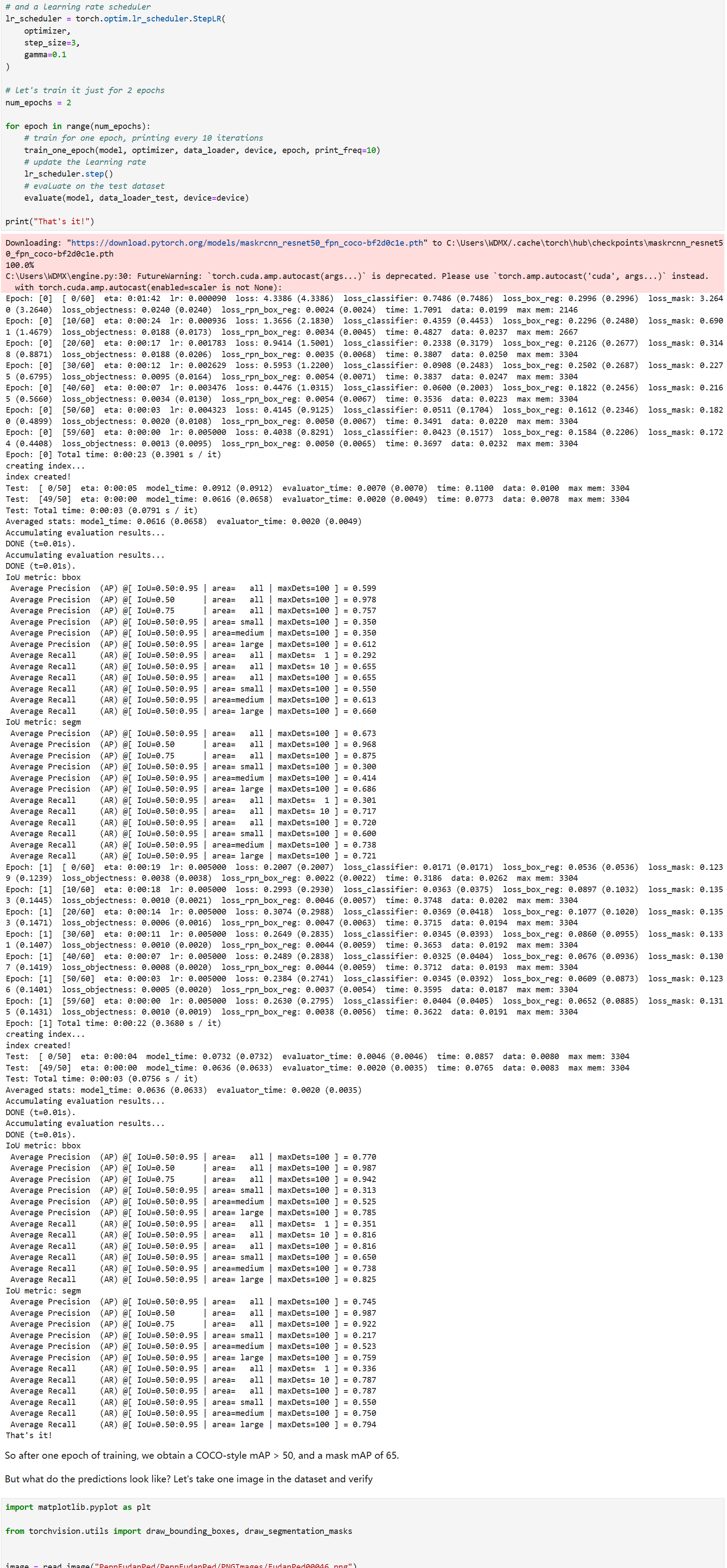


图9 模型训练

因此，经过一个 epoch 的训练，我们获得了 COCO 风格的 mAP > 50，并且 Mask mAP 为 65。但预测是什么样的呢？让我们在应用到数据集并验证，使用 matplotlib.pyplot 可视化输出以便对验证结果有更好的理解。



图10 验证与结果可视化