

8、使用 DetectNet 定位对象坐标

先前的图像识别示例输出表示整个输入图像的类概率。我们在本教程中强调的第二个深度学习功能是检测对象，并找到视频中这些对象所在的位置（即提取其边界框）。这是使用'detectNet' - 或对象检测/本地化网络执行的。

该 `detectNet` 对象接受 2D 图像作为输入，并输出检测到的边界框的坐标列表。为了训练对象检测模型，首先使用预训练的 ImageNet 识别模型（如 GoogLeNet）以及除了源图像之外的训练数据集中包括的边界坐标标签。

本教程包含以下预训练的 DetectNet 模型：

1. **ped-100** (single-class pedestrian detector)
2. **multiped-500** (multi-class pedestrian + baggage detector)
3. **facenet-120** (single-class facial recognition detector)
4. **coco-airplane** (MS COCO airplane class)
5. **coco-bottle** (MS COCO bottle class)
6. **coco-chair** (MS COCO chair class)
7. **coco-dog** (MS COCO dog class)

与前面的示例一样，提供了用于使用 `detectNet` 的控制台程序和相机程序。

从命令行检测对象

该 `detectnet-console` 程序可用于查找图像中的对象。要加载 repo 附带的预训练对象检测模型之一，可以将预训练模型名称指定为第 3 个参数 `detectnet-console`：

```
$ ./detectnet-console dog_1.jpg output_2.jpg coco-dog
```

上面的命令将处理 `dog_1.jpg`，使用预训练的 DetectNet-COCO-Dog 模型将其保存到 `output_1.jpg`。这是各种各样的快捷方式，因此如果您不愿意，您不需要自己训练模型。



上述过程如果失败的话，如果报错是 model file
networks/SSD-Mobilenet-v2/ssd_mobilenet_v2_coco.uff' was not found.

这个可以应该没有上面模型，到网盘的其他模型文件夹把这个模型像添加到下面
network 路径下并解压，方法是跟前面安装 tensorRT 一样的，需要重新编译下 cmake 和
make。

```

nx@nx-desktop:~/jetson-inference/data/networks$ ls
alexnet_noprob.prototxt      facenet-120                      FCN-Alexnet-SYNTHIA-Summer-SD.tar.gz
alexnet.prototxt            facenet-120.tar.gz              GoogleNet-ILSVRC12-subset
bvlc_alexnet.caffemodel     FCN-Alexnet-Aerial-FPV-4ch-720p GoogleNet-ILSVRC12-subset.tar.gz
bvlc_alexnet.caffemodel.1.1.7103.GPU.FP16.engine FCN-Alexnet-Aerial-FPV-4ch-720p.tar.gz googlenet_noprob.prototxt
bvlc_googlenet.caffemodel   FCN-Alexnet-Aerial-FPV-720p    googlenet.prototxt
bvlc_googlenet.caffemodel.1.1.7103.GPU.FP16.engine FCN-Alexnet-Aerial-FPV-720p.tar.gz ilsvrc12_synset_words.txt
Deep-Homography-COCO       FCN-Alexnet-Cityscapes-HD      multiped-500
DetectNet-COCO-Airplane     FCN-Alexnet-Cityscapes-HD.tar.gz multiped-500.tar.gz
DetectNet-COCO-Airplane.tar.gz FCN-Alexnet-Cityscapes-SD      ped-100
DetectNet-COCO-Bottle       FCN-Alexnet-Cityscapes-SD.tar.gz ped-100.tar.gz
DetectNet-COCO-Bottle.tar.gz FCN-Alexnet-Pascal-VOC         ssd_coco_labels.txt
DetectNet-COCO-Chair        FCN-Alexnet-Pascal-VOC.tar.gz SSD-Mobilenet-v2
DetectNet-COCO-Chair.tar.gz FCN-Alexnet-SYNTHIA-CVPR16     SSD-Mobilenet-v2.tar.gz
DetectNet-COCO-Dog          FCN-Alexnet-SYNTHIA-CVPR16.tar.gz Super-Resolution-BSD500
DetectNet-COCO-Dog.tar.gz   FCN-Alexnet-SYNTHIA-Summer-HD Super-Resolution-BSD500.tar.gz
detectnet.prototxt         FCN-Alexnet-SYNTHIA-Summer-SD

```

提供预训练的 DetectNet 模型

下面是使用 repo 下载的预训练的 DetectNet 快照表（位于 data/networks 运行 cmake 步骤后的目录中）以及 detectnet-console 用于加载预训练模型的相关参数：

DIGITS model	CLI argument	classes
DetectNet-COCO-Airplane	coco-airplane	airplanes

DIGITS model	CLI argument	classes
DetectNet-COCO-Bottle	coco-bottle	bottles
DetectNet-COCO-Chair	coco-chair	chairs
DetectNet-COCO-Dog	coco-dog	dogs
ped-100	pednet	pedestrians
multiped-500	multiped	pedestrians, luggage
facenet-120	facenet	faces

这些都已经应用了上面的 python 层补丁。

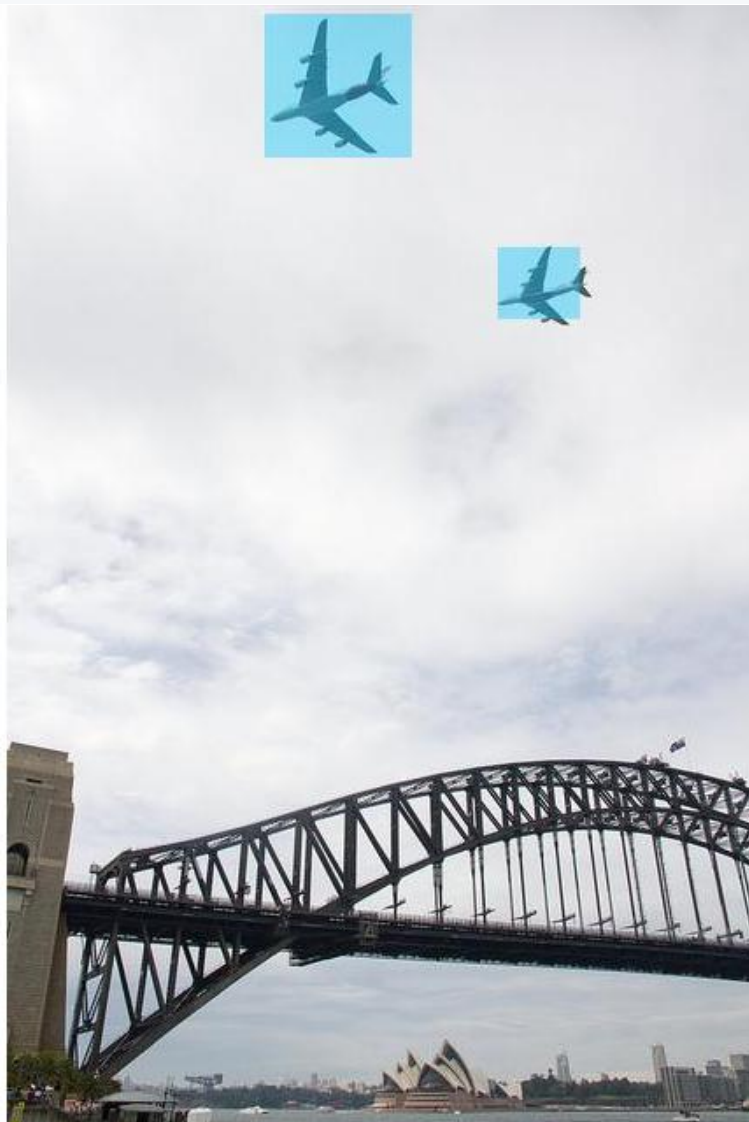
在 Jetson 上运行其他 MS-COCO 模型

让我们尝试运行一些其他 COCO 模型。这些培训数据都包含在上面下载的数据集中。虽然上面的 DIGITS 训练示例是针对 coco-dog 模型的，但是可以遵循相同的程序来训练 DetectNet 对样本 COCO 数据集中包含的其他类。

```
$ ./detectnet-console bottle_0.jpg output_3.jpg coco-bottle
```



```
$ ./detectnet-console airplane_0.jpg output_3.jpg coco-airplane
```



在 Jetson 上运行行人模型

回购中还包括预训练的 DetectNet 模型以检测人类。这些 `pednet` 和 `multiped` 模型识别行人，同时 `facenet` 识别面孔（来自 [Fddb](#)）。以下是在拥挤的空间中同时检测多个人的示例：

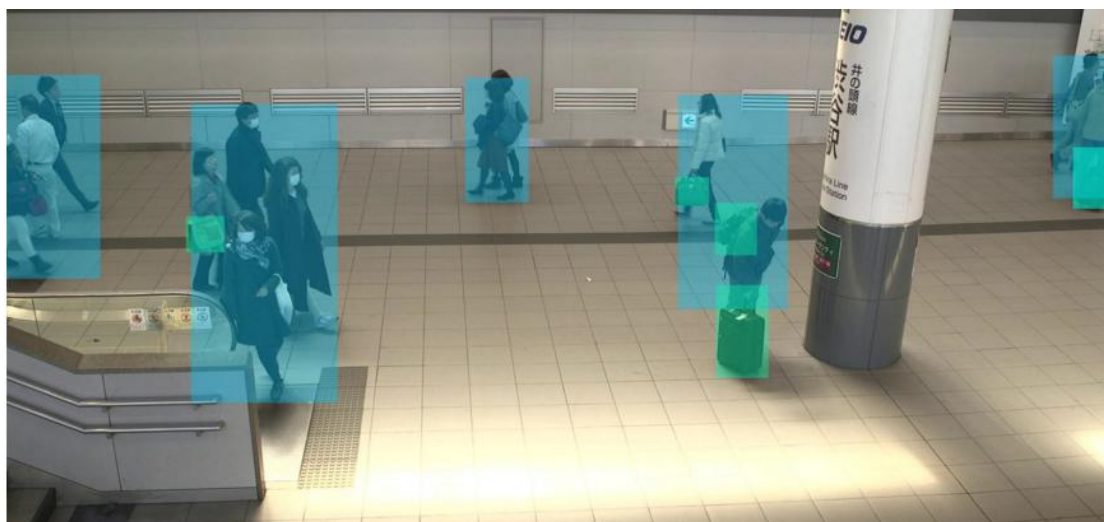
```
$ ./detectnet-console peds-004.jpg output-4.jpg multiped
```




多类目标检测模型

使用多重模型（PEDNET_MULTI）时，对于包含行李或行李以及行人的图像，第二个对象类将使用绿色叠加层渲染。

```
$ ./detectnet-console peds-003.jpg output-3.jpg multiped
```



接下来教程，我们将在实时摄像机流上运行对象检测。