

---

## 8.Using DetectNet to locate object coordinates

---

We mainly learn how to find the location of the target object in the video (extract its bounding box).

The [detectNet](#) object accepts the 2D image as input and outputs a list of coordinates of the detected bounding box.

First,we need to use a pre-trained ImageNet recognition model (such as Googlenet).

This tutorial contains the following pre-trained DetectNet models:

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)**

### Detect objects from the command line

The [detectnet-console](#) program can be used to find objects in an image. To load one of the pre-trained object detection models attached to the repo, you can specify the pre-trained model name as the third parameter, detectnet-console:

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

The above command will process dog\_1.jpg and save it to output\_1.jpg using the pre-trained DetectNet-COCO-Dog model.



[在此处键入]

!Tips:

If the above process fails and an error message appears in the system, "model file networks/SSD-Mobilenet-v2/ssd\_mobilenet\_v2\_coco.uff" was not found."

The reason for this error is that the corresponding model is missing. We have provided this model, and then add it to the network path and decompress it. The method is the same as the previous installation of tensorRT. You need to recompile cmake and 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           FCN-Alexnet-Aerial-FPV-4ch-720p
bvlc_alexnet.caffemodel      FCN-Alexnet-Aerial-FPV-4ch-720p.tar.gz
bvlc_alexnet.caffemodel.1.1.7103.GPU.FP16.engine  FCN-Alexnet-Aerial-FPV-720p
bvlc_googlenet.caffemodel     FCN-Alexnet-Aerial-FPV-720p.tar.gz
bvlc_googlenet.caffemodel.1.1.7103.GPU.FP16.engine  FCN-Alexnet-Cityscapes-HD
Deep-Homography-COCO         FCN-Alexnet-Cityscapes-HD.tar.gz
Deep-Homography-COCO.tar.gz  FCN-Alexnet-Cityscapes-SD
DetectNet-COCO-Airplane       FCN-Alexnet-Cityscapes-SD.tar.gz
DetectNet-COCO-Airplane.tar.gz FCN-Alexnet-Pascal-VOC
DetectNet-COCO-Bottle         FCN-Alexnet-Pascal-VOC.tar.gz
DetectNet-COCO-Bottle.tar.gz  FCN-Alexnet-SYNTHIA-CVPR16
DetectNet-COCO-Chair          FCN-Alexnet-SYNTHIA-CVPR16.tar.gz
DetectNet-COCO-Chair.tar.gz   FCN-Alexnet-SYNTHIA-Summer-HD
DetectNet-COCO-Dog            FCN-Alexnet-SYNTHIA-Summer-HD.tar.gz
DetectNet-COCO-Dog.tar.gz     FCN-Alexnet-SYNTHIA-Summer-SD
detectnet.prototxt            GoogleNet-ILSVRC12-subset
                             GoogleNet-ILSVRC12-subset.tar.gz
                             googlenet_noprob.prototxt
                             googlenet.prototxt
                             ilsvrc12_synset_words.txt
                             multyped-500
                             multyped-500.tar.gz
                             ped-100
                             ped-100.tar.gz
                             SSD-Mobilenet-v2
                             SSD-Mobilenet-v2.tar.gz
                             Super-Resolution-BSD500
                             Super-Resolution-BSD500.tar.gz
```

### Providing pre-trained DetectNet model

The following is a pre-trained DetectNet snapshot table downloaded by repo (It is located in the directory after we run **cmake** in the data/networks directory.) and the relevant parameters of the detectnet-console for loading the pre-trained model:

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

These have already applied the above python layer patch.

### Running other MS-COCO models on Jetson

We can try to run some other COCO models. These training data are included in the data set downloaded above.

You can input the following command:

[在此处键入]

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



You can input the following command:

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



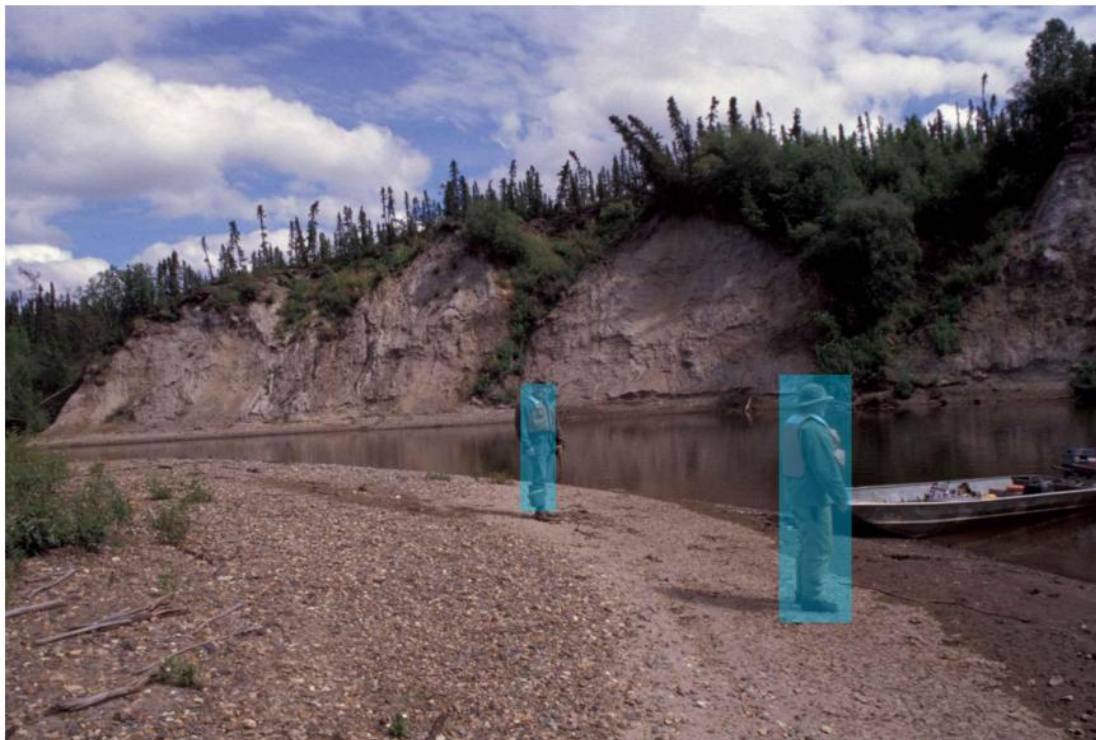
Running the pedestrian model on Jetson



[在此处键入]

A pre-trained **DetectNet** model is be used to detect humans. These **pednet** and **multiped** models recognize pedestrians while **facenet** recognizes faces (from [Fddb](#)). Here's an example of detecting multiple people simultaneously in a crowded space: You can input the following command:

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



### Multi-class target detection model

We need to input the following command:

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

When using the multi-model (**PEDNET\_MULTI**), the second object class will be rendered with a green overlay. (For example: Some images containing baggage and pedestrians.)

As shown blew.

