8.Using DetectNet to locate object coordinates

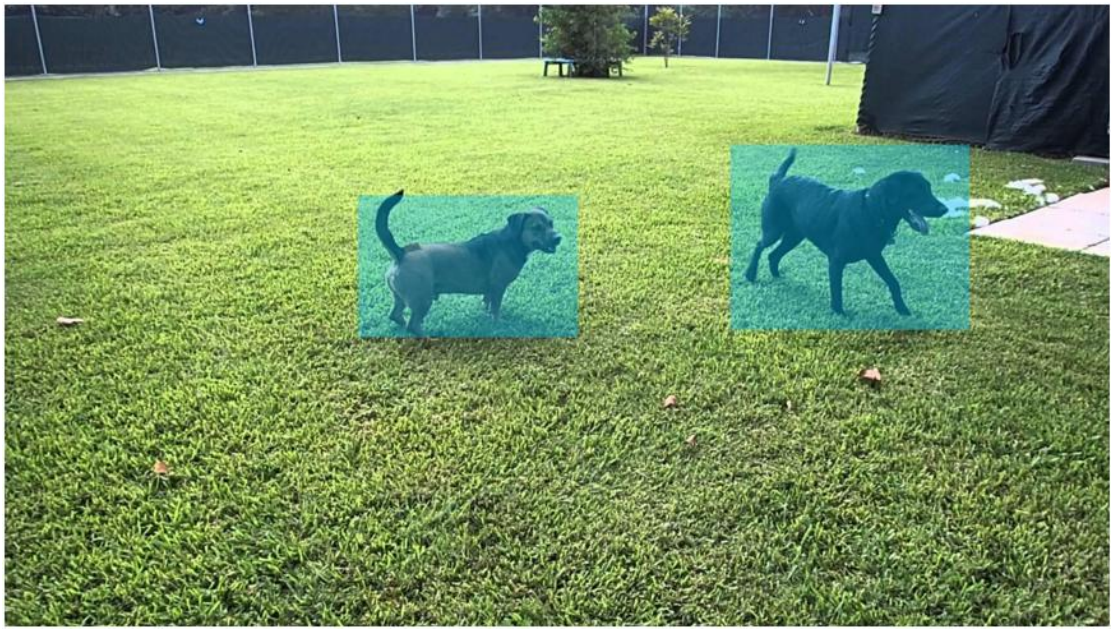
We mainly learn how to find the location of the target object in the video (extract its bounding box).
The detectNet object accepts the 2D image as input and outputs a list of coordinates of the detected bounding box.
First,we need to use a pre-trained ImageNet recognition model (such as Googlenet).

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.



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
multiped-500	multiped	pedestrians, luggage
facenet-120	facenet	faces

These have already applied the above python layer patch.

Running other MS-COCO models on Jetson

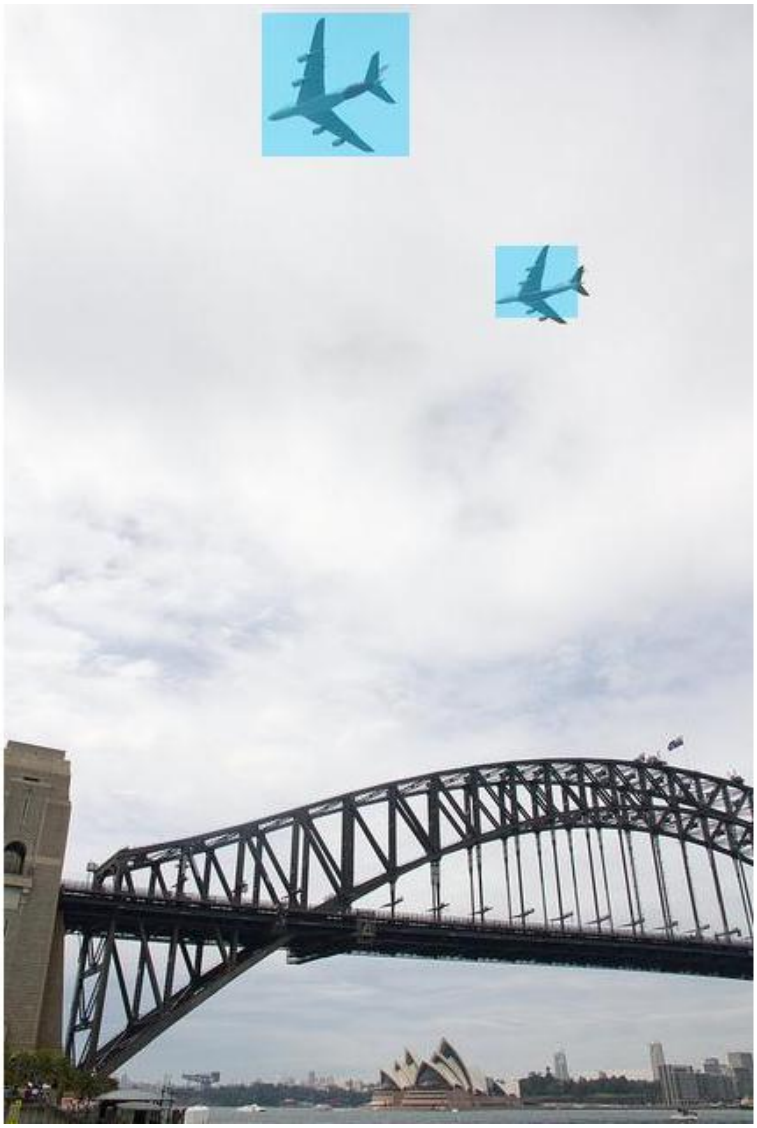
We can try to run some other COCO models. These training data are included in the data set downloaded above.

You can input the following command:

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

You can input the following command:

```
$ ./detectnet-console ./images/airplane_0.jpg output_4.jpg --network=coco-airplane
```



Running the pedestrian model on Jetson

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 ./images/peds_0.jpg output_5.jpg --network=multiped
```

Multi-class target detection model

We need to input the following command:

```
$ ./detectnet-console ./images/peds_3.jpg output_6.jpg --network=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.