RETINANET FOR CUSTOM OBJECT DETECTION

1)Initially for object detection we need to train the retinanet and trained model is saved in a .h5 file.Then this .h5 file is used for testing.

How to train:
2) For training on custom images we need a collection of images and their labels.

Images:
i)Collection of images on which you want to train the retinanet.

Labels:
i) We need to label the objects in the images using tool named “label-me”
ii) Using label me tool we draw a bounding box across the objects in image we want to train
iii)The label me tool gives us output in the form of .xml

3) So Now we will have both images and their respective .xml files with us. we just give the directory name where this folder is stored as Input to code.

4)The code itself separates xml files from images.These xml files are converted into two csv files.
i)classes.csv
ii)annotations.csv

5)The other Input we give to the code is weights file.we use pretrained weights for training to get good results.The Input file we use is “resnet50_coco_best_v2.0.1.h5”.

6)The other changes we can make in this code is number of epochs we want to run,steps per epoch,batch size.

Inputs - images and .xml label files, pretrained weights, epochs,batch size,steps per epoch.

7)After training we store the model in a .h5 file.we can use this .h5 file when testing

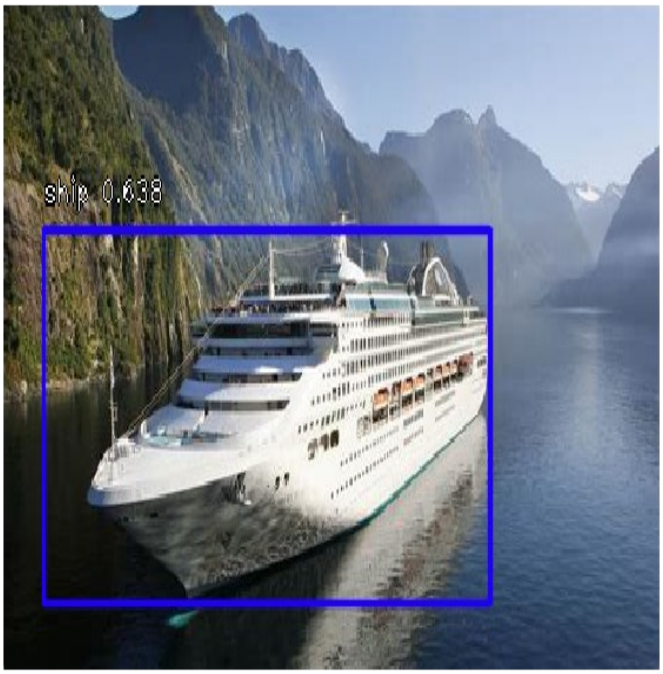
Testing retinanet:

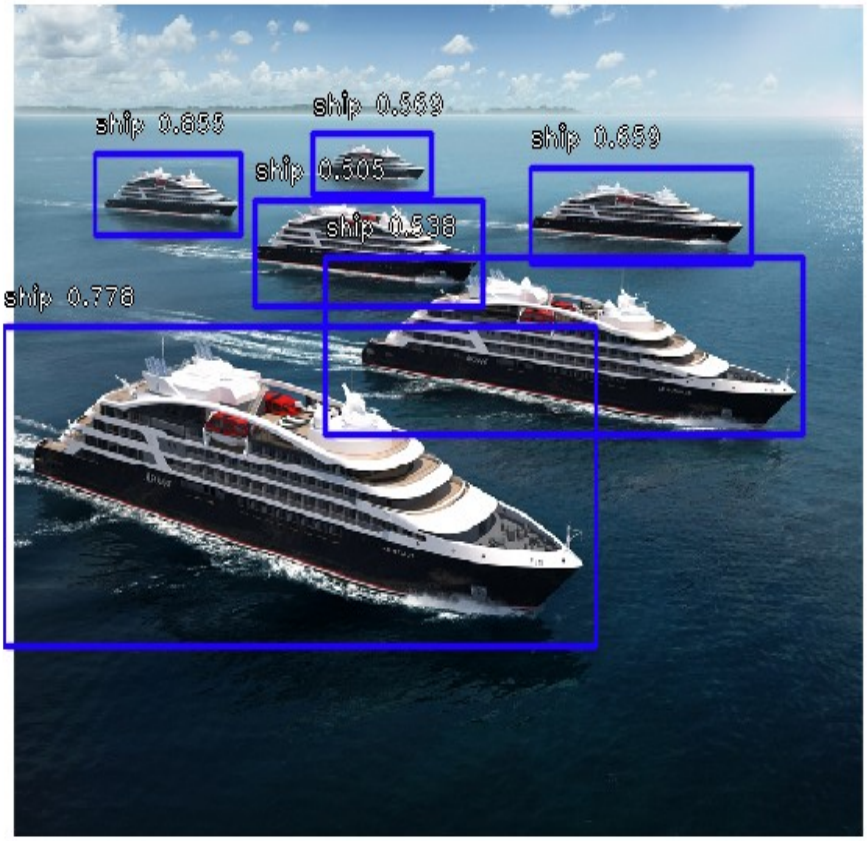
Some exampe Input Images:

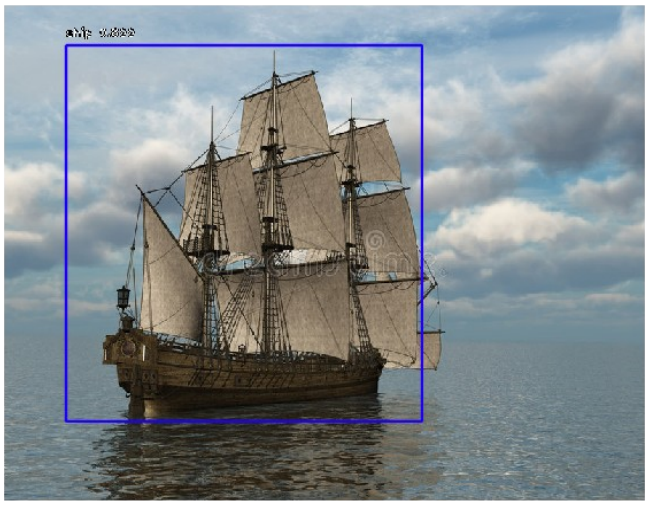
8)The .h5 file we stored when training can now be used for testing.

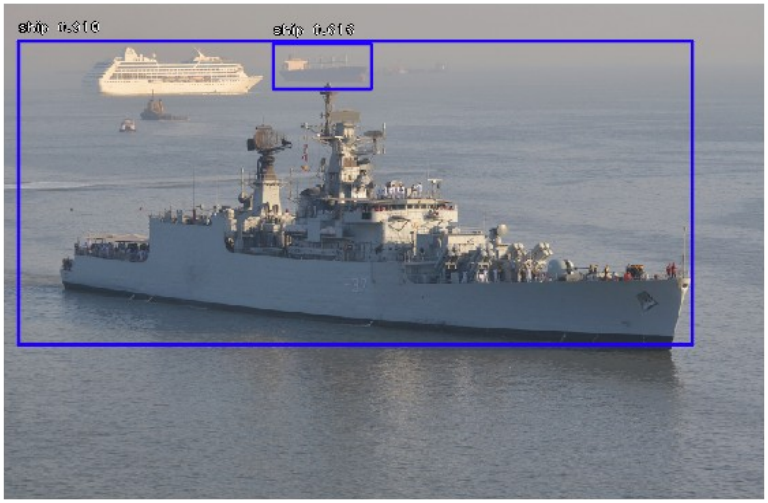
9) we can give the file location of this .h5 file as input and test the detection for the test image.

10)output images :





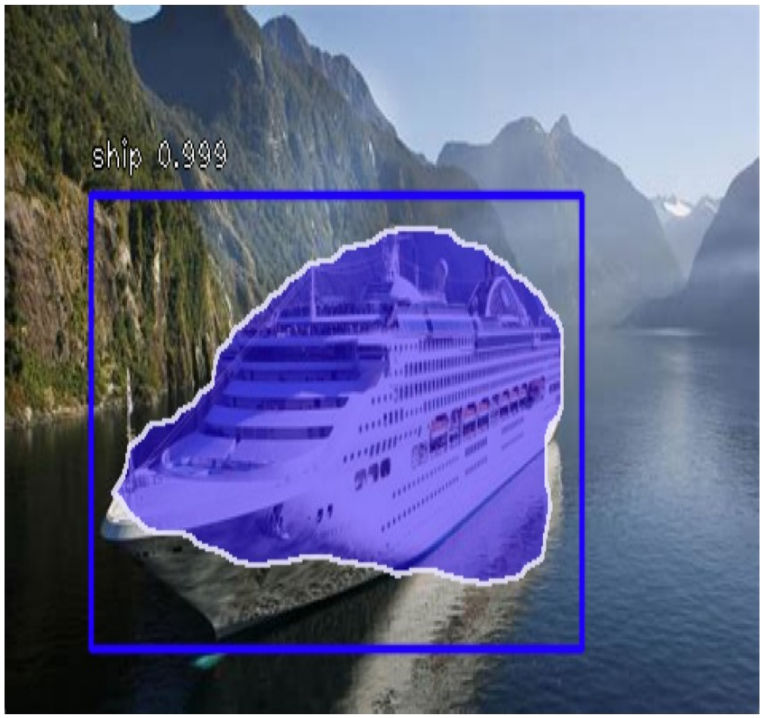


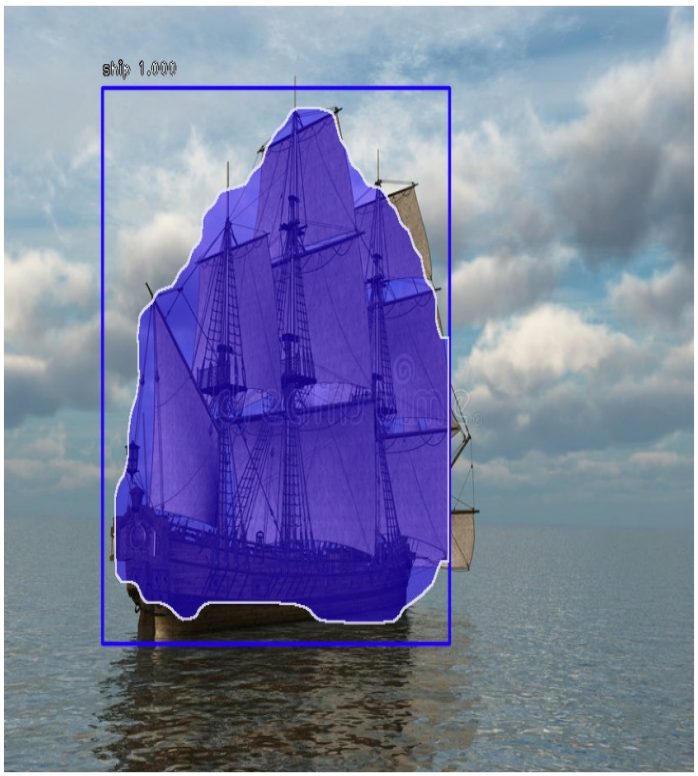


11) The output has bounding box for the object detected with its label and probability percentage.

We can even mask the output using Retinamask code.

example image :





12)The above image is a result of Retinamask and it has outputs like bounding box, label , mask,probability.