yolov5实现船舶目标检测

一、实现

采用yolov5-5.0版本，模型选择yolov5s，训练300epochs。

选择Seaships的7000张照片作为数据集。

其中也可以看到小目标也能得到较好的检测：

二、更改日志

1、文件配置

1、引入文件夹：

新建make_voc_dir.py文件

```
import os
os.makedirs('VOCdevkit/VOC/Annotations')
os.makedirs('VOCdevkit/VOC/JPEGImages')
```

首先选取了六十张图片和该六十张照片的xml文件作为测试放入了相对应的文件夹中。

2、voc到yolo格式转换

创建voc_to_yolo.py

```
"""
# VOC数据集 转 YOLO数据集 格式
"""


import xml.etree.ElementTree as ET
import pickle
import os
from os import listdir, getcwd
from os.path import join
import random
from shutil import copyfile


# 定义六个类型
classes=["ore carrier","passenger ship","container ship","bulk cargo
carrier","general cargo ship","fishing boat"]

# classes=["ball"]
```

```python
# 划分比率
TRAIN_RATIO = 75 # 也可以选择其他


def clear_hidden_files(path):
    dir_list = os.listdir(path)
    for i in dir_list:
        abspath = os.path.join(os.path.abspath(path), i)
        if os.path.isfile(abspath):
            if i.startswith("._"):
                os.remove(abspath)
        else:
            clear_hidden_files(abspath)


# size是原图的宽和高
def convert(size, box):
    dw = 1. / size[0]
    dh = 1. / size[1]
    x = (box[0] + box[1]) / 2.0
    y = (box[2] + box[3]) / 2.0
    w = box[1] - box[0]
    h = box[3] - box[2]
    x = x * dw
    w = w * dw
    y = y * dh
    h = h * dh
    return (x, y, w, h)


def convert_annotation(image_id):
    in_file = open('VOCdevkit/VOC/Annotations/%s.xml' % image_id, 'rb')
    out_file = open('VOCdevkit/VOC/YOLOLabels/%s.txt' % image_id, 'w')
    tree = ET.parse(in_file)
    root = tree.getroot()
    size = root.find('size')
    w = int(size.find('width').text)
    h = int(size.find('height').text)

    for obj in root.iter('object'):
        difficult_elem = obj.find('difficult')
        if difficult_elem is not None:
            difficult = difficult_elem.text
            # 可能还需要将字符串转换为整数或其他数据类型
            if difficult is not None:
                difficult = int(difficult)
        else:
            # 如果没有找到 'difficult' 元素，设置一个默认值
            difficult = 0  # 或者 '0'，根据你的需要
        cls = obj.find('name').text
        if cls not in classes or int(difficult) == 1:
            continue
        cls_id = classes.index(cls)
        xmlbox = obj.find('bndbox')
```

```
        b = (float(xmlbox.find('xmin').text), float(xmlbox.find('xmax').text),
float(xmlbox.find('ymin').text),
             float(xmlbox.find('ymax').text))
        bb = convert((w, h), b)
        out_file.write(str(cls_id) + " " + " ".join([str(a) for a in bb]) + '\n')
    in_file.close()
    out_file.close()


wd = os.getcwd()
data_base_dir = os.path.join(wd, "VOCdevkit/")
if not os.path.isdir(data_base_dir):
    os.mkdir(data_base_dir)
work_sapce_dir = os.path.join(data_base_dir, "VOC/")
if not os.path.isdir(work_sapce_dir):
    os.mkdir(work_sapce_dir)
annotation_dir = os.path.join(work_sapce_dir, "Annotations/")
if not os.path.isdir(annotation_dir):
    os.mkdir(annotation_dir)
clear_hidden_files(annotation_dir)
image_dir = os.path.join(work_sapce_dir, "JPEGImages/")
if not os.path.isdir(image_dir):
    os.mkdir(image_dir)
clear_hidden_files(image_dir)

# 这个部分可以不要
yolo_labels_dir = os.path.join(work_sapce_dir, "YOLOLabels/")
if not os.path.isdir(yolo_labels_dir):
    os.mkdir(yolo_labels_dir)
clear_hidden_files(yolo_labels_dir)

yolov5_images_dir = os.path.join(data_base_dir, "images/")
if not os.path.isdir(yolov5_images_dir):
    os.mkdir(yolov5_images_dir)
clear_hidden_files(yolov5_images_dir)
yolov5_labels_dir = os.path.join(data_base_dir, "labels/")
if not os.path.isdir(yolov5_labels_dir):
    os.mkdir(yolov5_labels_dir)
clear_hidden_files(yolov5_labels_dir)
yolov5_images_train_dir = os.path.join(yolov5_images_dir, "train/")
if not os.path.isdir(yolov5_images_train_dir):
    os.mkdir(yolov5_images_train_dir)
clear_hidden_files(yolov5_images_train_dir)
yolov5_images_test_dir = os.path.join(yolov5_images_dir, "val/")
if not os.path.isdir(yolov5_images_test_dir):
    os.mkdir(yolov5_images_test_dir)
clear_hidden_files(yolov5_images_test_dir)
yolov5_labels_train_dir = os.path.join(yolov5_labels_dir, "train/")
if not os.path.isdir(yolov5_labels_train_dir):
    os.mkdir(yolov5_labels_train_dir)
clear_hidden_files(yolov5_labels_train_dir)
yolov5_labels_test_dir = os.path.join(yolov5_labels_dir, "val/")
if not os.path.isdir(yolov5_labels_test_dir):
    os.mkdir(yolov5_labels_test_dir)
clear_hidden_files(yolov5_labels_test_dir)
```

```
# 这两个部分yolov5_train.txt, yolov5_val.txt可以不要
train_file = open(os.path.join(wd, "yolov5_train.txt"), 'w')
test_file = open(os.path.join(wd, "yolov5_val.txt"), 'w')
train_file.close()
test_file.close()
train_file = open(os.path.join(wd, "yolov5_train.txt"), 'a')
test_file = open(os.path.join(wd, "yolov5_val.txt"), 'a')
list_imgs = os.listdir(image_dir)  # list image_one files
prob = random.randint(1, 100)
print("Probability: %d" % prob)
for i in range(0, len(list_imgs)):
    path = os.path.join(image_dir, list_imgs[i])
    if os.path.isfile(path):
        image_path = image_dir + list_imgs[i]
        voc_path = list_imgs[i]
        (nameWithoutExtention, extention) =
os.path.splitext(os.path.basename(image_path))
        (voc_nameWithoutExtention, voc_extention) =
os.path.splitext(os.path.basename(voc_path))
        annotation_name = nameWithoutExtention + '.xml'
        annotation_path = os.path.join(annotation_dir, annotation_name)
        label_name = nameWithoutExtention + '.txt'
        label_path = os.path.join(yolo_labels_dir, label_name)
    prob = random.randint(1, 100)
    print("Probability: %d" % prob)
    if (prob < TRAIN_RATIO):  # train dataset
        if os.path.exists(annotation_path):
            train_file.write(image_path + '\n')
            convert_annotation(nameWithoutExtention)  # convert label
            copyfile(image_path, yolov5_images_train_dir + voc_path)
            copyfile(label_path, yolov5_labels_train_dir + label_name)
    else:  # test dataset
        if os.path.exists(annotation_path):
            test_file.write(image_path + '\n')
            convert_annotation(nameWithoutExtention)  # convert label
            copyfile(image_path, yolov5_images_test_dir + voc_path)
            copyfile(label_path, yolov5_labels_test_dir + label_name)
train_file.close()
test_file.close()
```

可以看到在VOCdevkit文件夹中，有一些更改。

3、训练文件改动

在train.py中修改如下：

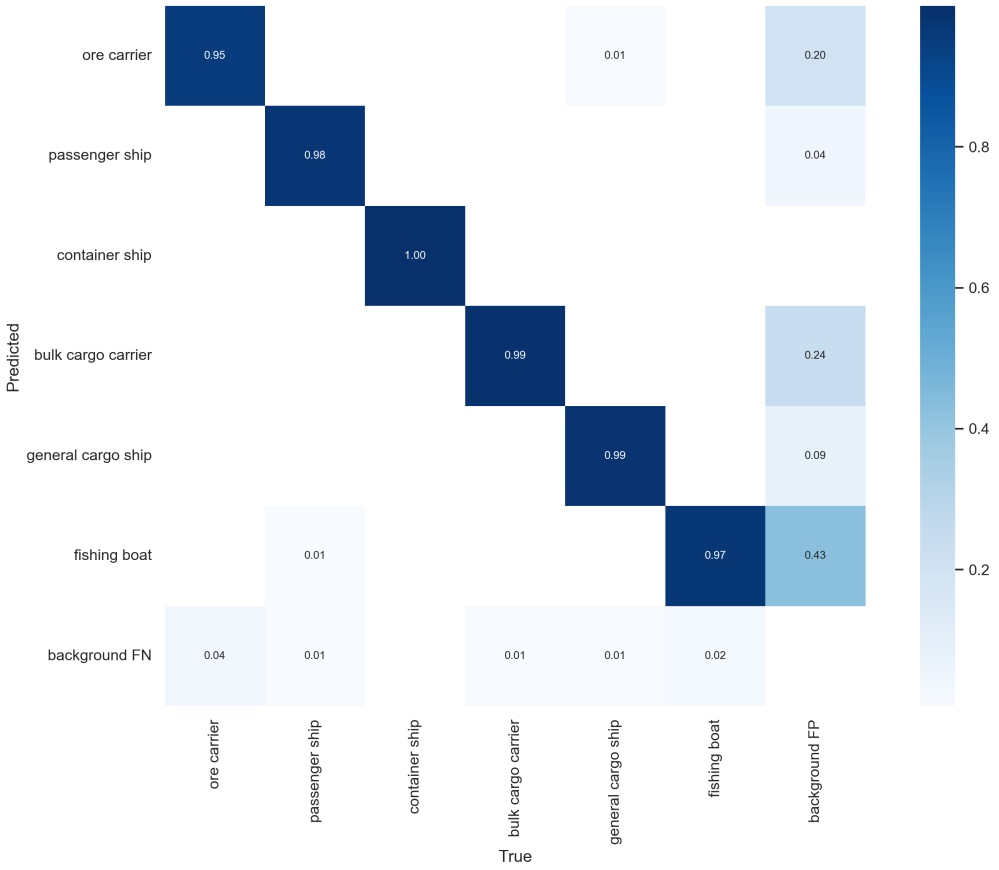
运用yolov5s的模型（在github官网需要下载）

```
parser = argparse.ArgumentParser()
parser.add_argument('--weights', type=str, default='weights/yolov5s.pt',
help='initial weights path')
parser.add_argument('--cfg', type=str, default='models/yolov5s.yaml',
help='model.yaml path')
parser.add_argument('--data', type=str, default='data/voc.yaml', help='data.yaml
path')
parser.add_argument('--hyp', type=str, default='data/hyp.scratch.yaml',
help='hyperparameters path')
parser.add_argument('--epochs', type=int, default=300)
parser.add_argument('--batch-size', type=int, default=16, help='total batch size
for all GPUs')
parser.add_argument('--img-size', nargs='+', type=int, default=[640, 640],
help='[train, test] image sizes')
parser.add_argument('--rect', action='store_true', help='rectangular training')
parser.add_argument('--resume', nargs='?', const=True, default=True, help='resume
most recent training')
```

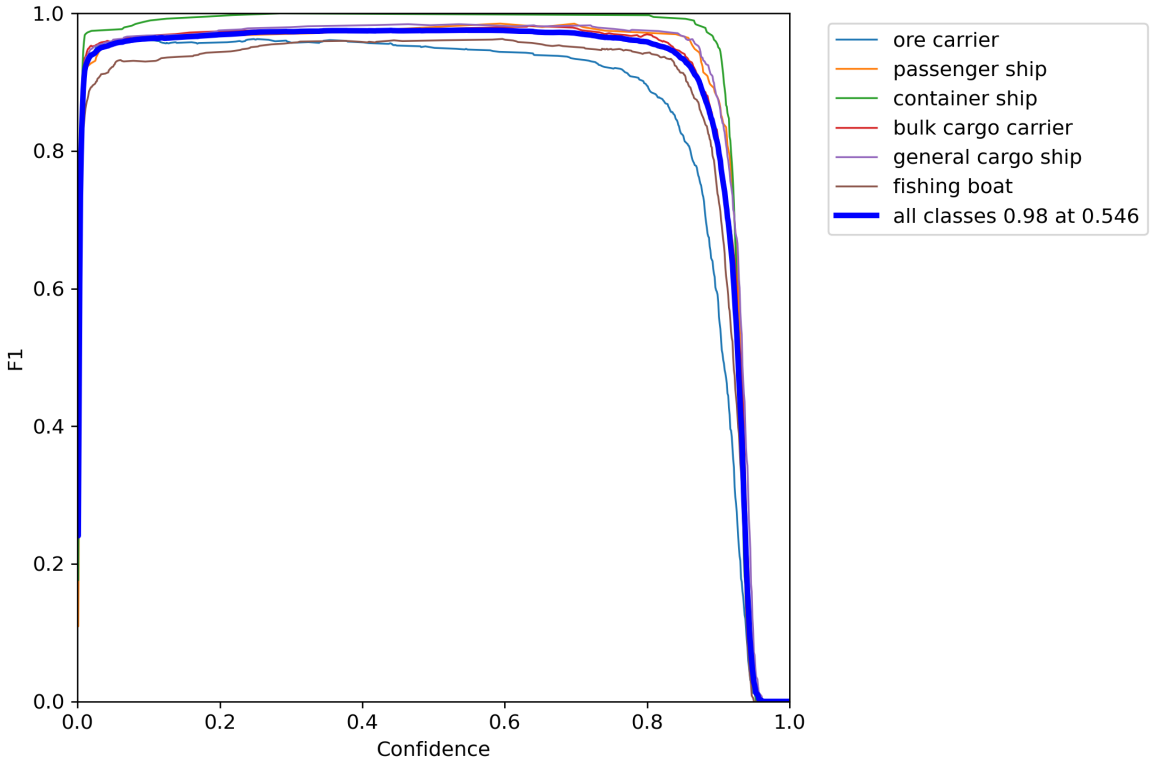
三、结果

1、指标

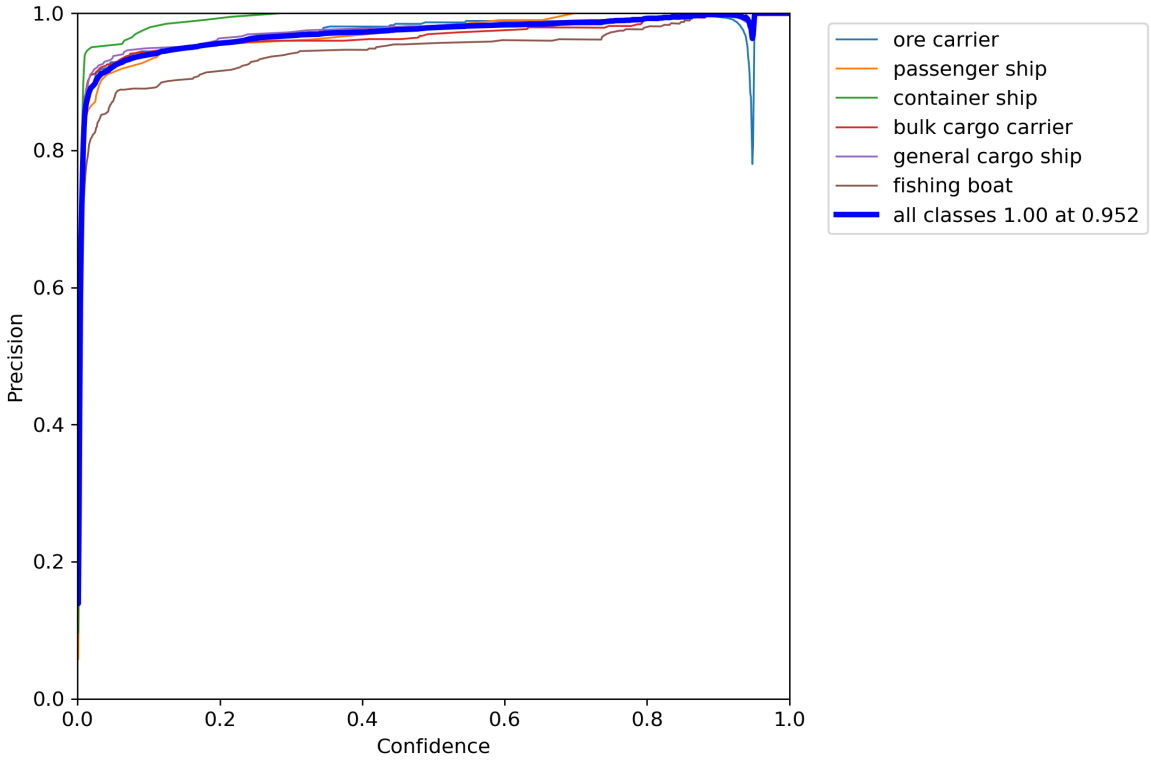
1、混淆矩阵



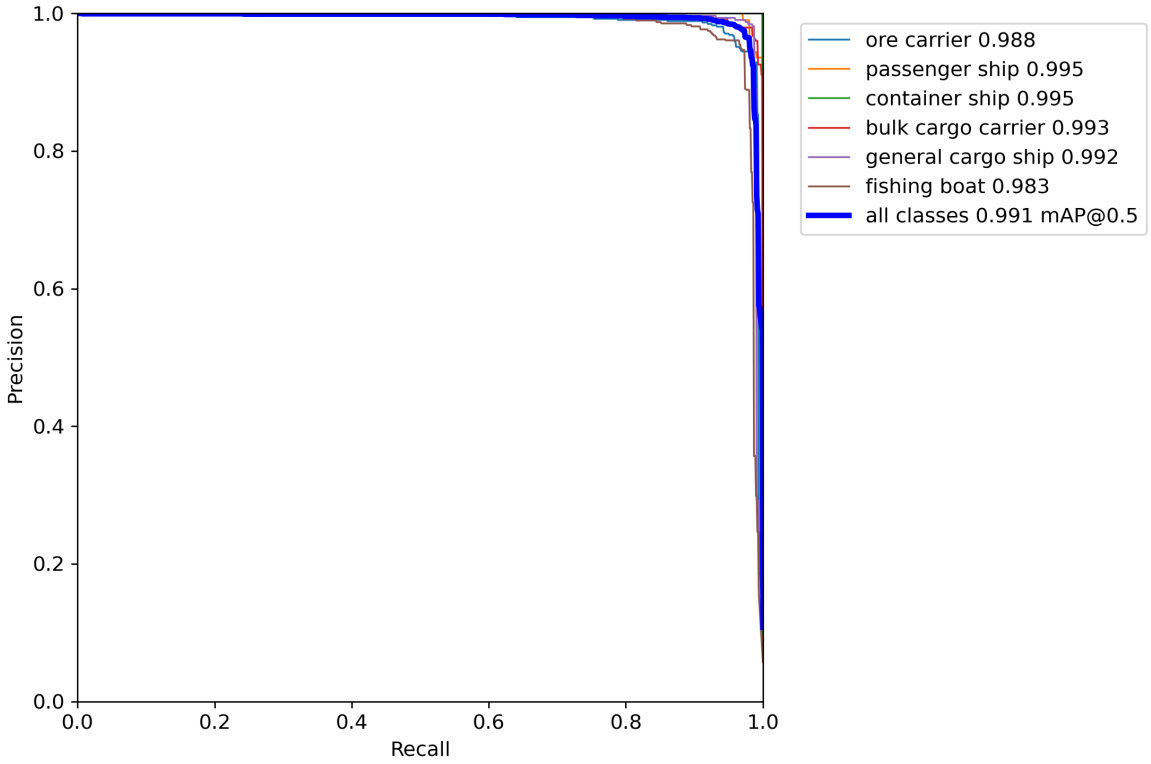
2、F1_curve.png



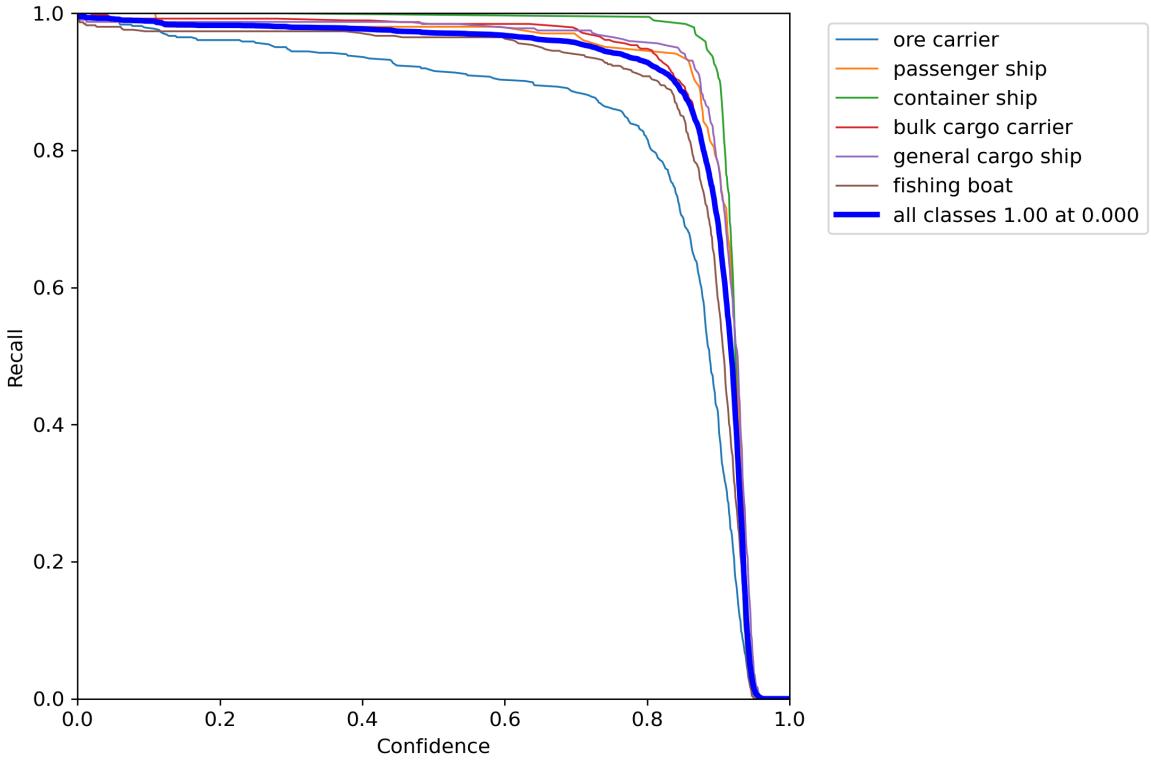
3、P_curve.png



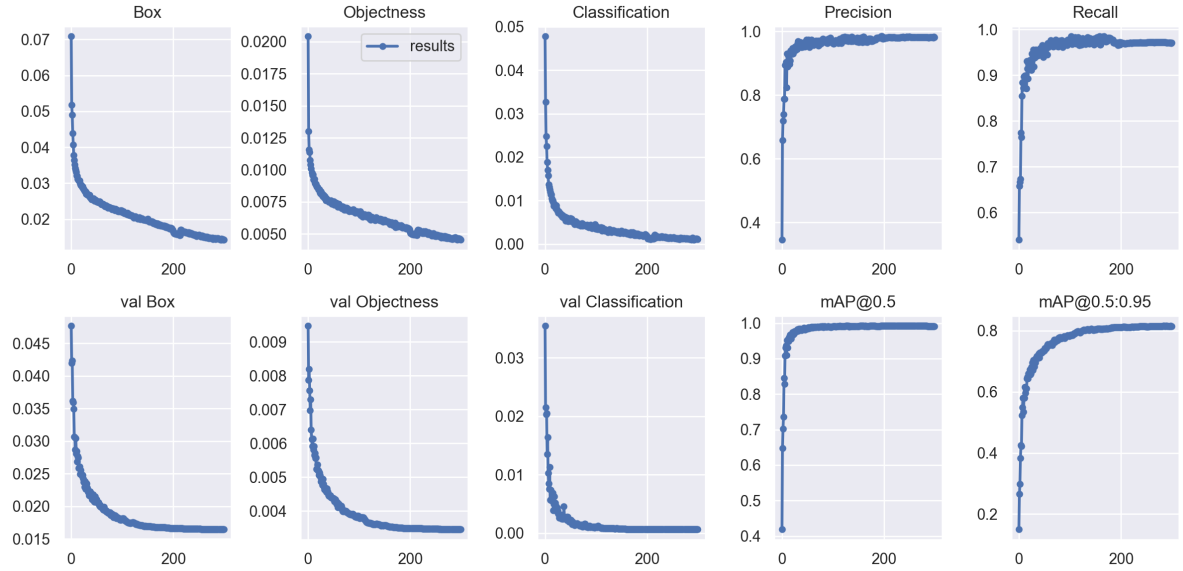
4、PR_curve.png



5、R_curve.png



6、results.png




```
               Class        Images        Labels             P             R        mAP@.5  mAP@.5:.95: 100%|██████████| 47/47 [00:18<00:00,  2.55it/s]
                 all          1491          1943         0.981          0.97         0.991         0.815
         ore carrier          1491           486         0.988         0.912         0.988         0.791
     passenger ship           1491           102         0.985          0.98         0.995          0.81
     container ship           1491           192             1         0.997         0.995         0.861
  bulk cargo carrier          1491           387         0.973         0.984         0.993         0.847
  general cargo ship          1491           319         0.984         0.984         0.992         0.825
        fishing boat          1491           457         0.958         0.965         0.983         0.753
100 epochs completed in 3.138 hours.
```

####