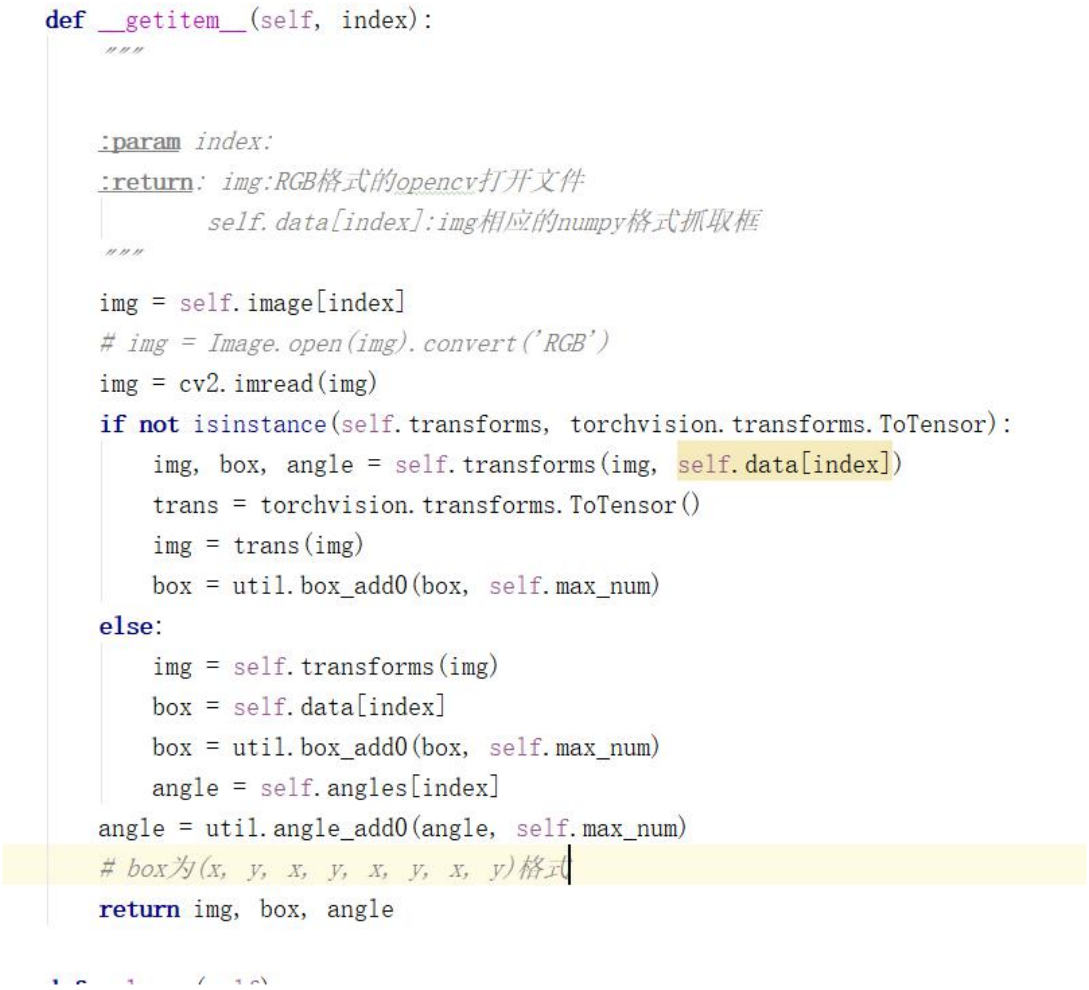
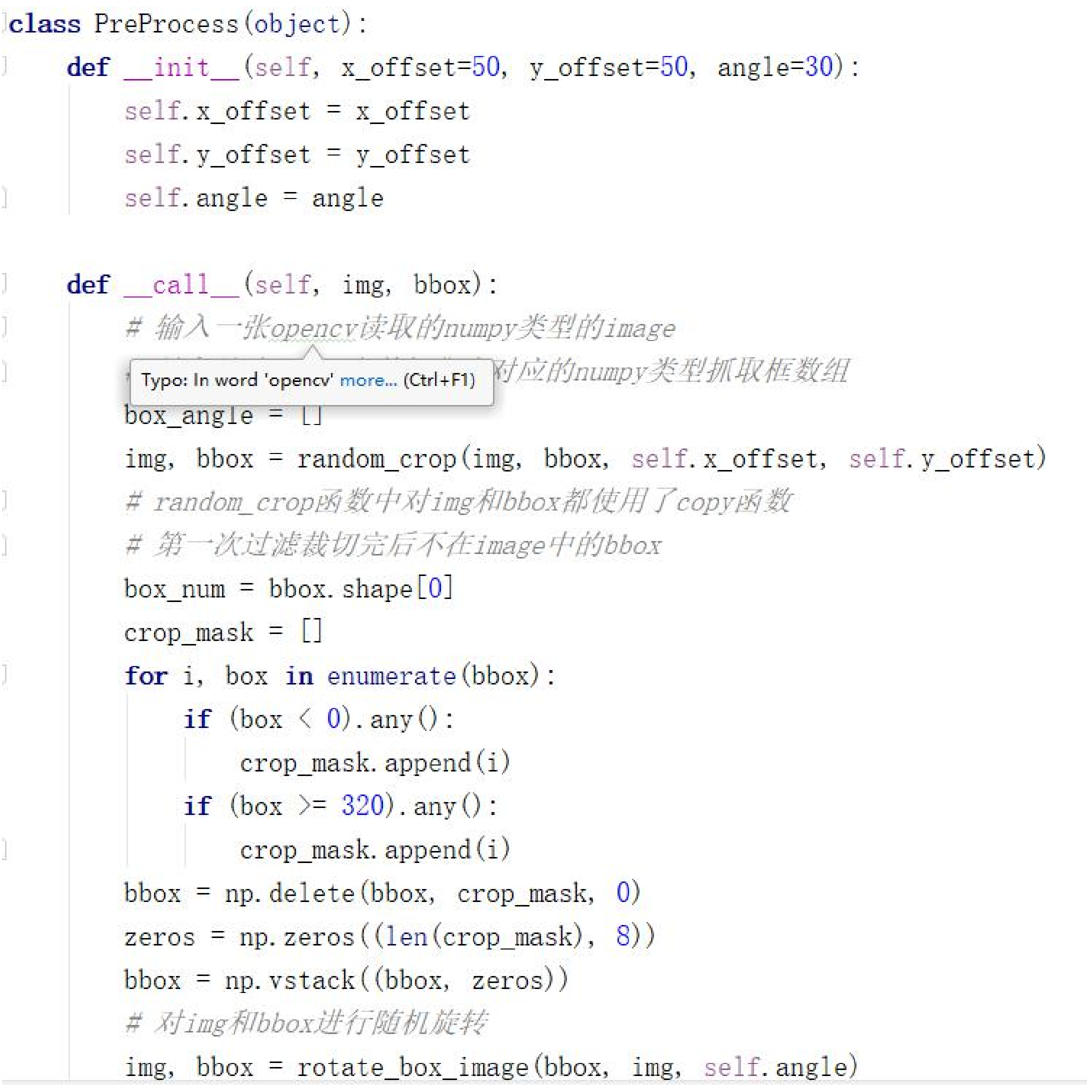
实验进度

数据集以及数据增强



```
def __getitem__(self, index):
    """


    :param index:
    :return: img:RGB格式的opencv打开文件
            self.data[index]:img相应的numpy格式抓取框
    """
    img = self.image[index]
    # img = Image.open(img).convert('RGB')
    img = cv2.imread(img)
    if not isinstance(self.transforms, torchvision.transforms.ToTensor):
        img, box, angle = self.transforms(img, self.data[index])
        trans = torchvision.transforms.ToTensor()
        img = trans(img)
        box = util.box_add0(box, self.max_num)
    else:
        img = self.transforms(img)
        box = self.data[index]
        box = util.box_add0(box, self.max_num)
        angle = self.angles[index]
    angle = util.angle_add0(angle, self.max_num)
    # box为(x, y, x, y, x, y, x, y)格式
    return img, box, angle
```

数据集以及数据增强



```
class PreProcess(object):
    def __init__(self, x_offset=50, y_offset=50, angle=30):
        self.x_offset = x_offset
        self.y_offset = y_offset
        self.angle = angle

    def __call__(self, img, bbox):
        # 输入一张opencv读取的numpy类型的image
        #                    对应的numpy类型抓取框数组
        box_angle = []
        img, bbox = random_crop(img, bbox, self.x_offset, self.y_offset)
        # random_crop函数中对img和bbox都使用了copy函数
        # 第一次过滤裁切完后不在image中的bbox
        box_num = bbox.shape[0]
        crop_mask = []
        for i, box in enumerate(bbox):
            if (box < 0).any():
                crop_mask.append(i)
            if (box >= 320).any():
                crop_mask.append(i)
        bbox = np.delete(bbox, crop_mask, 0)
        zeros = np.zeros((len(crop_mask), 8))
        bbox = np.vstack((bbox, zeros))
        # 对img和bbox进行随机旋转
        img, bbox = rotate_box_image(bbox, img, self.angle)
```

Typo: In word 'opencv' more... (Ctrl+F1)

数据增强以及带角度anchor生成

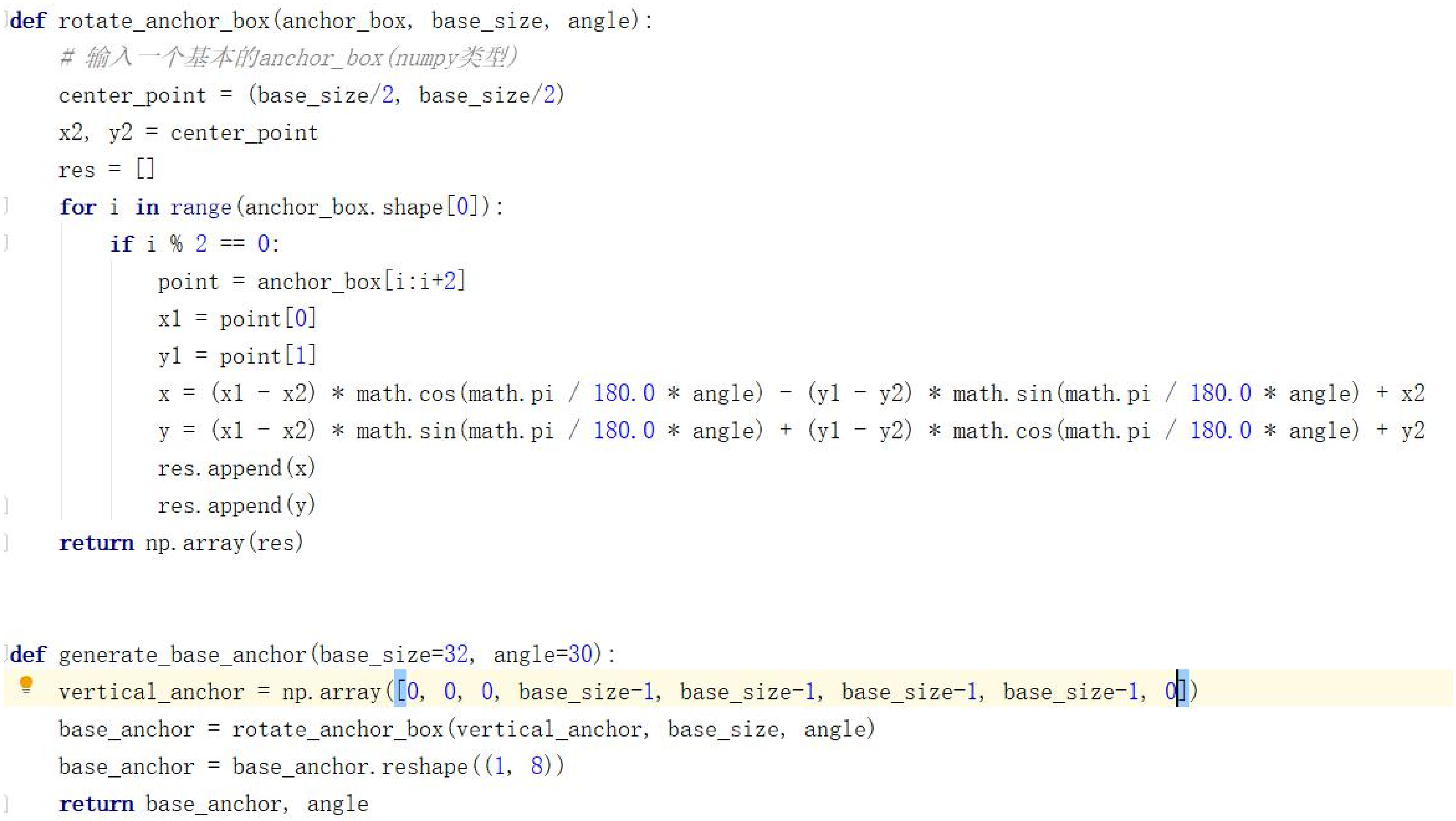
数据增强方式：

先中心随机裁剪（x、y轴随机偏移最多50个像素），然后整张图片顺时针随机旋转最多30°，最后整张图片在x轴和y轴上随机翻转得到结果。

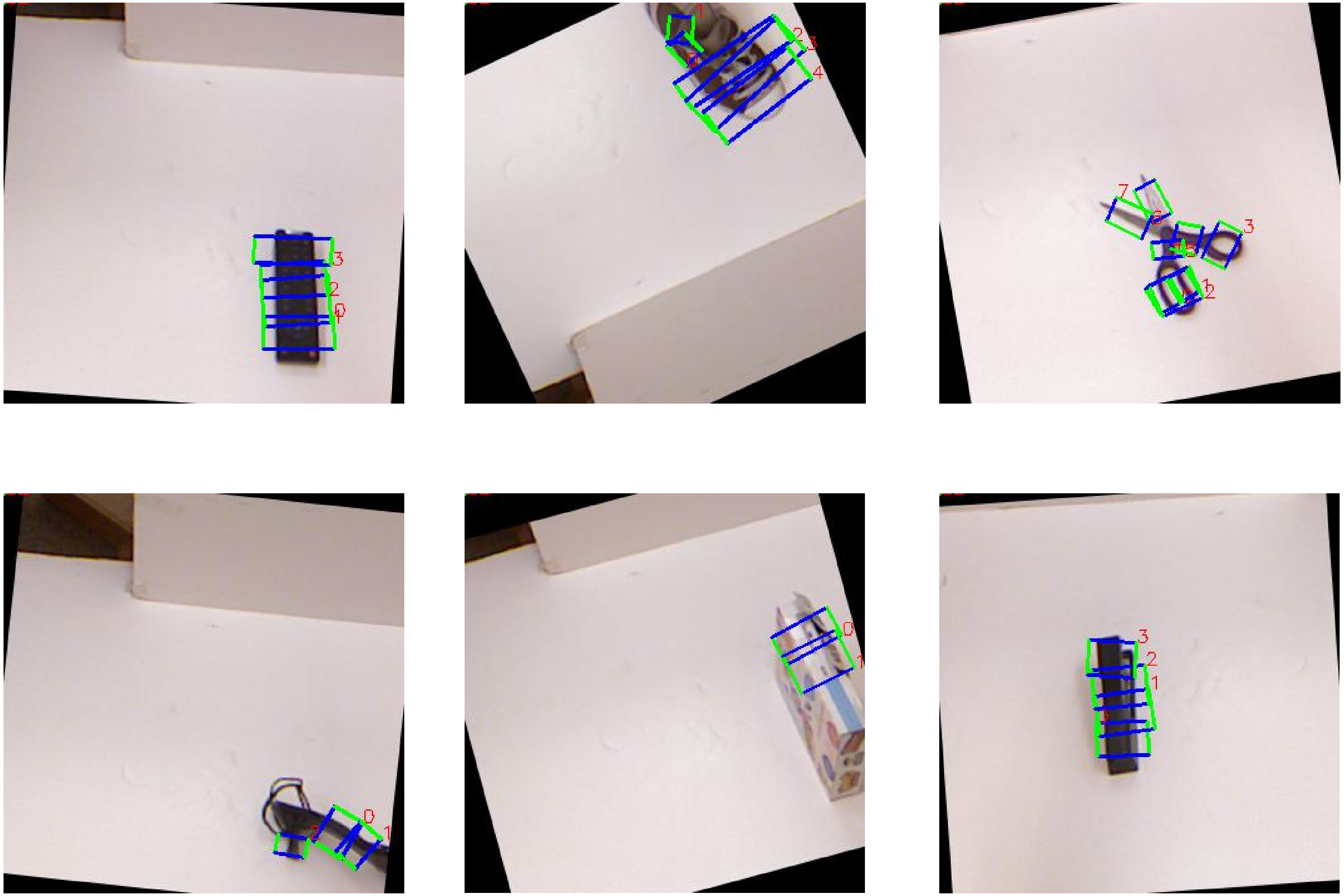
带角度的anchor生成：

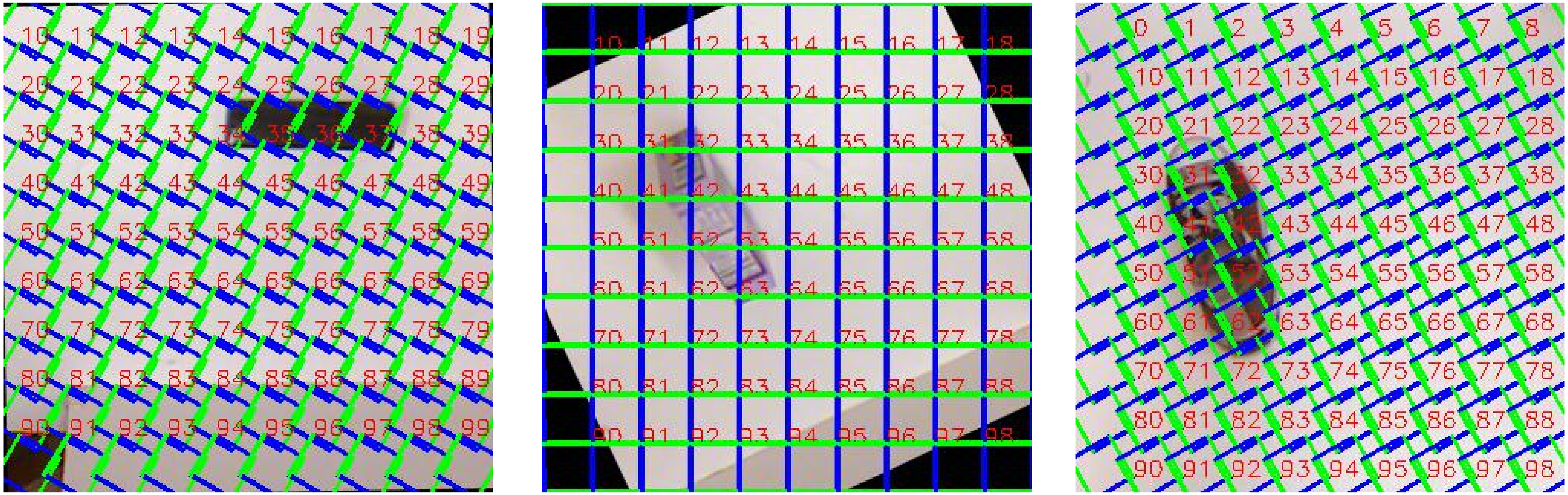
先生成一个普通的不带角度的anchor，之后将anchor的四个顶点坐标围绕anchor中心旋转指定的角度，最后计算anchor在图片上的所有偏移值，将anchor加上偏移值得到结果。

anchor生成



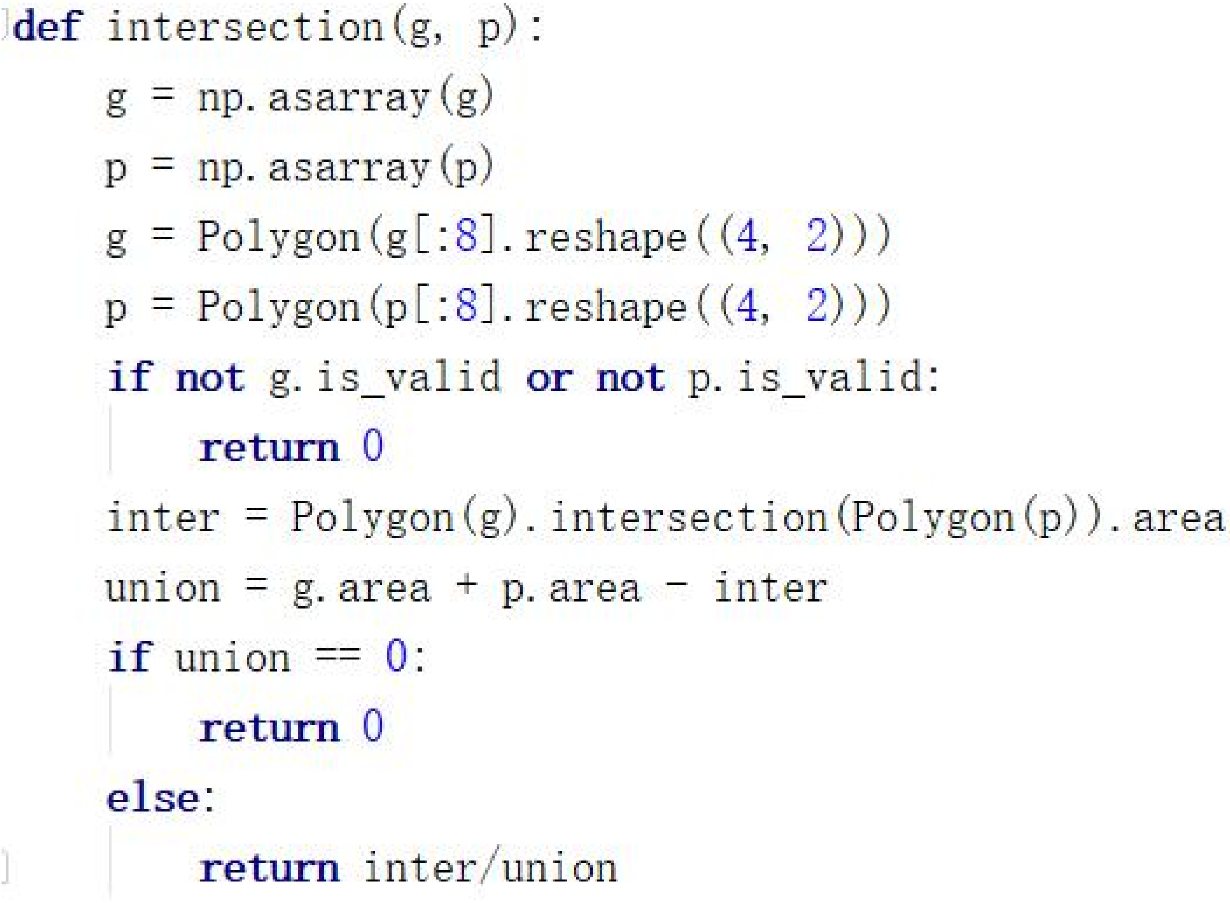
```python
def rotate_anchor_box(anchor_box, base_size, angle):
    # 输入一个基本的anchor_box(numpy类型)
    center_point = (base_size/2, base_size/2)
    x2, y2 = center_point
    res = []
    for i in range(anchor_box.shape[0]):
        if i % 2 == 0:
            point = anchor_box[i:i+2]
            x1 = point[0]
            y1 = point[1]
            x = (x1 - x2) * math.cos(math.pi / 180.0 * angle) - (y1 - y2) * math.sin(math.pi / 180.0 * angle) + x2
            y = (x1 - x2) * math.sin(math.pi / 180.0 * angle) + (y1 - y2) * math.cos(math.pi / 180.0 * angle) + y2
            res.append(x)
            res.append(y)
    return np.array(res)


def generate_base_anchor(base_size=32, angle=30):
    vertical_anchor = np.array([0, 0, 0, base_size-1, base_size-1, base_size-1, base_size-1, 0])
    base_anchor = rotate_anchor_box(vertical_anchor, base_size, angle)
    base_anchor = base_anchor.reshape((1, 8))
    return base_anchor, angle
```





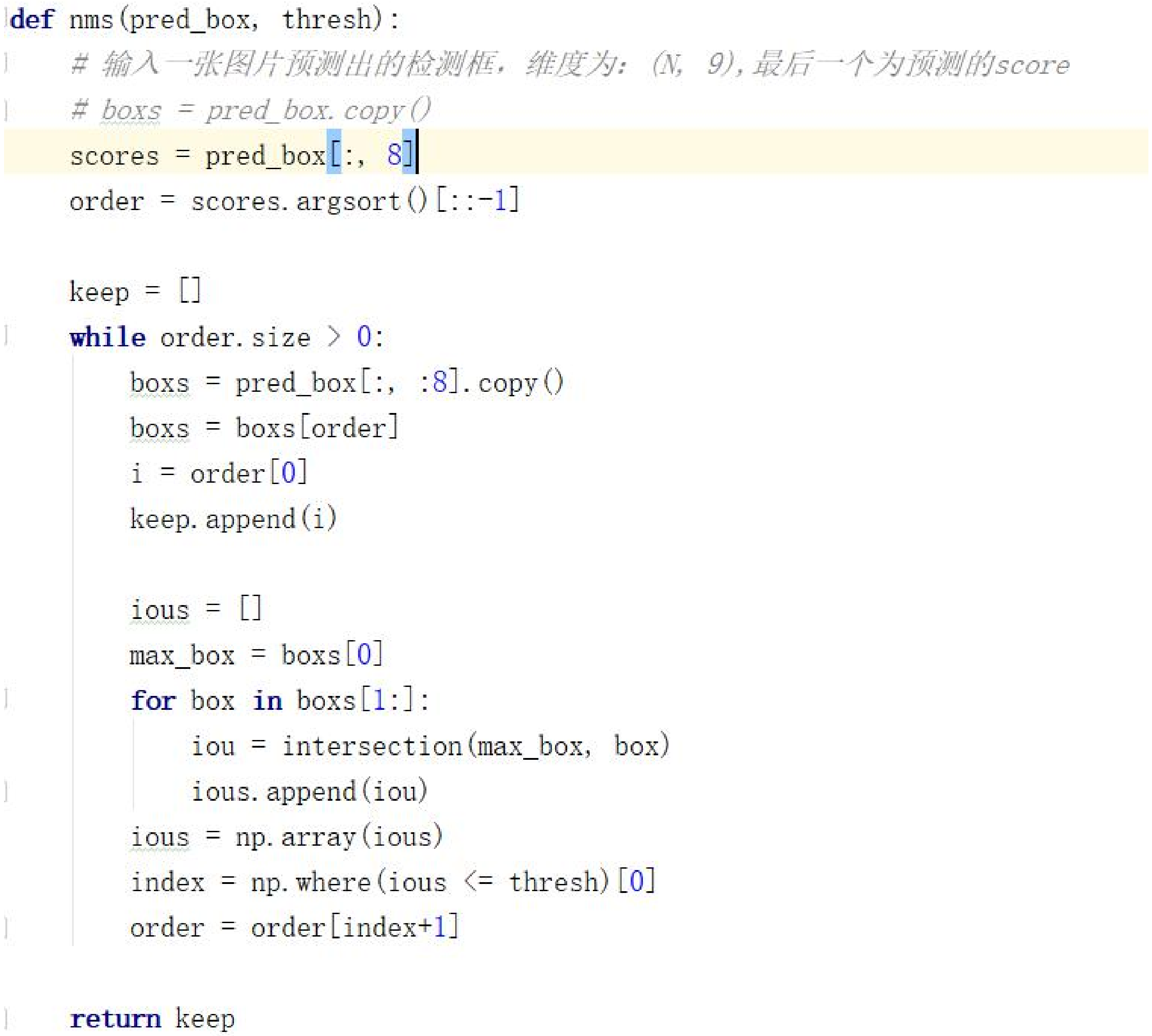
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
10 11 12 13 14 15 16 17 18
20 21 22 23 24 25 26 27 28
30 31 32 33 34 35 36 37 38
40 41 42 43 44 45 46 47 48
50 51 52 53 54 55 56 57 58
60 61 62 63 64 65 66 67 68
70 71 72 73 74 75 76 77 78
80 81 82 83 84 85 86 87 88
90 91 92 93 94 95 96 97 98
0 1 2 3 4 5 6 7 8
10 11 12 13 14 15 16 17 18
20 21 22 23 24 25 26 27 28
30 31 32 33 34 35 36 37 38
40 41 42 43 44 45 46 47 48
50 51 52 53 54 55 56 57 58
60 61 62 63 64 65 66 67 68
70 71 72 73 74 75 76 77 78
80 81 82 83 84 85 86 87 88
90 91 92 93 94 95 96 97 98

倾斜检测框的nms算法



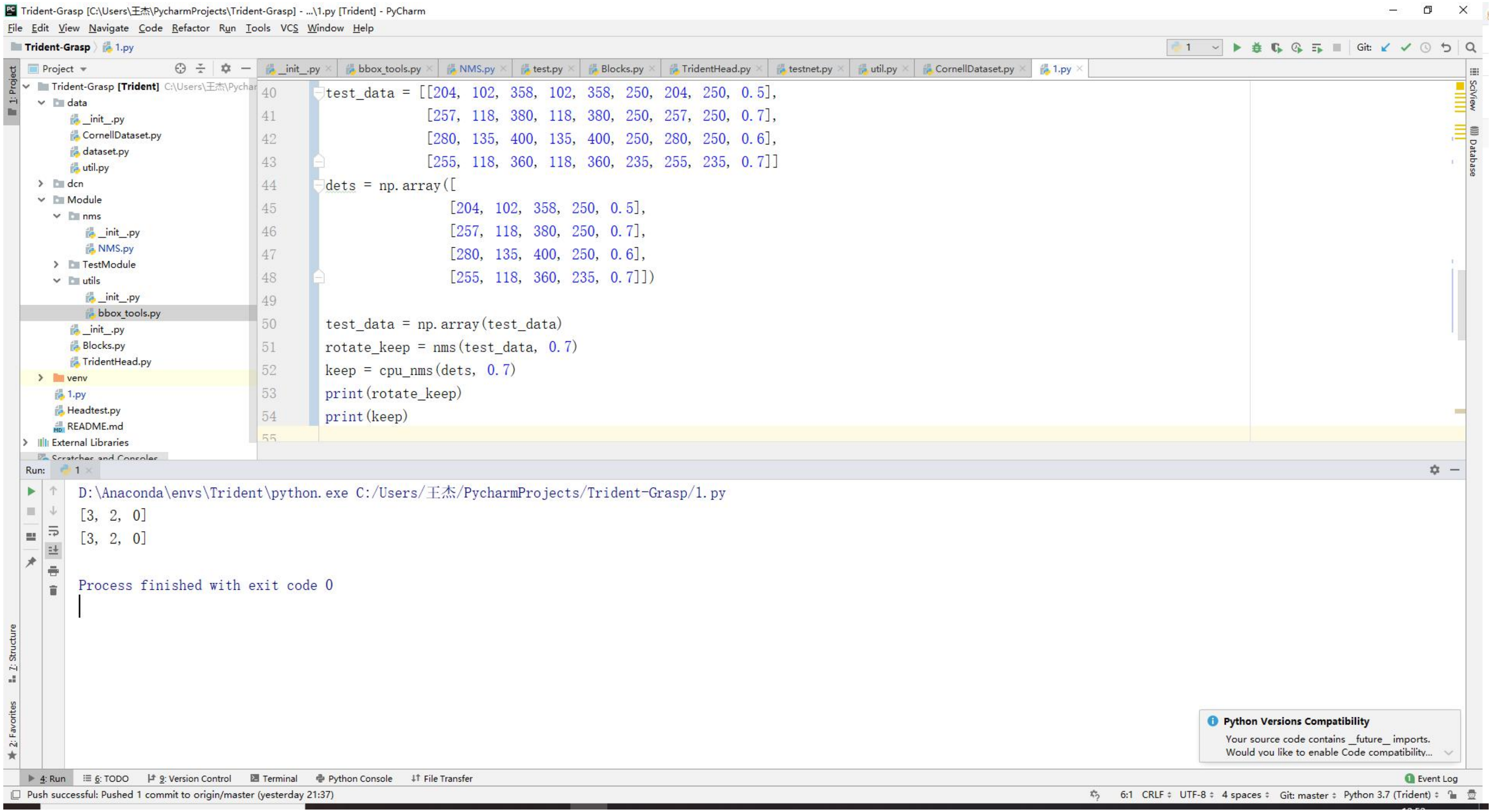
```python
def intersection(g, p):
    g = np.asarray(g)
    p = np.asarray(p)
    g = Polygon(g[:8].reshape((4, 2)))
    p = Polygon(p[:8].reshape((4, 2)))
    if not g.is_valid or not p.is_valid:
        return 0
    inter = Polygon(g).intersection(Polygon(p)).area
    union = g.area + p.area - inter
    if union == 0:
        return 0
    else:
        return inter/union
```

倾斜检测框的nms算法



```python
def nms(pred_box, thresh):
    # 输入一张图片预测出的检测框，维度为：(N, 9),最后一个为预测的score
    # boxs = pred_box.copy()
    scores = pred_box[:, 8]
    order = scores.argsort()[::-1]

    keep = []
    while order.size > 0:
        boxs = pred_box[:, :8].copy()
        boxs = boxs[order]
        i = order[0]
        keep.append(i)

        ious = []
        max_box = boxs[0]
        for box in boxs[1:]:
            iou = intersection(max_box, box)
            ious.append(iou)
        ious = np.array(ious)
        index = np.where(ious <= thresh)[0]
        order = order[index+1]

    return keep
```

倾斜检测框的nms算法



```python
test_data = [[204, 102, 358, 102, 358, 250, 204, 250, 0.5],
             [257, 118, 380, 118, 380, 250, 257, 250, 0.7],
             [280, 135, 400, 135, 400, 250, 280, 250, 0.6],
             [255, 118, 360, 118, 360, 235, 255, 235, 0.7]]
dets = np.array([
                [204, 102, 358, 250, 0.5],
                [257, 118, 380, 250, 0.7],
                [280, 135, 400, 250, 0.6],
                [255, 118, 360, 235, 0.7]])

test_data = np.array(test_data)
rotate_keep = nms(test_data, 0.7)
keep = cpu_nms(dets, 0.7)
print(rotate_keep)
print(keep)
```

```
D:\Anaconda\envs\Trident\python.exe C:/Users/王杰/PycharmProjects/Trident-Grasp/1.py
[3, 2, 0]
[3, 2, 0]

Process finished with exit code 0
```

Python Versions Compatibility

Your source code contains __future__ imports.
Would you like to enable Code compatibility...

Push successful: Pushed 1 commit to origin/master (yesterday 21:37)