6.4 商品物体检测模型训练

- 商品数据集

6.4.1 案例训练结果

6.4.2 案例思路

- image_generator：获取图片数据标注数据生成器
  - 标注数据分割

```
from utils.detection_generate import Generator
```

- 初始化模型参数以及冻结部分结构
- compile与fit_generator

6.4.2.1 获取Generator

- gen = Generator(gt, bbox_util, self.batch_size, self.image_path, train_keys, val_keys, self.input_shape, do_crop=Flase)
  - pickle.load()

6.4.2.3 初始化网络参数，微调网络

- for L in self.model.layers: if L.name in freeze: L.trainable = False

6.4.2.4 设置训练参数以及fit

- compile训练的时候注意使用函数的版本问题：
  - SSD300这个网络源码使用keras 1.2.2 这个库做的，所以里面的模型model,编译训练的时候都是自带的优化器和相关的损失计算
    - keras.optimizers.Adam()
    - 有区别与tensorflow.python.keras.optimizers.Adam()

- ## 6.4.3 多GPU训练代码修改

6.4.4 预测代码

- 预测效果好的：训练损失达到0.1左右的结果

6.5 Opencv-python介绍

pip install opencv-python

6.5.2 cv2视频读取处理

6.5.2.1 摄像头捕获视频

- CV.VideoVapture(0): cap
  - 默认0摄像头
  - cap=cv2.VideoCapture('filename.mp4')文件名及格式
- cap.read(): ret, frame
- cv2.imshow('frame', frame)

6.5.3 cv2 颜色空间变换

- frame：图片内容的数组
- 它是每一帧的画面，视频速度（每秒多少帧FPS）
  - 帧：一张张图片
- OpenCV默认的颜色顺序是BGR
- 对于BGR↔Gray的转换，使用的flag是cv2.COLOR_BGR2GRAY
- 对于BGR↔HSV的转换， 使用的flag是cv2.COLOR_BGR2HSV
- 对于BGR↔RGB的转换，使用的flag是cv2.COLOR_BGR2RGB
- 变换：将格式变换

6.5.4 cv2画图函数

6.5.4 画矩形

画圆形

画文本

```
# 左上角，右下角坐标
cv2.rectangle(frame, (300, 300), (500, 400), (0, 255, 0), 3)
# 圆的中心，半径
cv2.circle(frame, (380, 380), 63, (0, 0, 255), -1)
# 文本内容，左上角位置，
cv2.putText(frame, 'python', (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 4, (255,
255, 255), 2, cv2.LINE_AA)
```

6.6 案例：视频物体检测测试

- 配置获取相关预测数据类别，网络参数
- 获取摄像头视频

    - cap = cv2.VideoCapture(file_path)
- 获取摄像每帧数据，进行格式形状处理
    - 1、图片读取大小(780, 1280, 3)—>(300, 300)
    - 2、BGR—>RGB
    - 注意：保留原始图片数据，要进行画图显示视频要用，画框在原始图片中画
- 模型预测、结果NMS过滤
    - 1、一帧（一张图片）（300， 300， 3）——>(1, 300, 300, 3)
    - 2、x = preprocess_input(np.array(inputs)) y = self.model.predict(x)
    - self.bbox_util.detection_out(y)
    - (, 7308, 21)(, 7308, 8) (, 7308, 4)——>(, 200, 6) 位置+概率+类别
- 画图：显示物体位置，FPS值（每秒帧数）
    - 对于每一帧图片去进行显示
    - 画出这一帧中所有物体框的位置 cv2.rectangle(to_draw, (xmin, ymin), (xmax, ymax), self.class_colors[class_num], 2)
    - 画出文本框

      cv2.rectangle(),cv2.putText()
    - 画出FPS
        - fps = "FPS: " + str(cap.get(cv2.CAP_PROP_FPS))