YOLOv3的挑战集和mAP计算

在同学们完成YOLOv3的模型训练之后要对挑战集进行测试，测试结果需要使用mAP来计算

首先是上传YOLOv3的挑战集，将附件中的picture.zip解压之后，使用其中的lslm-test和eval.txt替换原来的lslm-test和eval.txt。

之后是对挑战集的测试以及mAP的计算。

对infer函数以及主函数进行了改动：

|  |
| --- |
| def infer(image\_path):      origin, tensor\_img, resized\_img = read\_image(image\_path)      input\_w, input\_h = origin.size[0], origin.size[1]      image\_shape = np.array([input\_h, input\_w], dtype='int32')      # print("image shape high:{0}, width:{1}".format(input\_h, input\_w))      t1 = time.time()      batch\_outputs = exe.run(inference\_program,                              feed={feed\_target\_names[0]: tensor\_img,                                    feed\_target\_names[1]: image\_shape[np.newaxis, :]},                              fetch\_list=fetch\_targets,                              return\_numpy=False)      period = time.time() - t1      print("predict cost time:{0}".format("%2.2f sec" % period))      bboxes = np.array(batch\_outputs[0])  # print(bboxes)  # 用于展示一张图片用于预测的效果      if bboxes.shape[1] != 6:          print("No object found in {}".format(image\_path))          return      labels = bboxes[:, 0].astype('int32').tolist()      scores = bboxes[:, 1].astype('float32').tolist()      boxes = bboxes[:, 2:].astype('float32').tolist()        if image\_path == "data/data6045/lslm-test/2.jpg":          last\_dot\_index = image\_path.rfind('.')          out\_path = image\_path[:last\_dot\_index]          out\_path += '-result.jpg'          draw\_bbox\_image(origin, boxes, labels, out\_path)        predict = []      for i in range(len(labels)):          predictTmp = []          predictTmp.append(labels[i])          predictTmp.append(scores[i])          for j in boxes[i]:              predictTmp.append(j)          predict.append(predictTmp)      f = open("./input/detection-results/" + image\_path[24:-4]+'.txt', 'w')      for i in predict:          for j in i:              f.write(str(j) + ' ')          f.write('\n')      f.close()      return predict  if \_\_name\_\_ == '\_\_main\_\_':      if os.path.exists('./input') == False:          os.mkdir('./input')          os.mkdir('./input/detection-results')          os.mkdir('./input/ground-truth')      file\_path = os.path.join(train\_parameters['data\_dir'], 'eval.txt')      images = [line.strip() for line in open(file\_path)]      for line in images:          image\_path = line          parts = line.split('\t')          filename = parts[0]          filename\_path = os.path.join(train\_parameters['data\_dir'], parts[0])          infer(filename\_path)          bbox\_labels = []          for object\_str in parts[1:]:              if len(object\_str) <= 1:                  continue              bbox\_sample = []              object = json.loads(object\_str)              bbox\_sample.append(float(train\_parameters['label\_dict'][object['value']]))              bbox = object['coordinate']              bbox\_sample.append(float(bbox[0][0]))              bbox\_sample.append(float(bbox[0][1]))              bbox\_sample.append(float(bbox[1][0]))              bbox\_sample.append(float(bbox[1][1]))              bbox\_labels.append(bbox\_sample)          f = open("./input/ground-truth/" + filename\_path[24:-4]+'.txt', 'w')          for i in bbox\_labels:              for j in i:                  f.write(str(j) + ' ')              f.write('\n')          f.close() |

在运行完成之后，初始目录下会多了一个input目录，里面包含了detection-results和ground-truth两个目录。分别存放着预测值以及真实值。

将文件里附带的main.py文件上传到初始目录之下，同时运行这个文件，即可得到整个预测结果的mAP，同时初始目录下会出现一个output目录，其中存放着和mAP相关的更加详细的数据。