

PCB 缺陷检测模型代码使用说明

作者：唐三立

完成时间：2018/12/13

邮箱：tangsanli@sjtu.edu.cn, 51774885@qq.com

一、简介

本代码为深度学习框架下的 PCB 缺陷检测模型训练与测试代码，对每一张二值化的被测图，均有一张配准的不含缺陷的模板图，这对图像同时输入 PCB 缺陷检测模型，通过模型计算得到被测图上缺陷的位置与类型。本代码具有一定的创新点，代码中不包含数据集（数据集见第四章介绍）。此模型用于检测常见的六类 PCB 缺陷：断路、短路、针孔、露铜、毛刺及缺口，模型能够稳定检测 3x3 尺寸及以上的缺陷，同时对尺寸小于 3x3 的缺陷泛化能力也较强（详见第七章实验分析）。

本文提出的缺陷检测模型与传统的目标检测模型有相似也有不同。其中，

相似：（1）缺陷检测任务是目标检测任务的一种特例；

（2）与目标检测任务有相同的评估指标，如 mAP 与 F-score 等；

不同：（1）检测的目标为具有纹理特征的 PCB 缺陷；

（2）模型的输入为一对具有相同尺寸的模板图与被测图，而非传统目标检测中的单张图片，因此需要同时处理这一对图片，并进行特征融合；

（3）模型中创新地提出了分组金字塔池化结构，在保持很高的计算效率情况下大大提高了模型的检测性能；

为了避免重复造车轮，此代码在 ssd 开源代码¹进行修改，增加了分组金字塔池化模块，以及处理成对输入的模板图与被测图特征融合模块，大大提高了模型对尺寸变化较大的 PCB 缺陷的检出能力。

模型的检出结果图 1 所示：

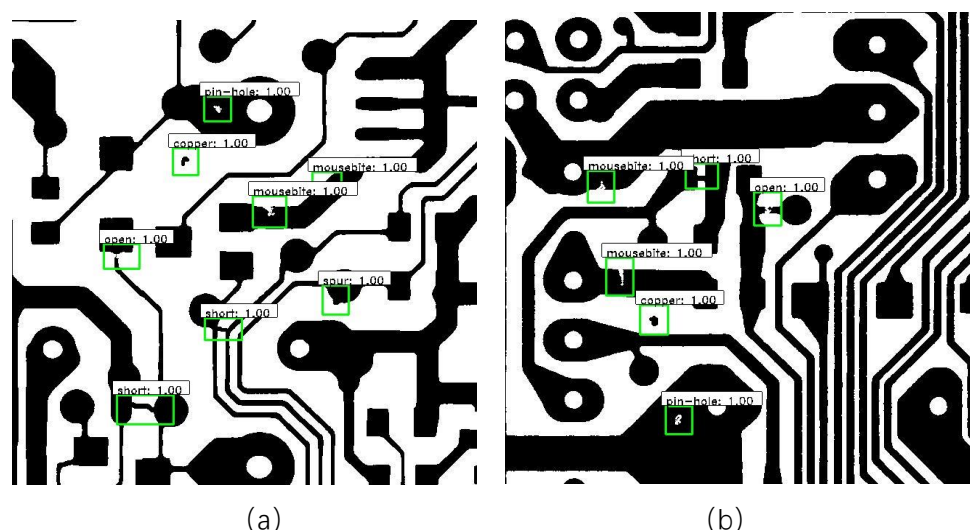


图 1. PCB 缺陷检测模型结果图，其中绿色框为检出框，字符为检出缺陷类别，数字为置信度（0 至 1），由于缺陷特征明显，此处省略展示两张对应的无缺陷的模板图

¹ ssd 模型开源代码：https://github.com/pierluigiferrari/ssd_keras

二、 环境配置

硬件环境：

推荐：GPU TITAN X （显存 12GB），SSD 固态硬盘 128G；

最低：GPU （显存大于等于 2G）

注：模型参数两仅约 2M，独立显卡一般均能运行测试。

系统环境：

Ubuntu 16.04

Tensorflow 1.8.0

Keras 2.2.2

python 2.7

三、 评估方法

检测性能评估方法²：

- (1) mAP：即每一类缺陷的 pr 曲线下方面积的均值；
- (2) F-score：计算公式如下，其中 P, R 分别表示准确率(precision)与召回率(recall)。
$$F\text{-score} = 2 * P * R / (P + R)$$

计算效率评估方法：

- (1) FPS：帧率，即每秒可完成检测的图片个数。

四、 数据集下载

可以使用本文作者标注的包含 1,500 对尺寸为 640x640 的模板图与被测图的数据集 *DeepPCB* 对模型进行训练及测试³，将下载的 DeepPCB 数据集放置于 *./datasets/* 下，此时数据集的目录结构应该为：

```
-ssd-PCB-GPP-MaxPooling/  
  -bounding_box_utils/  
  -data_generator/  
  -datasets/  
    -PCBData/  
      -group00041/  
      -group12000/  
      -group12100/  
      - ...  
      - test.txt  
      - trainval.txt  
  - eval_utils  
  - ...
```

² 评估代码详见：<https://github.com/tangsanli5201/DeepPCB/evaluation>

³ DeepPCB 数据集下载路径为：<https://github.com/tangsanli5201/DeepPCB>

五、 模型训练

若不想重复训练模型，可直接使用已经训练好的模型参数：`./ssd7_epoch-490_loss-0.2872.h5` 直接跳至第六章进行模型的测试。否则，需要按照如下步骤配置训练参数：

1. 模型训练脚本为 `./train_pcb.py`，需要根据当前项目路径适当修改内容：
 - (1) **86 行**：train_labels_filename = “至 DeepPCB 数据集中 trainval.txt 的绝对路径”
 - (2) **111 行**：batch_size, 单步训练的 batch 大小，在 12G 内存的 GPU 中推荐设为 16
 - (3) **163 行**：final_epoch, 修改模型的最大迭代次数，推荐使用 500 次（约训练半天时间）

2. 对 DeepPCB 数据集中的 trainval.txt 与 test.txt 中每行的图片与标注文件路径修改为对应的绝对路径（推荐使用 sublime text 进行批量修改），如 trainval.txt 中将每个 group 单词前加上到数据集根目录的文件夹路径：

/path/to/your/dataset/PCBData/group20085/20085/20085000.jpg

/path/to/your/dataset/PCBData/group20085/20085/20085000.txt

...

(注：上述两行在 trainval.txt 中为一行，使用空格分开，此处只是自动换行了。)

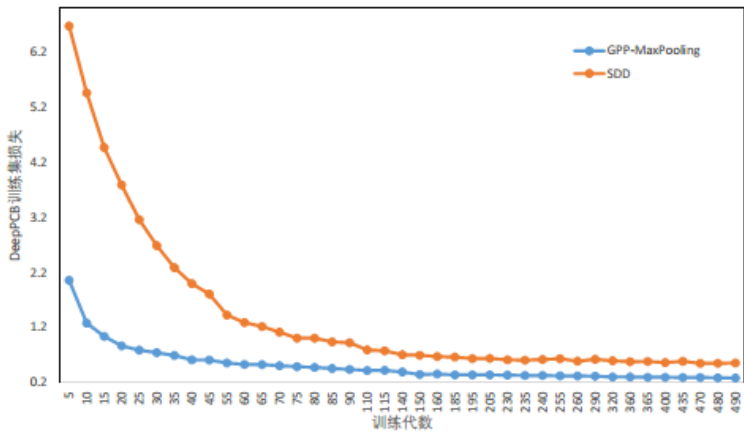
3. 按照上述步骤配置完参数后，在项目文件夹根目录下运行以下指令进行训练：

`python train_pcb.py`

4. 训练过程中得到的模型参数将自动保存在项目文件夹根目录下。

模型的训练过程如图 2 所示：

图 2 模型在 DeepPCB 数据集上训练过程曲线



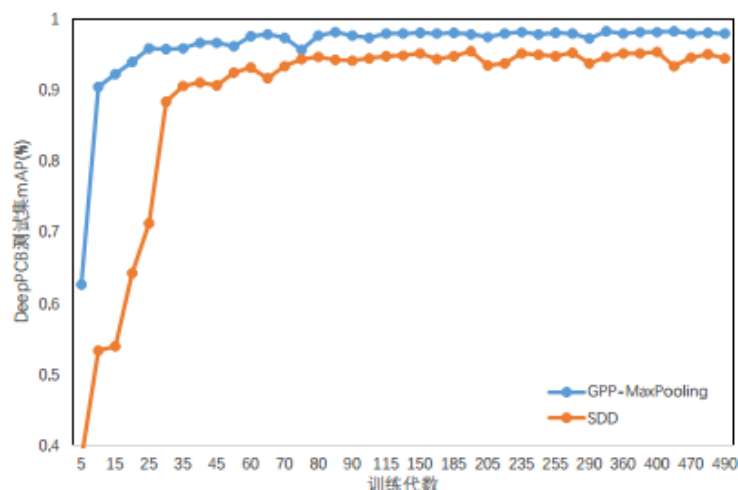


图 2 PCB 缺陷检测模型训练过程

六、 模型测试

模型的测试脚本为 `./test_pcb.py`，需要简单修改测试路径与测试 batchsize 即可：

1. 修改测试脚本 `./test_pcb.py`：
 - (1) 162 行: `weights_path =` “模型网络参数的绝对路径”，如: `"/path/to/this/project/ssd7_epoch-490_loss-0.2872.h5"`;
 - (2) 184 行: `test_txt =` “至 DeepPCB 数据集中 `test.txt` 的绝对路径”
 - (3) 134 行: `confidence_threshold =` 置信度阈值 0 至 1 浮点数，小于该阈值的不会被检出，推荐设置 0.8 至 0.9。
 - (4) 136 行: `batch_size =` 单步测试的 batch 大小，与测试时间有关，不影响测试性能，推荐设置 16。
2. 按照第五章 2.中所示修改测试集 `test.txt` 中的路径为绝对路径。
3. 按照上述步骤配置完参数后，在项目文件夹根目录下运行以下指令进行测试：
`python test_pcb.py`
4. 预测图片结果位于 `./images/` 路径下。

七、 说明

本文为投至 ICASSP2019 会议论文（在审阶段）：**ONLINE PCB DEFECT DETECTOR ON A NEW PCB DEFECT DATASET** 所完成的代码使用说明，仅供学术研究使用，未经本人允许，严禁重复发表，严禁商用。