Lab: PCB Defect Detection

Xueqian Zhang, Yingjing Jiang

Lab Description

This project code is modified based on Tiny-Defect-Detection-for-PCB. Minor changes are made so that it would be more readable and executable for students whose computers don't have GPUs and need specific details on how to implement this wonderful project and some common problems that might be encountered during the process. Therefore if you have GPU on your laptop and would like to learn more about this project you could check the original website.

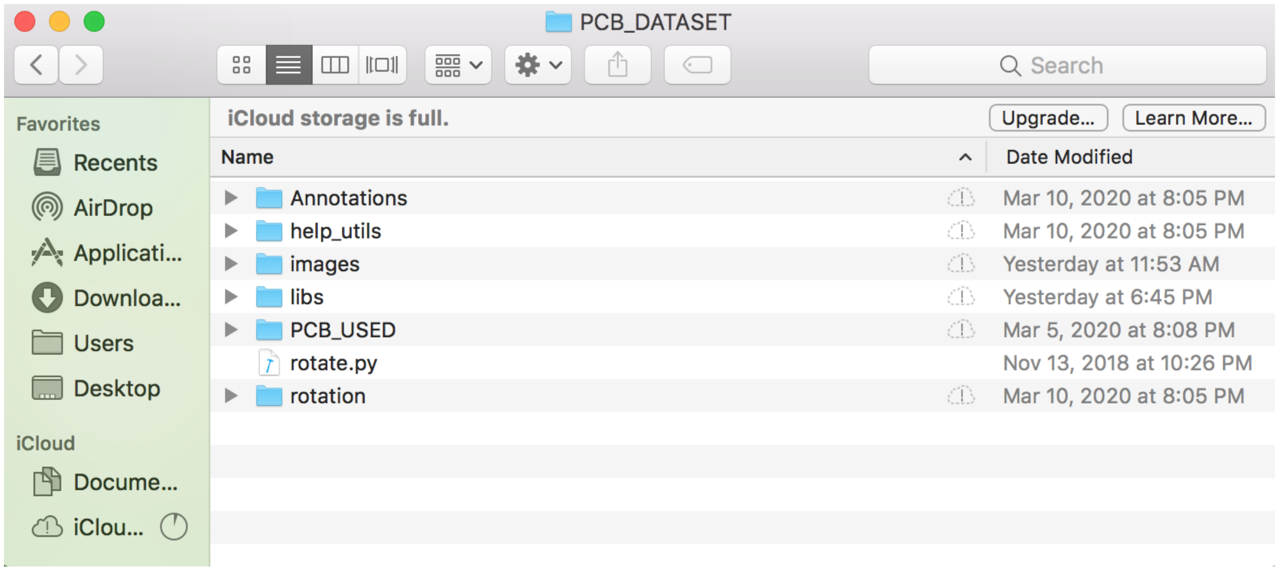
1) This lab explains how to download, and install python3.6 and some corresponding packages, information of the PCB data set that you are going to use, and the process for running the codes to implement the deep learning model to get the defect detection results as well as the model evaluation.

2) This lab is written for Mac OS system. If you use a different system, you could also follow most of the steps since they should be compatible to other systems.

3). If you have specific questions for the system option or packages installation, contact the instructor for advice.

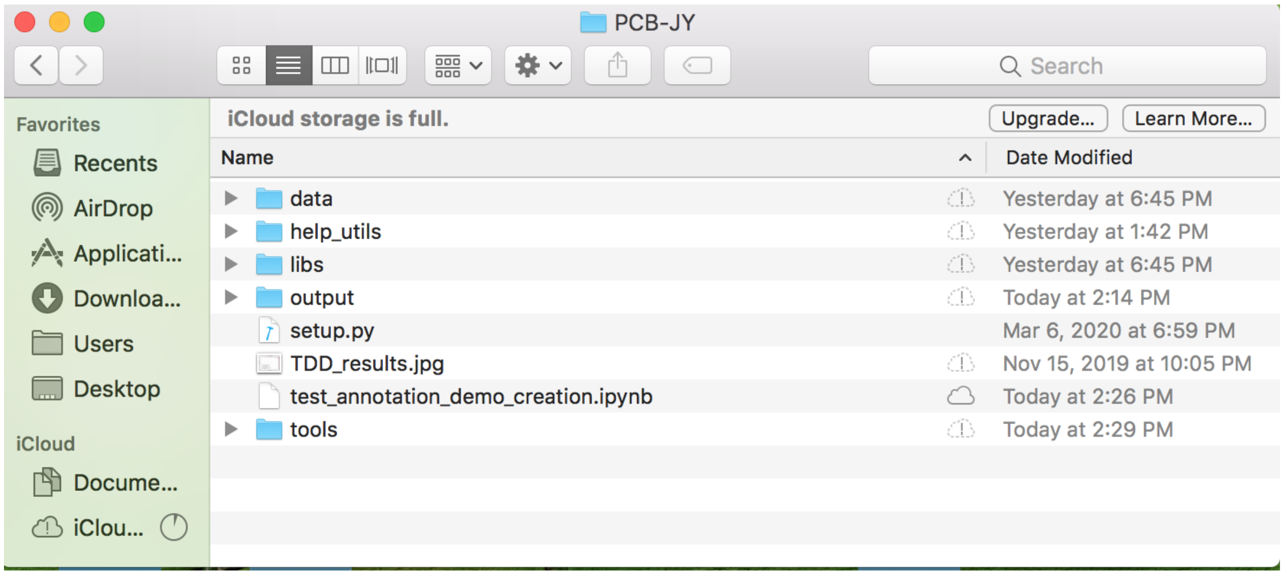
Task 1: Download PCB Defect Dataset and Program

1. Download PCB defect dataset and program from Google Drive, unzip to two different folders called PCB_DATASET and PCB-JY, where you will see the dataset and program folders organized as Figure 1 and Figure 2.



Name	Date Modified
Annotations	Mar 10, 2020 at 8:05 PM
help_utils	Mar 10, 2020 at 8:05 PM
images	Yesterday at 11:53 AM
libs	Yesterday at 6:45 PM
PCB_USED	Mar 5, 2020 at 8:08 PM
rotate.py	Nov 13, 2018 at 10:26 PM
rotation	Mar 10, 2020 at 8:05 PM

Figure 1: PCB_DATASET

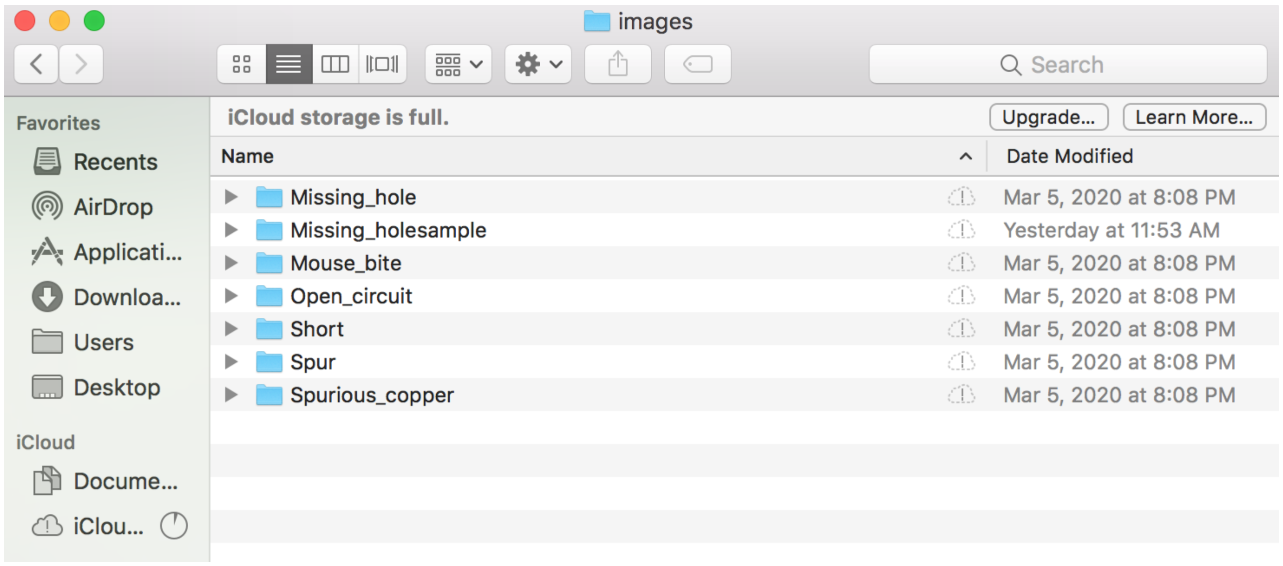


Name	Date Modified
data	Yesterday at 6:45 PM
help_utils	Yesterday at 1:42 PM
libs	Yesterday at 6:45 PM
output	Today at 2:14 PM
setup.py	Mar 6, 2020 at 6:59 PM
TDD_results.jpg	Nov 15, 2019 at 10:05 PM
test_annotation_demo_creation.ipynb	Today at 2:26 PM
tools	Today at 2:29 PM

Figure 2: PCB-JY

2. In PCB_DATASET/images, there are 6 types of defects made by photoshop, a graphics editor published by Adobe Systems. The defects defined in the dataset are: missing hole, mouse bite, open circuit, short, spur, and spurious copper. (Figure 3)

   For example, you can see a missing hole in Figure 4.



Name	Date Modified
Missing_hole	Mar 5, 2020 at 8:08 PM
Missing_holesample	Yesterday at 11:53 AM
Mouse_bite	Mar 5, 2020 at 8:08 PM
Open_circuit	Mar 5, 2020 at 8:08 PM
Short	Mar 5, 2020 at 8:08 PM
Spur	Mar 5, 2020 at 8:08 PM
Spurious_copper	Mar 5, 2020 at 8:08 PM

Figure 3: Images

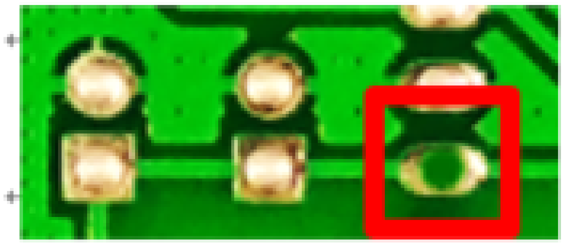


Figure 4: Missing Hole

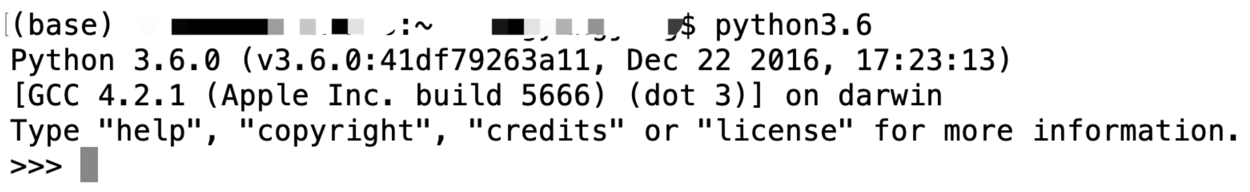
3. There are many files in PCB-JY folders. Some basic files and folders that you need to know their functions. For instance, tools folder contains most of the executive *.py files that you would execute later. The inference.py will employ the training model to get the results with detection on images. The eval.py will get the evaluation of your detection precision of your training model. The test.py is similar to inference.py and used for demo images. For simplicity you could use inference.py in the tools folder for both demo images and all images and only change the path of your image(The instructions are shown in the latter pages) The train.py file is used for training with the dataset to gain the weights for the model. In this lab, you don't need to train this model since it will take a long time. If you would like to train your own model, you could see Tiny-Defect-Detection-for-PCB Train part.

   The other folders libs and help_utils and data are what needed to be imported while running *.py files. They contain functions or parameters' values needed in your main functions of the *.py files. For instance, the model parameters that needed for the training network can be checked in '$PATH_ROOT/PCB-JY/libs/configs/cfgs.py'. Modification of this file will change your training parameters and hence your training network.

Task 2: Download and Install Python3.6 and Packages

1. Download the Python3.6 at the buttom of official website, depending on the operating system for your computer. An executable file will be downloaded to your computer.

2. Execute the downloaded file to install Python3.6. You can run *python*3.6 in terminal to check whether it is successfully installed. It should show *Python* 3.6.0 as Figure 5.



```
(base) [redacted]:~ [redacted]$ python3.6
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 5: Python3.6 Installation

3. Once Python3.6 is installed successfully, you can start to install packages for this project. Execute the following code in terminal to install those packages.

   1) tensorflow1.14.0

   ```
   python3.6 -m pip install tensorflow==1.14.0
   ```

   2) tfplot

   ```
   python3.6 -m pip install tensorflow-plot
   ```

   3) opencv(cv2)

   ```
   python3.6 -m pip install opencv-python
   ```

   4) numpy

   ```
   python3.6 -m pip install numpy
   ```

   5) Cython

   ```
   python3.6 -m pip install Cython
   ```

   6) Image

   ```
   python3.6 -m pip install Image
   ```

   7) sklearn

   ```
   python3.6 -m pip install sklearn
   ```

*TODO : may have other packages : time, os, ...need a new computer without python*3.6 *to try*

Task 3: Change Working Directory of the Program

There are some changes of working directory you need to make to make sure the program can run successfully in your computer.

1. In PCB-JY/lib/configs/cfgs.py, at line37, change test_annotate_path to your own path.

2. In PCB-JY/output/trained_weights/FPN_Res101_0117_OHEM/checkpoint, change the two working directory to your own path. You can use the code below to help you open this file.

```
    cd $PATH_ROOT/output/trained_weights/FPN_Res101_0117_OHEM/
    emacs checkpoint
```

   then, change the working directory and press Ctrl-x and Ctrl-s to save the changes.

Task 4: Test the Program with Demos

Some demo images have already been extracted out and provided in demos_backup folder. (check PCB-JY/tools/demos_backup/). Now you can use these images to first test the correctness of your program running.

1. Change your working directory into '$PATH_ROOT/tools/' in your terminal.

```
        cd $PATH_ROOT/tools/
```

2. Run the following command in your terminal to get the detected images and save them in the demo_results folder.

```
        python3.6 inference.py --data_dir='demos_backup'
                               --save_dir='demo_results'
```

3. Create the corresponding demos for test_annotaions. Run the test_annotation_demo_creation.ipynb file under your PCB-JY directory in the Jupyter Notebook. If you haven't installed Jupyter Notebook before, you could run the following command in your terminal.

```
       pip install notebook
```

   Congratulations, you have installed Jupyter Notebook! To run the notebook, run the following command at the terminal.

```
       jupyter notebook
```

   Open the test_annotation_demo_creation.ipynb file and run it cell by cell. To learn more about running the Jupyter Notebook, see Jupyter Notebook.

4. Run the following command in your terminal to get the evaluation result figure for demos. You can compare all the results with the result in demo_results.

```
       python3.6 eval.py --eval_imgs='demos_backup'
                         --annotation_dir='test_annotation'
                         --GPU='0'
```

Task 5: Run the Program with All Images

Make a new folder to put images after detecting, and get the working directory of your downloaded dataset (data_dir) and the new folder (save_dir). Follow the instructions below, and you will find the detected results of images in the new folder.

1. Change your working directory into '$PATH_ROOT/tools' in your terminal.

```
cd $PATH_ROOT/tools/
```

2. In your 'PCB_DATASET/images/* directory', drag all the images out to directly under 'PCB_DATASET/images/'. Similarly, in your 'PCB_DATASET/Annotations/*/' directory, drag all the xml files out to directly under 'PCB_DATASET/Annotations/'.

   Note : Here the images should all be extracted out from different folders. That is to say you need to confirm that the files under path 'PCB_DATASET/images/' are all jpg files not folders (Missing_hole,Spur,...).

3. Create a new directory to save the results (for example PCBresults). Run the following command in your terminal to get the detected images for all images and save them in the PCBresults folder.

```
python3.6 inference.py --data_dir='$PATH/PCB_DATASET/images/'
                       --save_dir='$PATH/PCBresults'
```

4. Run the following command in your terminal to get the evaluation result figure for all images.

```
python3.6 eval.py --eval_imgs='$PATH/PCB_DATASET/images/'
                  --annotation_dir='$PATH/PCB_DATASET/Annotations'
                  --GPU='0'
```

Task 6: Possible Problems and Solutions

If you meet any problems, you could try the following steps to debug:

1. Check whether you have any packages needed that you forget to download.
2. Check if the version of the python or tensorflow you are running is different from the version suggested.
3. Check whether you have changed all the paths correctly.
4. If 1-3 steps don't work, according to the error message, go directly to that *.py file or other files, comment out some lines and debug from the beginning until you find the line that cause the problem.