Object Detection: Keras Implementation of MaskRCNN

Prepared By: Lim Jun Jie

A. Introduction

Object detection is a subset of computer vision which is also an automated method for locating interesting objects in an image with respect to the background. By utilizing the object detection algorithm, it can help us to do a lot of works. For example, we can use it for pothole defect detection in the smart car, batch inspection of the defected PCB boards, etc. Hence, it is good for us to know how object detection works based on the deep learning and computer vision. In order to do the object detection, we will implement Convolutional Neural Network (CNN).

There is a lot of methods in CNN used for object detection such as YOLO, Fast RCNN, Faster RCNN, Mask RCNN, etc. However, this tutorial will focus on the implementation of MaskRCNN which is classified under the region proposal network and it is the extending Faster RCNN for the level segmentation. For more information about these algorithms, please refer to https://medium.com/@umerfarooq_26378/from-r-cnn-to-mask-r-cnn-d6367b196cfd (explanation of the evolution of regional proposal network from RCNN to Mask RCNN) and https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e . (explanation of YOLO and Region Proposal Network)

Does it look like complicated? Don't worry, we will guide you step by step for the Keras implementation of MaskRCNN for you to create your own object detection model in this tutorial.

B. Minimum Requirement

The model is trained by using Google Colab. After that, it is tested using Windows 10, AMD RYZEN 3550H, NVDIA GeForce GTX1650 (4GB GPU).

C. Project Setup

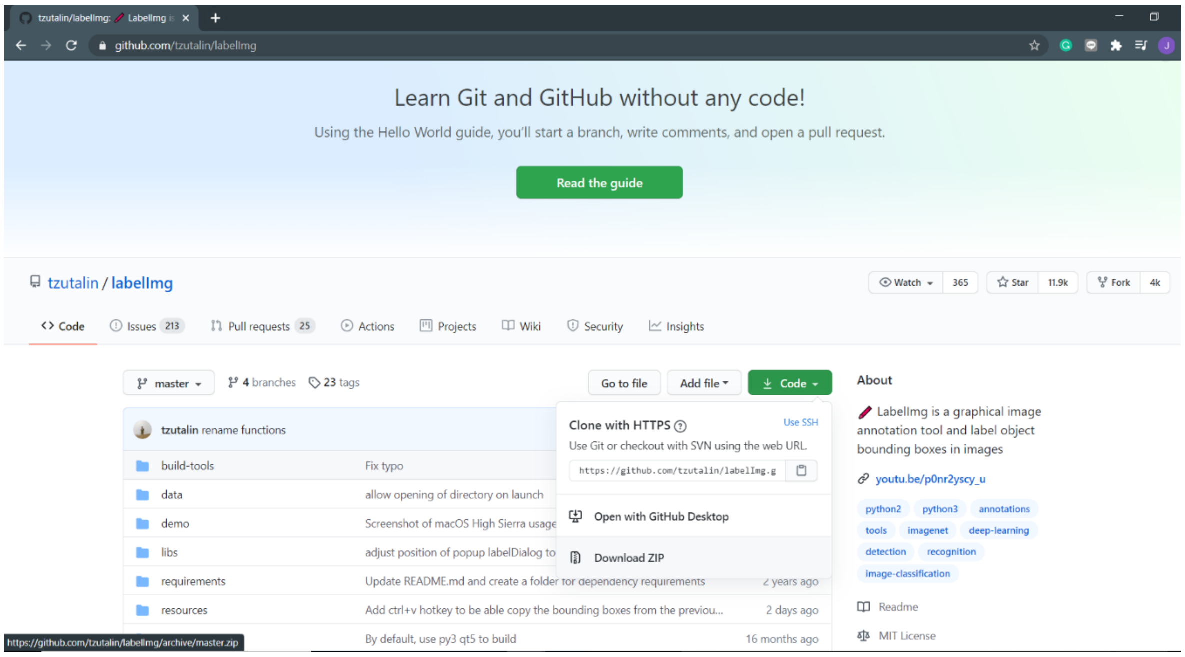
1. Install the suitable dependencies

Create a virtual environment by using the anaconda and install the dependencies based on the gpu.yml in my github link https://github.com/JJLim99/Implementation-of-TensorFlow-GPU-CUDA-in-Windows.git

D. Procedure

1. Preparation of dataset

In order to train a custom object detection model, it is suggested that you collect your dataset and separate them into train and validation folders based on the ratio 8:2. Then, annotating your dataset using labelImg to get their respective *.xml files. Open the following link https://github.com/tzutalin/labelImg to download the labelImg zip file and extract it to your preferable location.



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

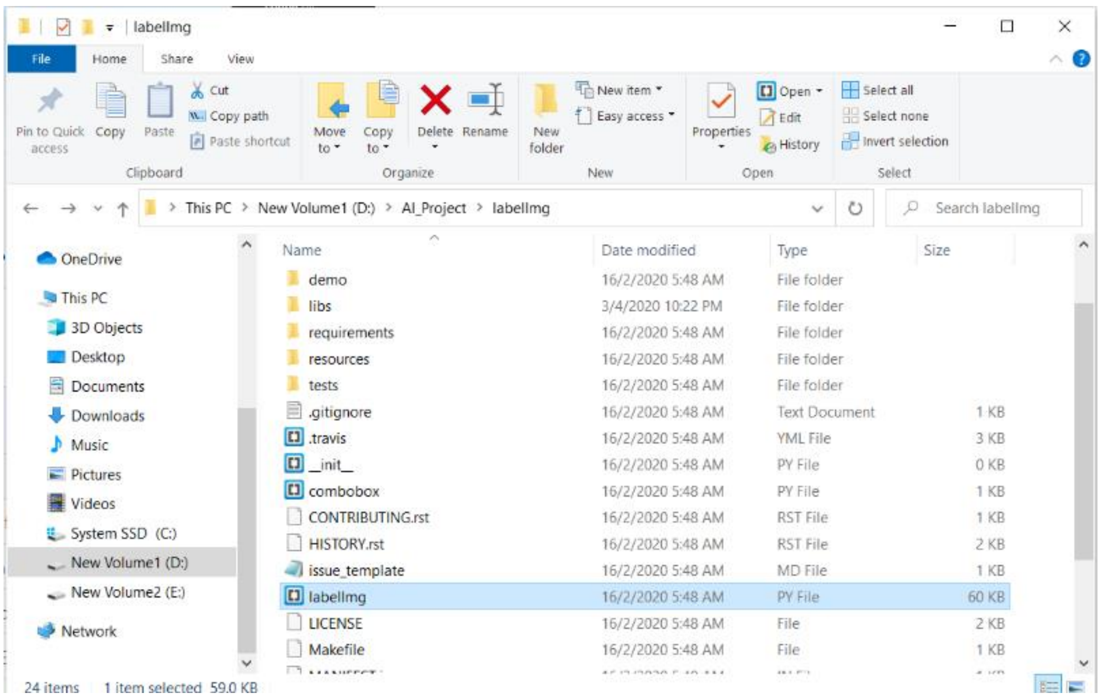
Use Git or checkout with SVN using the web URL.

LabelImg is a graphical image annotation tool and label object bounding boxes in images

tzutalin rename functions

build-tools	Fix typo	
data	allow opening of directory on launch	
demo	Screenshot of macOS High Sierra usage	
libs	adjust position of popup labelDialog to	
requirements	Update README.md and create a folder for dependency requirements	2 years ago
resources	Add ctrl+v hotkey to be able copy the bounding boxes from the previou...	2 days ago
	By default, use py3 qt5 to build	16 months ago

After opening the labelImg link, click the "Code", then click "Download ZIP" to download the labelImg zip file

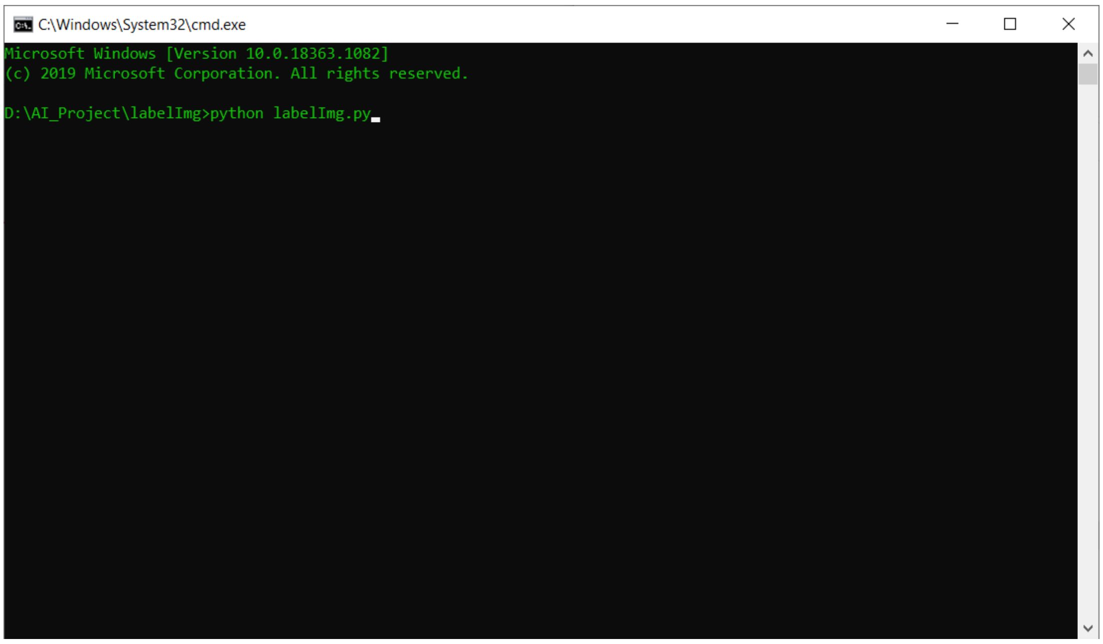


This PC > New Volume1 (D:) > AI_Project > labelImg

Name	Date modified	Type	Size
demo	16/2/2020 5:48 AM	File folder	
libs	3/4/2020 10:22 PM	File folder	
requirements	16/2/2020 5:48 AM	File folder	
resources	16/2/2020 5:48 AM	File folder	
tests	16/2/2020 5:48 AM	File folder	
.gitignore	16/2/2020 5:48 AM	Text Document	1 KB
.travis	16/2/2020 5:48 AM	YML File	3 KB
__init__	16/2/2020 5:48 AM	PY File	0 KB
combobox	16/2/2020 5:48 AM	PY File	1 KB
CONTRIBUTING.rst	16/2/2020 5:48 AM	RST File	1 KB
HISTORY.rst	16/2/2020 5:48 AM	RST File	2 KB
issue_template	16/2/2020 5:48 AM	MD File	1 KB
labelImg	16/2/2020 5:48 AM	PY File	60 KB
LICENSE	16/2/2020 5:48 AM	File	2 KB
Makefile	16/2/2020 5:48 AM	File	1 KB

24 items 1 item selected 59.0 KB

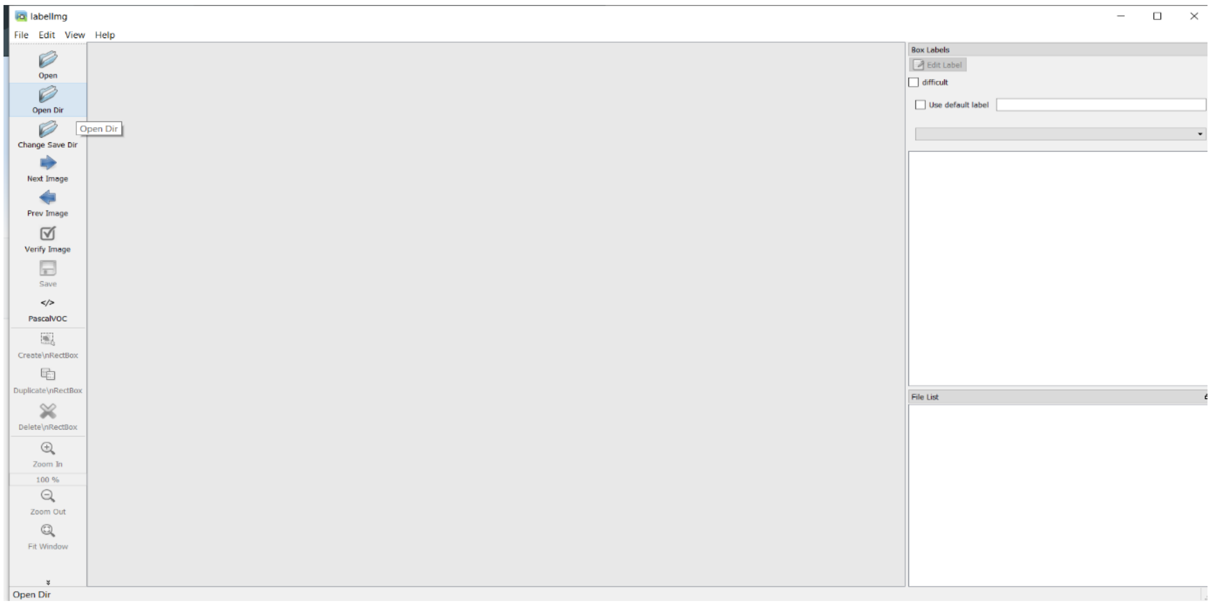
After extracting the zip file, you should see the list of files as shown in the above picture



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\AI_Project\labelImg>python labelImg.py
```

Go to command prompt, change your directory to the labelImg location, then opening labelImg by typing the command "python labelImg.py", then press the "enter" button



You will see the interface as shown in the above picture after opening the labelImg successfully. You can start annotating your dataset now.

- "Open Dir" is for you to open the directory of your dataset.
- "Change Save Dir" is for you to change the directory of your annotated dataset.
- "PascalVOC" will save the annotated file in *.xml format.
- "YOLO" will save the annotated file in *.json format.

# Since we are implementing the object detection using MaskRCNN, hence make sure you are annotating the dataset in *.xml format.

# YouTube link for the guidelines of using labelImg:
https://www.youtube.com/watch?v=VsZvT69Ssbs

2. Implementation of MaskRCNN
   - Go to your preferable path using command prompt, clone the main repository by typing the following command;
     git clone
     https://github.com/JJLim99/Tutorial_ObjectDetection_Keras_MaskRCNN.git
   - After that, preparing your dataset folder as shown in the README.md
   - Open ObjectDetection_Keras_MaskRCNN.ipynb by using Jupyter Notebook (If you have enough GPU memory for training as mentioned in the Minimum

Requirement section) or Google Colab (You can use the free 15GB GPU memory for a fixed time).

- Further instructions can be found in the ObjectDetection_Keras_MaskRCNN.ipynb. Please take note on the comment sections for updating the relevant parameters.

E. Result

By referring the instructions under the RealTime_Webcam section in README.md, you can test your object detection model via the video or webcam. It’s a real time system !!!