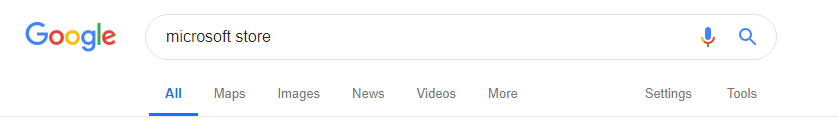
**#We will use Window to cover image related tasks, Linux to cover developing related tasks.**

**# ‘#’is just comment. Do not write in cmd. Just read it.**

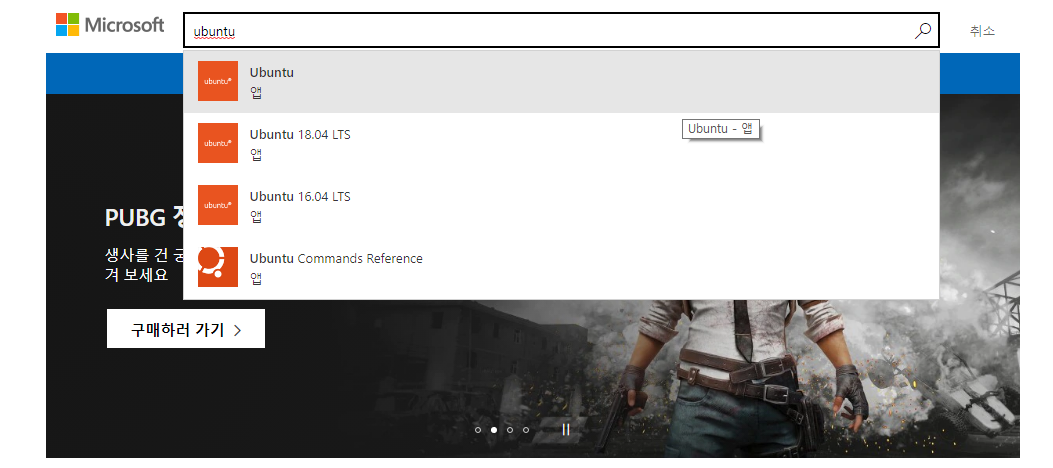
1. **set up environment(tensorflowCPUver)**

**WSL(window subsystem linux) install**

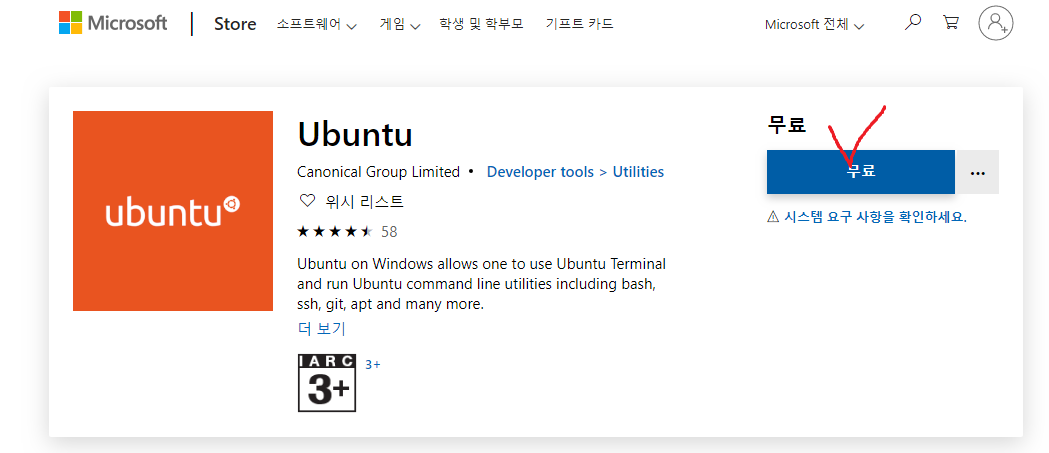
-Go to microsoft store.



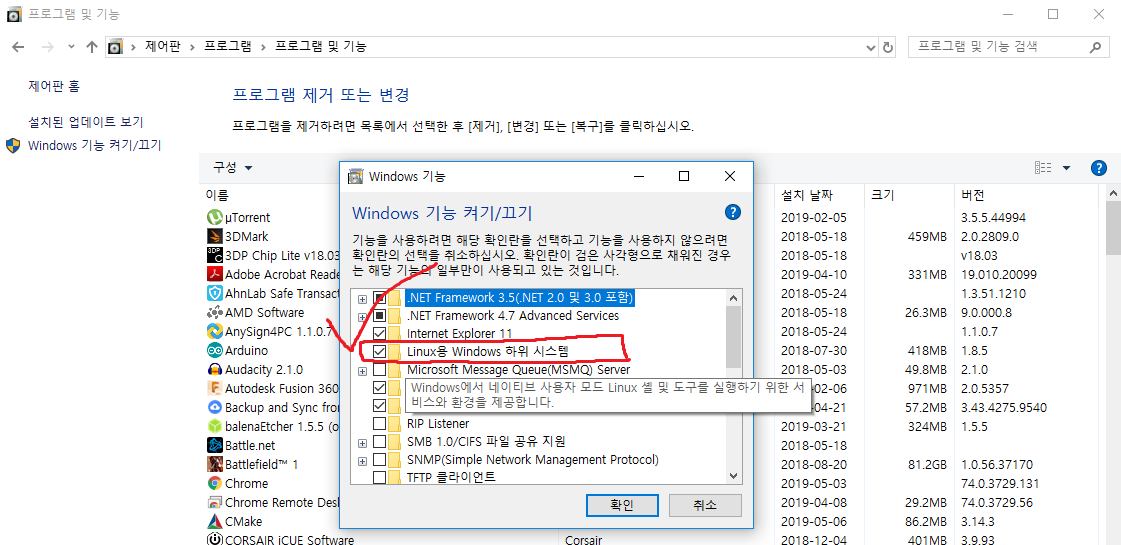
-search ‘ubuntu’ on microsoft store.



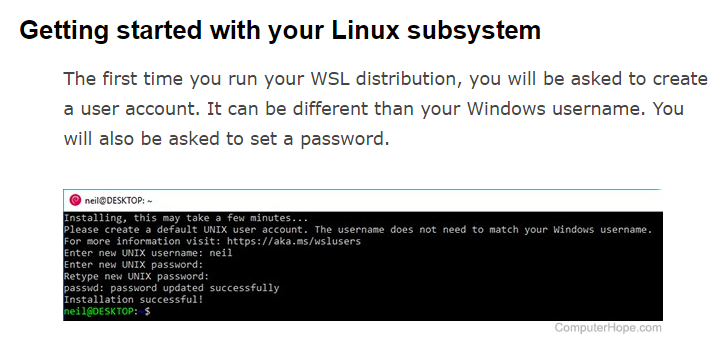
-install ubuntu



-go to ‘control panel’->’program’->’windows funtion on/off’->’window subsystem for linux’ and check for ‘window subsystem for linux’.



* Set up username,password.



1. **python2.x + tensorflow install**

$sudo apt-get update

$sudo apt upgrade

$sudo apt-get install python2.7

$sudo pip install tensorflow

1. **Python3.x + tensorflow install**

$sudo apt-get update

$sudo apt upgrade

$sudo apt-get install python3.6

$sudo pip3 install tensorflow

1. **Install necessary libraries and packages**

$ sudo apt-get install protobuf-compiler python-pil python-lxml python-tk

$ sudo pip3 install jupyter

$ sudo pip3 install matplotlib

$ sudo pip3 install Pillow

1. **Tensorflow object-detection model install**

$ mkdir work

$ mkdir ~/work/tensorflow

$ cd ~/work/tensorflow

$ git clone https://github.com/tensorflow/models

1. **protobuf compile**

$ cd ~/Downloads

$ wget <https://github.com/google/protobuf/releases/download/v3.3.0/protoc-3.3.0-linux-x86_64.zip>

$ unzip protoc-3.3.0-linux-x86\_64.zip -d ~/work/tensorflow/models/research/protoc-3.3.0

$ cd ~/work/tensorflow/models/research

$ protoc-3.3.0/bin/protoc object\_detection/protos/\*.proto --python\_out=.

1. **Add libraries to PYTHONPATH**

$ sudo vim ~/.bashrc

#bashrc 파일의 제일 아랫줄에 아래의 명령어 추가 후 저장 및 종료(=esc 후 :wq!)

export PYTHONPATH=$PYTHONPATH:~/work/tensorflow/models/research/:~/work/tensorflow/models/research/slim:~/work/tensorflow/models/research/object\_detection/

1. **Test after installation**

$ cd ~/work/tensorflow/models/research

$ sudo python3 setup.py build

$ sudo python3 setup.py install

$ cd slim

$ sudo python3 setup.py build

$ sudo python3 setup.py install

$ cd ~/work/tensorflow/models/research/object\_detection/

$ sudo python3 builders/model\_builder\_test.py

1. **Set up environment(tensorflowGPUver)**
2. **Install OS(ubuntu16.04)**

**Reference :** <https://cupjoo.tistory.com/53>

1. **Install CUDA toolkt 9.0&CUDNN 7.1 for CUDA 9.0 Reference :**<https://gist.github.com/Brainiarc7/470a57e5c9fc9ab9f9c4e042d5941a40>

**Reference :** <https://mc.ai/install-cudnn-7-1-install-on-ubuntu-18-04-and-cuda-9-0/>

# first get the PPA repository driver

$ sudo add-apt-repository ppa:graphics-drivers/ppa

# install graphic driver

$ sudo ubuntu-drivers list

$ sudo ubuntu-drivers autoinstall

$ sudo service lightdm restart

#reboot the computer after blink

#check whether installed

$ nvidia-smi

# install other import packages

$ sudo apt-get install g++ freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libglu1-mesa libglu1-mesa-dev

# CUDA 9 requires gcc 6

$ sudo apt install gcc-6

$ sudo apt install g++-6

# if there is a problem like ‘E: Unable to locate package gcc-6’, try this.

$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test

$ sudo apt-get update

$sudo apt-get install gcc-6 g++-6

# download one of the "runfile (local)" installation packages from cuda toolkit archive

$ wget https://developer.nvidia.com/compute/cuda/9.0/Prod/local\_installers/cuda\_9.0.176\_384.81\_linux-run

# make the download file executable

$ chmod +x cuda\_9.0.176\_384.81\_linux-run

$ sudo ./cuda\_9.0.176\_384.81\_linux-run --override

# Answer questions following while installation begin

# You are attempting to install on an unsupported configuration. Do you wish to continue? y

# Install NVIDIA Accelerated Graphics Driver for Linux-x86\_64 384.81? n

# Install the CUDA 9.0 Toolkit? y

# set up symlinks for gcc/g++

$ sudo ln -s /usr/bin/gcc-6 /usr/local/cuda/bin/gcc

$ sudo ln -s /usr/bin/g++-6 /usr/local/cuda/bin/g++

# setup your paths

$ echo 'export PATH=/usr/local/cuda-9.0/bin:$PATH' >> ~/.bashrc

$ echo 'export LD\_LIBRARY\_PATH=/usr/local/cuda-9.0/lib64:$LD\_LIBRARY\_PATH' >> ~/.bashrc

$ source ~/.bashrc

# install cuDNN v7.1

# in order to download cuDNN you have to regeistered here https://developer.nvidia.com/developer-program/signup

# then download cuDNN v7.1 form <https://developer.nvidia.com/cudnn>

$ wget <https://s3.amazonaws.com/open-source-william-falcon/cudnn-9.0-linux-x64-v7.1.tgz>

$ tar -xzvf [cudnn-9.0-linux-x64-v7.1.tgz](https://s3.amazonaws.com/open-source-william-falcon/cudnn-9.0-linux-x64-v7.1.tgz)

#copy the following files into the cuda toolkit directory.

$ sudo cp -P cuda/include/cudnn.h /usr/local/cuda-9.0/include

$ sudo cp -P cuda/lib64/libcudnn\* /usr/local/cuda-9.0/lib64/

$ sudo chmod a+r /usr/local/cuda-9.0/lib64/libcudnn\*

1. **Python3.x + tensorflow GPU version install**

$ sudo apt-get update

$ sudo apt upgrade

$ sudo apt-get install python3.6

$ pip3 install tensorflow-gpu==1.12.0

#Tensorflow version 1.12.0 is required (Cuda9.0+cudnn7.1+tesorflow1.12.0)

1. **Install necessary libraries and packages**

$ sudo apt-get install protobuf-compiler python-pil python-lxml python-tk

$ sudo pip3 install jupyter

$ sudo pip3 install matplotlib

$ sudo pip3 install Pillow

1. **Tensorflow object-detection model install**

$ mkdir work

$ mkdir ~/work/tensorflow

$ cd ~/work/tensorflow

$ git clone https://github.com/tensorflow/models

1. **protobuf compile**

$ cd ~/Downloads

$ wget <https://github.com/google/protobuf/releases/download/v3.3.0/protoc-3.3.0-linux-x86_64.zip>

$ unzip protoc-3.3.0-linux-x86\_64.zip -d ~/work/tensorflow/models/research/protoc-3.3.0

$ cd ~/work/tensorflow/models/research

$ protoc-3.3.0/bin/protoc object\_detection/protos/\*.proto --python\_out=.

1. **Add libraries to PYTHONPATH**

$ sudo vim ~/.bashrc

#bashrc 파일의 제일 아랫줄에 아래의 명령어 추가 후 저장 및 종료(=esc 후 :wq!)

export PYTHONPATH=$PYTHONPATH:~/work/tensorflow/models/research/:~/work/tensorflow/models/research/slim:~/work/tensorflow/models/research/object\_detection/

1. **Test after installation**

$ cd ~/work/tensorflow/models/research

$ sudo python3 setup.py build

$ sudo python3 setup.py install

$ cd slim

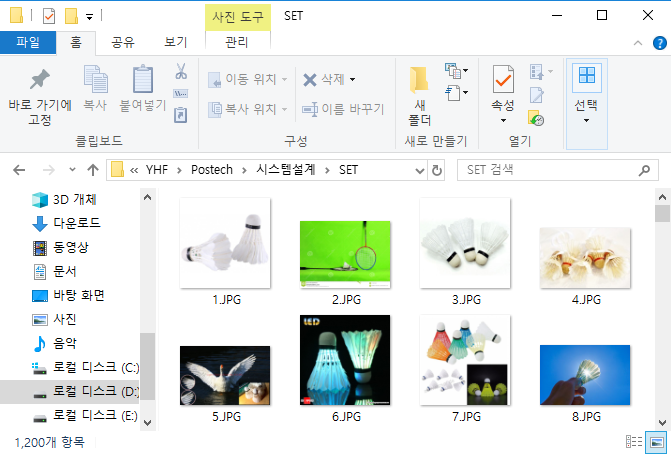
$ sudo python3 setup.py build

$ sudo python3 setup.py install

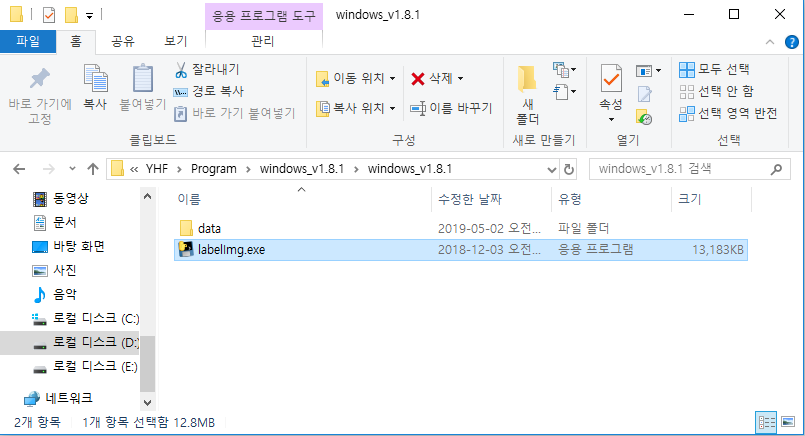
$ cd ~/work/tensorflow/models/research/object\_detection/

$ sudo python3 builders/model\_builder\_test.py

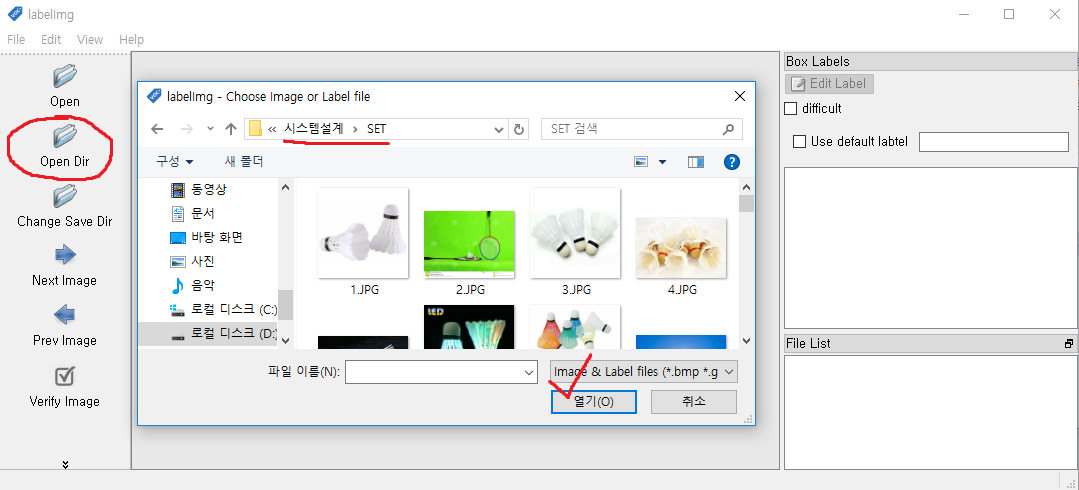
1. **Preparation for train dataset**
2. **Image file preparation (.png/.jpg) on Window.**

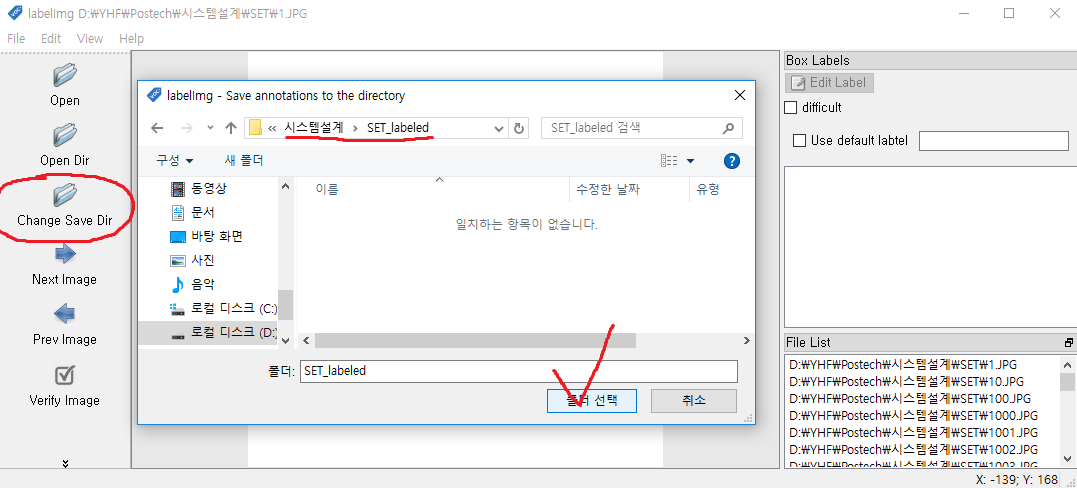


1. **Conversion from .png/.jpg to .xml using ‘labelImg’ program on Window.**

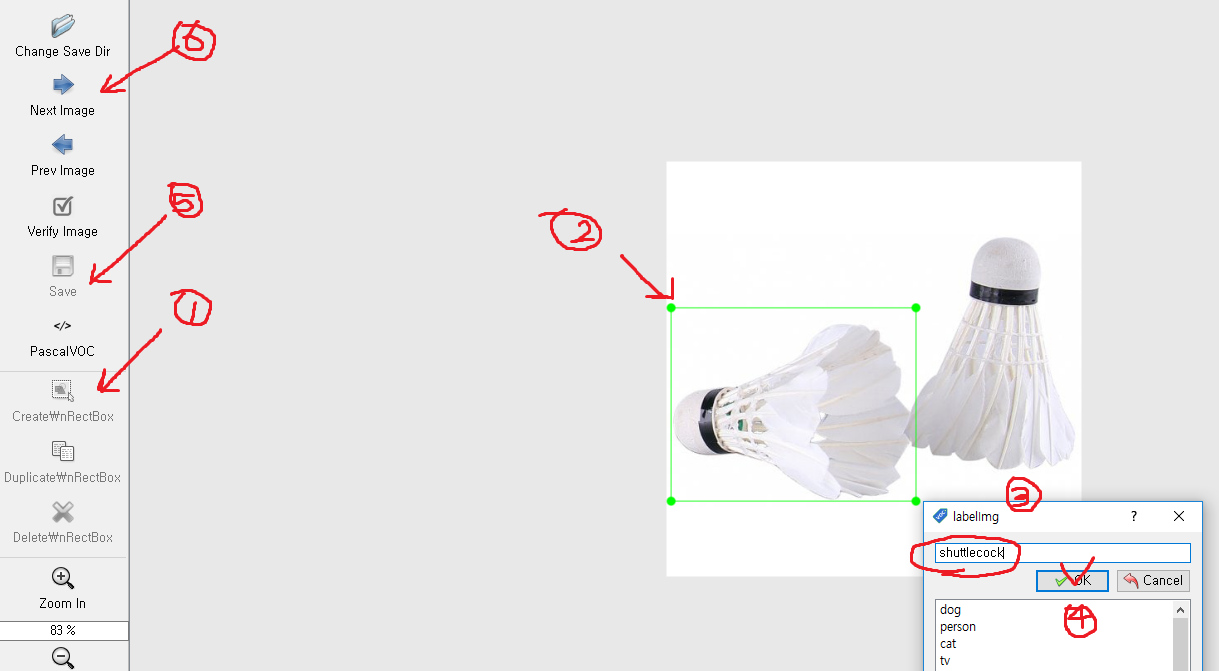
-install ‘labelImg.exe’ from [https://github.com/tzutalin/labelImg/releases](https://github.com/tzutalin/labelImg/releases)

-run ‘labelImg.exe’ and set ‘open dir’ and ‘save dir’ as you want..





-label your images. At step 3, set your class name.



1. **Conversion from .xml to .csv using ‘xml\_to\_csv.py’ program on Linux .**

$ cd ~/work/tensorflow/models/research/object\_detection/legacy

$ git clone <https://github.com/datitran/raccoon_dataset>

#in raccon\_dataset, there are several folders and files exists.

’anotations’ folder for .xml files(xml files converted from image files)

’data’ folder for .csv files.(csv files converted from csv files)

’images’ folder for .png/.jpg files.(images files to convert)

training for .config/.pb/.pbtxt files.(tensorflow model/setting files)

#we will change raccon\_dataset -> shuttlecock\_dataset

$ mv raccoon\_dataset shuttlecock\_dataset

$ cd shuttlecock\_dataset

$ rm -rf annotations

$ mkdir annotations

$ rm -rf data

$ mkdir data

$ rm -rf images

$ mkdir images

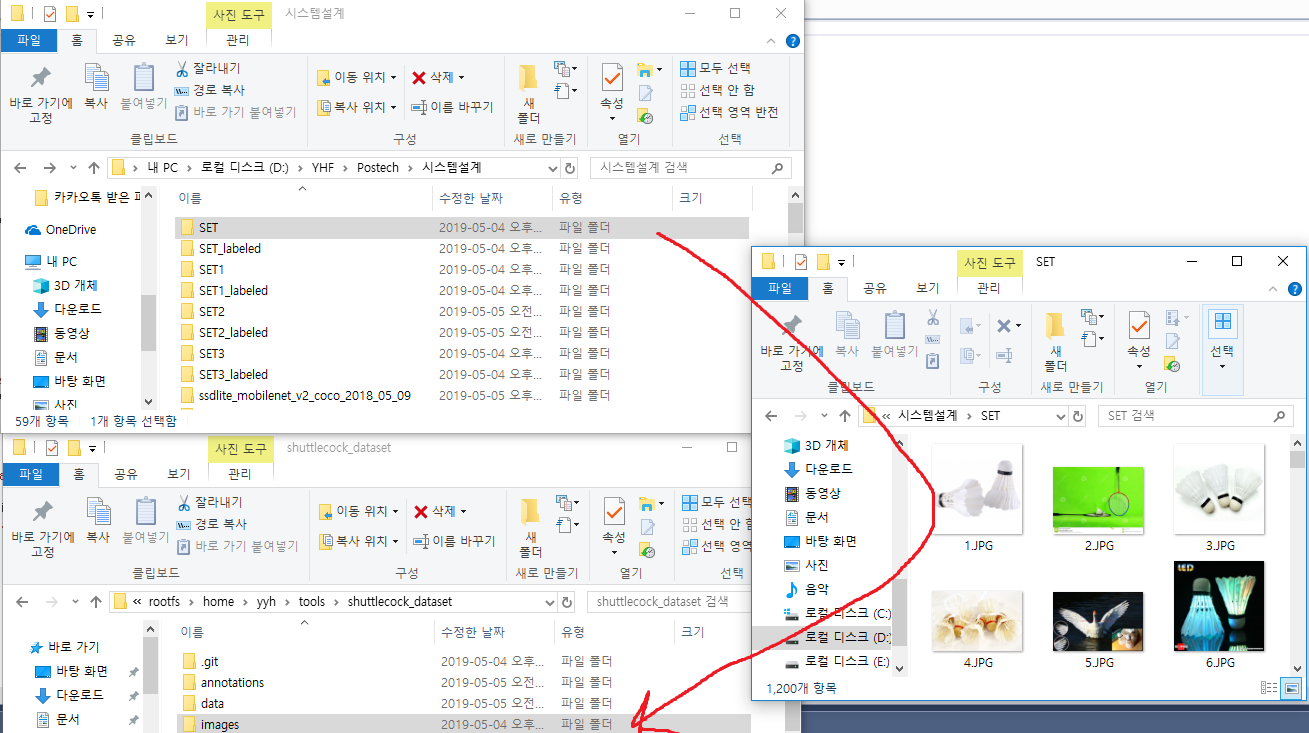
#copy your .png/.jpg files from Window directory to Linux directory(WSL) ‘images’

For me, Window directory for image files is

D:\YHF\Postech\시스템설계\SET

Linux directory is

C:\Users\YuYeounghun\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows\_79rhkp1fndgsc\LocalState\rootfs\home\yyh\work\tensorflow\models\research\object\_detection\legacy\shuttlecock\_dataset\images



**#After copy, close the terminal.**

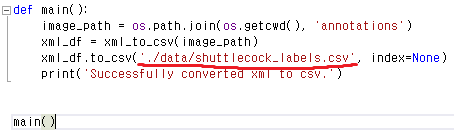
**#if you use WSL, Reopen new WSL terminal because you changed linux system from outside of linux system.**

$ cd ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset

$ sudo vim xml\_to\_csv.py

#change code

#Set your output file save directory and name of output file. Enter edit mode(=type ‘a’).

Use edit mode(=push ‘a’) and change code.

# code save and quit edit mode.(=push ‘esc’ + ‘:wq!’+’enter’)

$ pip3 install pandas

$ sudo python3 xml\_to\_csv.py

$ cd data

$ ls

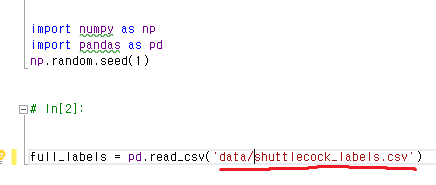
#check whether there is ‘shuttlecock\_label.csv’ or not

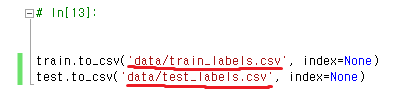
$ cd ..

$ jupyter nbconvert --to script 'split labels.ipynb'

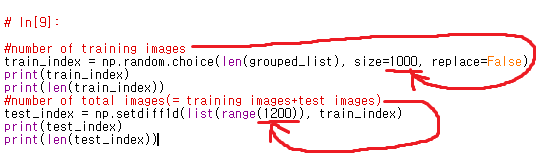
$ sudo vim ‘split labels.py’

#change input file name/path and output file name/path





#change number of training data/number of total images



$ sudo python3 ‘split labels.py’

$ cd data

$ ls

#check whether there is ‘train\_labels.csv’ and ‘test\_labels.csv’.

$ cd ..

1. **Conversion from .csv to .record using ‘generate\_tfrecord.py’ on Linux.**

**(.record is format used by tensorflow)**

$ sudo vim generate\_tfrecord.py

# change class code.



$ sudo python3 generate\_tfrecord.py --csv\_input=data/train\_labels.csv --output\_path=data/train.record --image\_dir=images

$ sudo python3 generate\_tfrecord.py --csv\_input=data/test\_labels.csv --output\_path=data/test.record --image\_dir=images

$ cd data

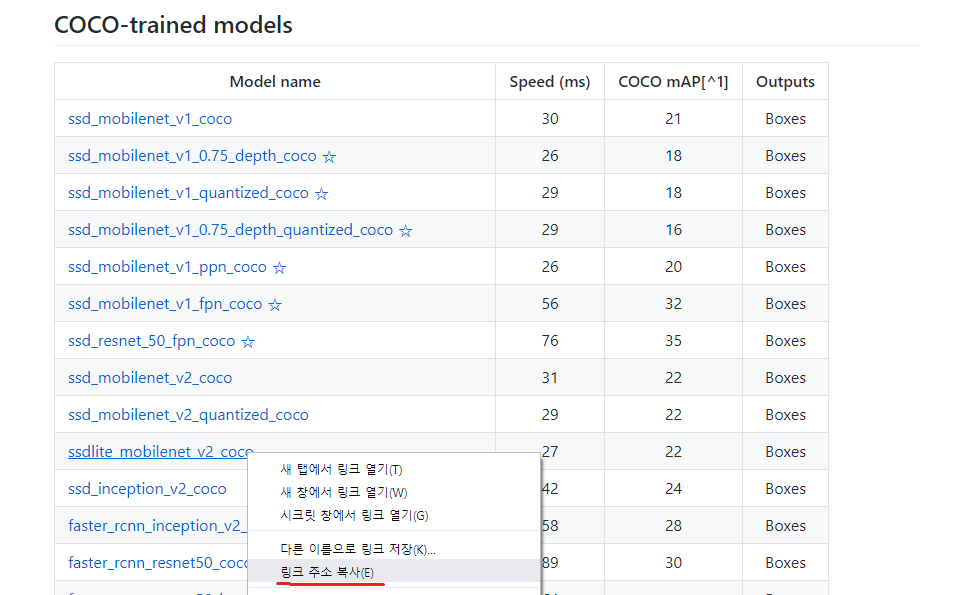
$ ls

#check whether there is ‘train.record’ and ‘test.record’.

$ cd ~

1. **Training**
2. **choose ‘pre-trained model’ on Window from website.**

<https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md>



Copy the link of your pre-trained model. In my case I will use ‘ssdlite\_mobilenet\_v2\_coco’

$ mkdir Downloads

$ cd Downloads

#you can paste your link by click right button. wget + ‘pasted link’

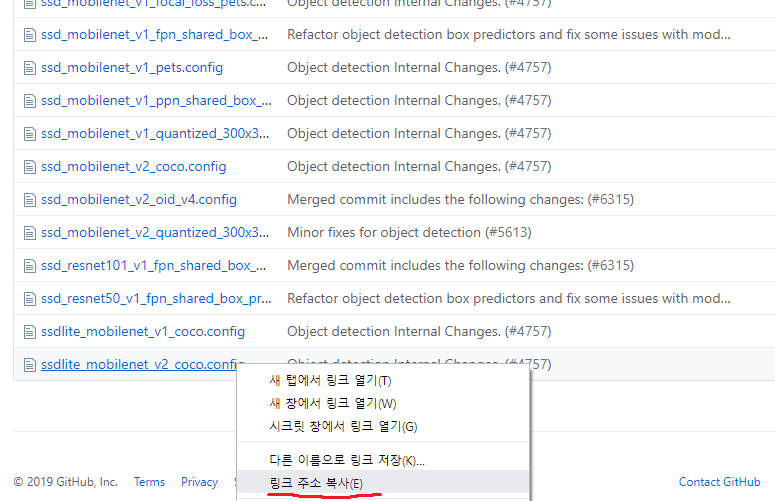
$ wget <http://download.tensorflow.org/models/object_detection/ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz>

$ ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset/training

$ tar -xzvf ~/Downloads/ssdlite\_mobilenet\_v2\_coco\_2018\_05\_09.tar.gz

1. **choose pre-trained model ‘config file’ on Window from website.**

<https://github.com/tensorflow/models/tree/master/research/object_detection/samples/configs>



Copy the link of your config file. In my case I will use ‘ssdlite\_mobilenet\_v2\_coco.config’

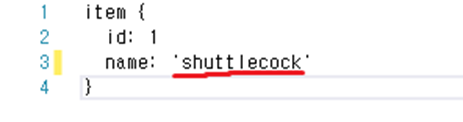
$ cd ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset/training

$ wget <https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/ssdlite_mobilenet_v2_coco.config>

1. **Edit ‘object-detection.pbtxt’**

$ sudo vim object-detection.pbtxt

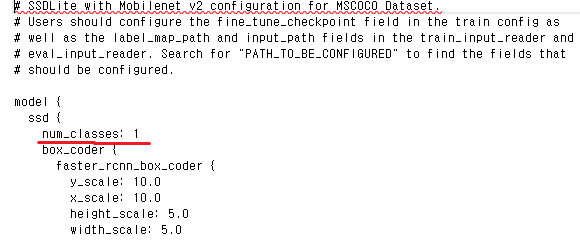
#change code according to your class name.



1. **Edit config file. For me, ‘ssdlite\_mobilenet\_v2\_coco.config’**

$ sudo vim ssdlite\_mobilenet\_v2\_coco.config

#change num\_classes.



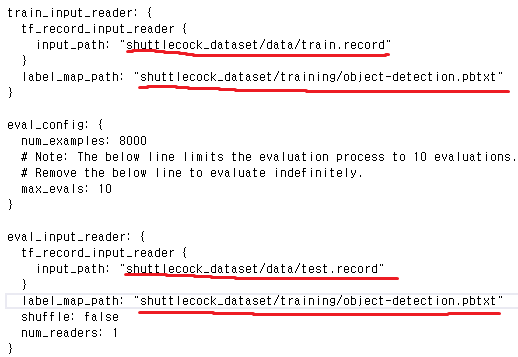
#change reference directories. Give file’s relative path from ‘train.py’

#if there is a problem, give the absolute path.

# ‘train.py’ is in this directory.

#~/work/tensorflow/models/research/object\_detection/legacy





1. **Train**

$ cd ~/work/tensorflow/models/research/object\_detection/legacy

#when libculos9.0 error occur try $sudo ldconfig /usr/local/cuda-9.0/lib64

$sudo python3 train.py --logtostderr --worker\_replicas=2 --train\_dir=shuttlecock\_dataset/training --pipeline\_config\_path=shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_coco.config

#Trained model will be saved in this directory.

#~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset/training

**6. Test**

$ cd ~/work/tensorflow/models/research/object\_detection/legacy

#for python unicode error, change every code ‘unicode’ to ‘str’ in .py file.

#for libcublas.so.9.0 import error, try command ‘sudo ldconfig /usr/loca/cuda-9.0/lib64’ or it probably CUDA 9.0 version is not operated/installed. Check version.

$ sudo python3 eval.py --logtostderr --checkpoint\_dir=shuttlecock\_dataset/training --eval\_dir=shuttlecock\_dataset/eval --pipeline\_config\_path=shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_coco.config

$cd ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset

$tensorboard --logdir=eval/

**\*\*\*\*\*For using Neural Compute Stick ver2\*\*\*\*\*\***

1. **Conversion from .ckpt/.pb (tensorflow model) to .xml/.bin (NCS2 openvino model)**
2. **Conversion from trained model to frozen model**

$ cd ~/work/tensorflow/models/research/object\_detection

$ python3 export\_inference\_graph.py --input\_type image\_tensor --pipeline\_config\_path legacy/shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_coco.config --trained\_checkpoint\_prefix legacy/shuttlecock\_dataset/training/model.ckpt-153000 --output\_directory legacy/shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_shuttlecock

1. **Install openvino toolkit for linux**

#Download openvino toolkit for linux on ~/downloads

$ tar -zxf l\_openvino\_toolkit\_p\_2019.1.133.tgz

$ cd l\_openvino\_toolkit\_p\_2019.1.133

$ sudo ./install.sh

$ sudo vim ~/.bashrc

#add this line at the last part of .bashrc and save

source /opt/intel/openvino/bin/setupvars.sh

$ source ~/.bashrc

$ cd /opt/intel/openvino/deployment\_tools/model\_optimizer/install\_prerequisites

$ sudo ./install\_prerequisites.sh

1. **Conversion from tensorflow model to NCS model using IR using meta graph**

$ cd /opt/intel/openvino/deployment\_tools/model\_optimizer

$ python3 mo\_tf.py --input\_model ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_shuttlecock/frozen\_inference\_graph.pb --tensorflow\_object\_detection\_api\_pipeline\_config ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_shuttlecock/pipeline.config --tensorflow\_use\_custom\_operations\_config /opt/intel/openvino/deployment\_tools/model\_optimizer/extensions/front/tf/ssd\_v2\_support.json --data\_type=FP16 --model\_name ssdlite\_mobilenet\_v2\_shuttlecock --output\_dir ~/work/tensorflow/models/research/object\_detection/legacy/shuttlecock\_dataset/training/ssdlite\_mobilenet\_v2\_shuttlecock

1. **Change past model to our new model in Raspberry pi.**

I saved .xml/.bin files to google drive on my computer and downloaded it on raspberry pi.

Change code of ‘SingleStickSSDwithRealsense\_OpenVINO\_NCS2.py’