yolo train

```
git clone https://github.com/puzzledqs/BBox-Label-Tool.git
```

convert.py

```
import os
from os import walk, getcwd
from PIL import Image

classes = ["targa"]

def convert(size, box):
    dw = 1./size[0]
    dh = 1./size[1]
    x = (box[0] + box[1])/2.0
    y = (box[2] + box[3])/2.0
    w = box[1] - box[0]
    h = box[3] - box[2]
    x = x*dw
    w = w*dw
    y = y*dh
    h = h*dh
    return (x,y,w,h)


"""-----------------------------------------------------------------------
"""

""" Configure Paths"""
mypath = "Labels/001/"
outpath = "Labels/output/"


cls = "001"


wd = getcwd()
list_file = open('%s/%s_list.txt'%(wd, cls), 'w')

""" Get input text file list """
txt_name_list = []
for (dirpath, dirnames, filenames) in walk(mypath):
    print(filenames)
    txt_name_list.extend(filenames)
    break
print(txt_name_list)

""" Process """
```

```
for txt_name in txt_name_list:
    # txt_file =  open("Labels/stop_sign/001.txt", "r")

    """ Open input text files """
    txt_path = mypath + txt_name
    print("Input:" + txt_path)
    txt_file = open(txt_path, "r")
    lines = txt_file.read().split('\r\n')   #for ubuntu, use "\r\n"
instead of "\n"

    """ Open output text files """
    txt_outpath = outpath + txt_name
    print("Output:" + txt_outpath)
    txt_outfile = open(txt_outpath, "w")


    """ Convert the data to YOLO format """
    ct = 0
    for line in lines:
        #print('lenth of line is: ')
        #print(len(line))
        #print('\n')
        if(len(line) >= 2):
            ct = ct + 1
            print(line + "\n")
            elems = line.split(' ')
            print(elems)
            cls_id = elems[0].split('\n')[0]
            xmin = elems[0].split('\n')[1]
            xmax = elems[2]
            ymin = elems[1]
            ymax = elems[3][:-1]
            #
            img_path = str('%s/Images/%s/%s.JPEG'%(wd, cls,
os.path.splitext(txt_name)[0]))
            #t = magic.from_file(img_path)
            #wh= re.search('(\d+) x (\d+)', t).groups()
            im=Image.open(img_path)
            w= int(im.size[0])
            h= int(im.size[1])
            #w = int(xmax) - int(xmin)
            #h = int(ymax) - int(ymin)
            # print(xmin)
            print(w, h)
            b = (float(xmin), float(xmax), float(ymin), float(ymax))
            bb = convert((w,h), b)
            print(bb)
            txt_outfile.write(str(cls_id) + " " + " ".join([str(a) for
a in bb]) + '\n')

    """ Save those images with bb into list"""
```

```
    if(ct != 0):
        list_file.write('%s/images/%s/%s.JPEG\n'%(wd, cls,
os.path.splitext(txt_name)[0]))

list_file.close()
```

Train.txt Text.txt

process.py

```
import glob, os

# Current directory
current_dir = os.path.dirname(os.path.abspath(__file__))

# Directory where the data will reside, relative to 'darknet.exe'
path_data = '*IMAGE DIRECTORY*'

# Percentage of images to be used for the test set
percentage_test = 10;

# Create and/or truncate train.txt and test.txt
file_train = open('train.txt', 'w')
file_test = open('test.txt', 'w')

# Populate train.txt and test.txt
counter = 1
index_test = round(100 / percentage_test)
for pathAndFilename in glob.iglob(os.path.join(current_dir, "*.JPEG")):
    title, ext = os.path.splitext(os.path.basename(pathAndFilename))

    if counter == index_test:
        counter = 1
        file_test.write(path_data + title + '.JPEG' + "\n")
    else:
        file_train.write(path_data + title + '.JPEG' + "\n")
        counter = counter + 1
```

Put images inside BBox-Label-Tool/Images/001/ convert to JPEG and delete old images

```
mogrify -format JPEG *.jpg
rm *.jpg
```

Go to main folder and run python main.py

```
python main.py
```

Write 001 inside Image Dir box and load

Create a label for each image

After that, exit and create a new directory inside Label

```
mkdir output
```

Run convert.py

```
python convert.py
```

Now create test.txt and train.txt with process.py

```
python process.py
```

```
├── Images (input)
│   ├── 001
│   │   ├── 20180319_113309.JPEG
│   └── targa
├── Labels (output)
│   ├── 001
│   │   ├── 20180319_113309.txt
│   └── output
│       ├── 20180319_113309.txt
```

Darknet

```
git clone https://github.com/pjreddie/darknet
cd darknet
make
```

Copy train.txt and test.txt inside darknet/cfg/

Create 3 files: obj.data obj.names obj.cfg

```
classes= *NUMBER CLASSES*
train  = *TRAIN DIRECTORY+
valid = *TEST DIRECTORY*
names = obj.names
backup = *BACKUP FOLDER*
```

```
*CLASS NAME*
```

Copy yolov2-tiny.cfg and change [region]:classes to classes = *NUMBER CLASSES* filters = (*NUMBER CLASSES* +5)*5

From:
https://wiki.csgalileo.org/ - **Galileo Labs**

Permanent link:
https://wiki.csgalileo.org/projects/plate

Last update: **2022/06/20 14:19**