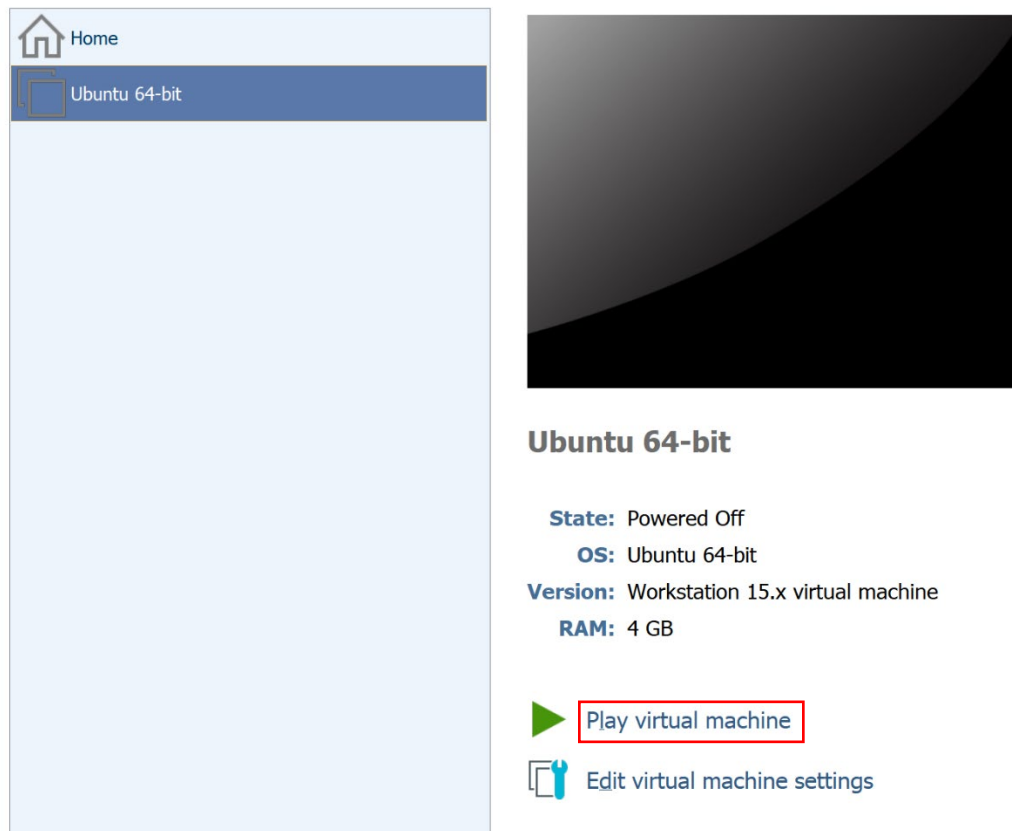


* Includes a fix for

- “Command “python setup.py egg_info” failed with error code 1 in ...”
- “ImportError: No module named modules.parser”
- “invalid int value: ‘./path2your_video’ ”
- ValueError when running yolo_video.py

Open VMWare Workstation and open the previous virtual machine



Open the terminal and download the repository for image classes

```
Kneron@ubuntu:~$ git clone https://github.com/pythonlessons/OIDv4_ToolKit.git
Cloning into 'OIDv4_ToolKit'...
remote: Enumerating objects: 439, done.
remote: Total 439 (delta 0), reused 0 (delta 0), pack-reused 439
Receiving objects: 100% (439/439), 34.09 MiB | 13.55 MiB/s, done.
Resolving deltas: 100% (155/155), done.
```

Go to OIDv3_ToolKit directory and download packages. Add sudo before pip3 will fix the “Command “python setup.py egg_info” failed with error code 1 in ...”

```
Kneron@ubuntu:~$ cd OIDv4_ToolKit/
Kneron@ubuntu:~/OIDv4_ToolKit$ sudo pip3 install -r requirements.txt
Kneron@ubuntu:~/OIDv4_ToolKit$ pip3 install lxml
```

Download 400 training images from Google Open Images Dataset V5. You can go to their website to choose a different object. Use python3 instead of python to fix the “ImportError: No module named modules.parser”

```
Kneron@ubuntu:~/OIDv4_ToolKit$ python3 main.py downloader --classes Binoculars --type_csv train --limit 400
```

Convert the images to XML file

```
Kneron@ubuntu:~/OIDv4_ToolKit$ python3 oid_to_pascal_voc_xml.py
```

```
Kneron@ubuntu:~/OIDv4_ToolKit$ gedit voc_to_YOLOv3.py
```

Edit the following

```

1 import xml.etree.ElementTree as ET
2 from os import getcwd
3 import os
4
5
6 dataset_train = 'OID/Dataset/train/'
7 dataset_file = '4_CLASS_test.txt'
8 classes_file = dataset_file[:-4]+'_classes.txt'
9
10
11 CLS = os.listdir(dataset_train)
12 classes=[dataset_train+CLASS for CLASS in CLS]
13 wd = getcwd()
14
15
16 def test(fullname):
17     bb = ""
18     in_file = open(fullname)
19     tree=ET.parse(in_file)
20     root = tree.getroot()
21     for i, obj in enumerate(root.iter('object')):
22         difficult = obj.find('difficult').text
23         cls = obj.find('name').text
24         if cls not in CLS or int(difficult)==1:
25             continue
26         cls_id = CLS.index(cls)
27         xmlbox = obj.find('bndbox')
28         b = (int(xmlbox.find('xmin').text), int(xmlbox.find('ymin').text), int(xmlbox.find('xmax').text),
29             int(xmlbox.find('ymax').text))
30         bb += (" " + ",".join([str(a) for a in b]) + ',' + str(cls_id))
31
32         # we need this because I don't know overlapping or something like that
33         if cls == 'Traffic_light':
34             list_file = open(dataset_file, 'a')
35             file_string = str(fullname)[:4]+' .jpg'+bb+'\n'
36             list_file.write(file_string)
37             list_file.close()
38             bb = ""
39
40     if bb != "":
41         list_file = open(dataset_file, 'a')
42         file_string = str(fullname)[:4]+' .jpg'+bb+'\n'
43         list_file.write(file_string)
44         list_file.close()
45
46
47 for CLASS in classes:
48     for filename in os.listdir(CLASS):
49         if not filename.endswith('.xml'):
50             continue
51         fullname = os.getcwd()+'/' +CLASS+'/' +filename
52         test(fullname)
53
54 for CLASS in CLS:
55     list_file = open(classes_file, 'a')
56     file_string = str(CLASS)+"\n"
57     list_file.write(file_string)
58     list_file.close()

```

Convert XML to Yolo v3 annotations. It should generate two txt files.

```

Kneron@ubuntu:~/OIDv4_ToolKit$ python3 voc_to_YOLOv3.py
Kneron@ubuntu:~/OIDv4_ToolKit$ ls
4_CLASS_test_classes.txt  classes.txt  LICENSE  modules  oid_to_pascal_voc_xml.py  requirements.txt
4_CLASS_test.txt          images      main.py  OID      README.md  voc_to_YOLOv3.py

```

Copy the txt files to keras-yolo3/model_data/

```
Kneron@ubuntu:~/OIDv4_ToolKit$ sudo cp -r 4_CLASS_test_classes.txt /home/Kneron/keras-yolo3/model_data
Kneron@ubuntu:~/OIDv4_ToolKit$ sudo cp -r 4_CLASS_test.txt /home/Kneron/keras-yolo3/model_data
```

```
Kneron@ubuntu:~/OIDv4_ToolKit$ cd ../keras-yolo3/
Kneron@ubuntu:~/keras-yolo3$ gedit train_bottleneck.py
```

Edit the following and increase the epoch for higher accuracy

```
16 def _main():
17     annotation_path = 'model_data/4_CLASS_test.txt'
18     log_dir = 'logs/000/'
19     classes_path = 'model_data/4_CLASS_test_classes.txt'
20     anchors_path = 'model_data/yolo_anchors.txt'
69     validation_steps=max(1, num_val//batch
70     epochs=50,
71     initial_epoch=10, max_queue_size=1)
72     model.save_weights(log_dir + 'trained_weights_
73
74     # train last layers with random augmented data
75     model.compile(optimizer=Adam(lr=1e-3), loss={
76         # use custom yolo_loss Lambda layer.
77         'yolo_loss': lambda y_true, y_pred: y_pred
78     batch_size = 16
79     print('Train on {} samples, val on {} samples,
80     model.fit_generator(data_generator_wrapper(lin
81         steps_per_epoch=max(1, num_train//batc
82         validation_data=data_generator_wrapper
83         validation_steps=max(1, num_val//batch
84         epochs=70,
85         initial_epoch=10,
86         callbacks=[logging, checkpoint])
87     model.save_weights(log_dir + 'trained_weights_
88
89     # Unfreeze and continue training, to fine-tune.
90     # Train longer if the result is not good.
91     if True:
92         for i in range(len(model.layers)):
93             model.layers[i].trainable = True
94     model.compile(optimizer=Adam(lr=1e-4), loss={'
change
95     print('Unfreeze all of the layers.')
96
97     batch_size = 4 # note that more GPU memory is
98     print('Train on {} samples, val on {} samples,
99     model.fit_generator(data_generator_wrapper(lin
100     steps_per_epoch=max(1, num_train//batch_si
101     validation_data=data_generator_wrapper(lin
102     validation_steps=max(1, num_val//batch_siz
103     epochs=100,
104     initial_epoch=60,
105     callbacks=[logging, checkpoint, reduce_lr
```

Start training

```
Kneron@ubuntu:~/keras-yolo3$ python3 train_bottleneck.py
```

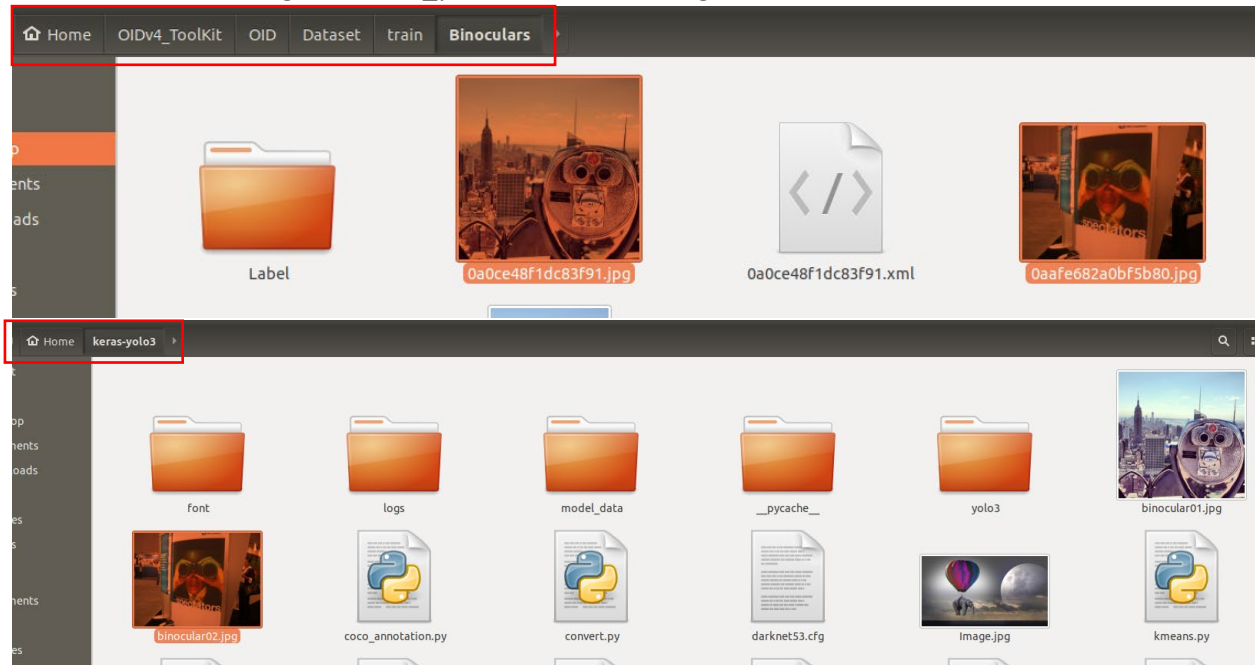
```
Kneron@ubuntu:~/keras-yolo3$ gedit yolo.py
```

```

22     _defaults = {
23         "model_path": 'logs/000/trained_weights_final.h5',
24         "anchors_path": 'model_data/yolo_anchors.txt',
25         "classes_path": 'model_data/4_CLASS_test_classes.txt',
26         "score" : 0.3,
27         "iou" : 0.45,
28         "model_image_size" : (416, 416),
29         "gpu_num" : 1,
30     }
31

```

Move some of the images to keras_yolo3 folder for testing



Change int to str to avoid the error "invalid int value: './path2your_video'"

```

Kneron@ubuntu:~/keras-yolo3$ gedit yolo_video.py

```

```

54     parser.add_argument(
55         "--input", nargs='?', type=str, required=False, default='./path2your_video',
56         help = "Video input path"
57     )
58

```

To fix ValueError when running yolo_video.py, make sure the object you want to detect is in the txt file

```

Kneron@ubuntu:~/keras-yolo3$ cd model_data/
Kneron@ubuntu:~/keras-yolo3/model_data$ sudo gedit 4_CLASS_test.txt

```

```

Open
1 Binoculars

```

```

Kneron@ubuntu:~/keras-yolo3$ python3 yolo_video.py --image

```

Enter the image name with extension when prompted


```
File Edit View Search Terminal Help
2020-12-02 17:34:55.591355: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency:
2304005000 Hz
2020-12-02 17:34:55.591781: I tensorflow/c
xecuting computations on platform Host. De
2020-12-02 17:34:55.591825: I tensorflow/c
(0): <undefined>, <undefined>
WARNING:tensorflow:From /home/Kneron/.loca
end.py:190: The name tf.global_variables i
ead.

WARNING:tensorflow:From /home/Kneron/.loca
end.py:74: The name tf.get_default_graph i
thead.

2020-
rning
. If
confi
t via
WARNI
end.p
rm in

WARNING
end.p
esize

logs/
WARNI
ps.py
d and
Instr
Use t
Input
(416,
2020-
led:
d 2 (
2020-
led:
d 2 (
Found
Binoculars 0.98 (378, 216) (622, 374)
1.6852283700009139
Input image filename:binocular02.jpg
(416, 416, 3)
Found 1 boxes for img
Binoculars 0.97 (378, 216) (622, 374)
0.4971851679983956
Input image filename:

ImageMagick: tmpvius_bbl.PNG
INTERNAP
Binoculars 0.97
spectators

ImageMagick: tmpp9_3bys2.PNG
Binoculars 0.98
```