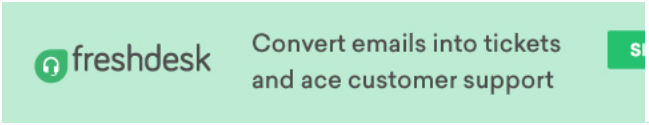
How to build a custom object detector using YOLOv3 in Python (/blog/yolov3-custom-object-detector)

2018-11-11 Taha Emara (/about/taha-emara)

Yolo (/categories/yolo) Computer Vision (/categories/computer-vision) Deep Learning (/categories/deeplearning) Opencv (/categories/opencv)



Introduction

After publishing the previous post How to build a custom object detector using Yolo (http://emaraic.com/blog/yolo-custom-object-detector), I received some feedback about implementing the detector in Python as it was implemented in Java. So, in this post, we will learn how to train YOLOv3 on a custom dataset using the Darknet framework and also how to use the generated weights with OpenCV DNN module to make an object detector. For this case, I collected a dataset for my Rubik's Cube to create a custom object detector to detect it.

Requirements:

1. Darknet, it is an open source neural network framework supports CPU and GPU computation
2. Anaconda or pip with python 3.5. In this tutorial, we will use anaconda, you can get it from here (https://www.anaconda.com/download/).
3. [Optional] In case you prefer using docker, you can build a docker image contains darknet and OpenCV 3.3 both are compiled with CUDA from this docker file https://github.com/tahaemara/yolo-custom-object-detector/tree/master/docker (https://github.com/tahaemara/yolo-custom-object-detector/tree/master/docker).
4. LabelImg (https://github.com/tzutalin/labelImg), it is an application to annotate the objects in images.
5. Source Code, you can get it from https://github.com/tahaemara/yolo-custom-object-detector (https://github.com/tahaemara/yolo-custom-object-detector/tree/master/python).
6. Webcam, I used ELP Sony IMX322 Sensor Mini USB Camera (https://www.amazon.com/gp/product/B071D3S4D4/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B071D3S4D4&linkCode=as2&tag=emaraic-20&linkId=94d84834a23397add5930efe168c8e46).

Dataset Collection and Annotating

I collected a dataset for my Rubik's Cube through my webcam with the size of (416x416) with different positions with different poses and scales to provided a reasonable accuracy. The next step is to annotate the dataset using LabelImg (https://github.com/tzutalin/labelImg) to define the location (Bounding box) of the object (Rubik's cube) in each image. Annotating process generates a text file for each image, contains the object class number and coordination for each object in it, as this format "(object-id) (x-center) (y-center) (width) (height)" in each line for each object. Coordinations values (x, y, width, and height) are relative to the width and the height of the image. I hand-labeled them manually with, it is really a tedious task.

The steps to annotate Rubik's cubes in images using LabelImg:

1. Create a folder contains images files and name it "images".
2. Create a folder contains annotations files and name it "labels". **Folders "images" and "labels" must be in the same directory.**
3. Open LabelImg application.
4. Click on "Open Dir" and then choose the Images folder.
5. Click on "Change Save Dir" and choose the labels folder.
6. Right below "Save" button in the toolbar, click "PascalVOC" button to switch to YOLO format.
7. You will find that all images are listed in the File List panel.
8. Click on the Image you want to annotate.
9. Click the letter "W" from your keyboard to draw the rectangle on the desired image object, type the name of the object on the popped up window.
10. Click "CTRL+S" to save the annotation to the labels folder.
11. Repeat steps 8 to 10 till you complete annotating all the images.

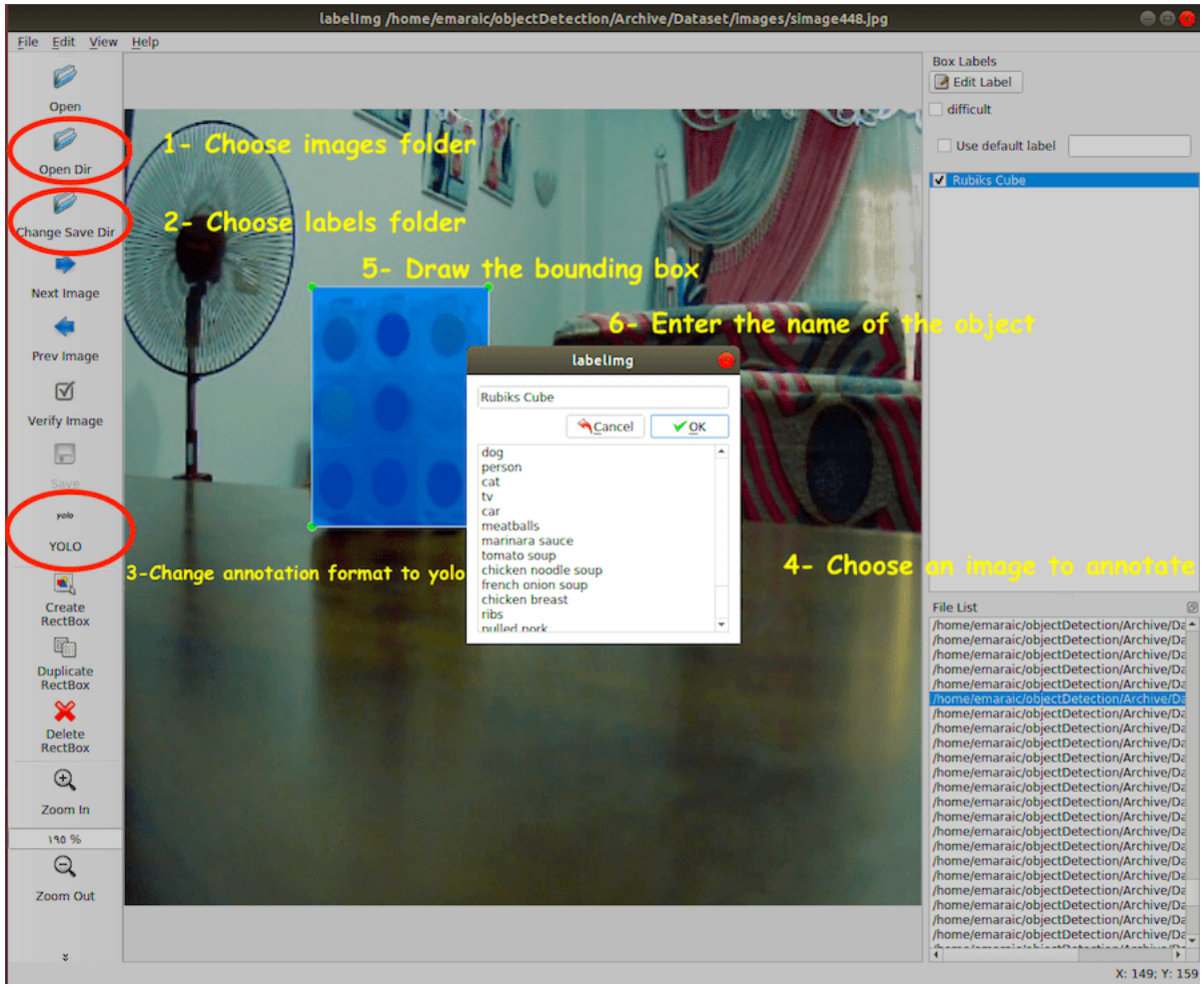


Figure 1: Annotation processes.

Installing Darknet

Darknet is an open source neural network framework supports CPU and GPU computation. You can follow the installation instructions darknet from the official website here (https://pjreddie.com/darknet/install/). In case you prefer using docker, I wrote a docker file by which you can build a docker image contains Darknet and OpenCV 3.3 both are compiled with CUDA. This docker file can be downloaded from https://github.com/tahaemara/yolo-custom-object-detector/tree/master/docker (https://github.com/tahaemara/yolo-custom-object-detector/tree/master/docker) .

Notes

If you try to install darknet with CUDA in Ubuntu 18.04 and face this problem "unsupported GNU version! gcc versions later than 6 are not supported!", just execute the following commands:

1. sudo apt-get install gcc-6 g++-6
2. sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-6 10
3. sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-6 10

Before the training process

Preparing training configuration files

- Generate train.txt and test.txt files

  After collecting and annotating dataset, we have two folders in the same directory the "images" folder and the "labels" folder. Now, we need to split dataset to train and test sets by providing two text files, one contains the paths to the images for the training set (train.txt) and the other for the test set (test.txt). This can be done using the following script [1] after editing the **dataset_path** variable to the location of your dataset folder. After running this script, the train.txt and test.txt files will be generated in the directory of this script.

```
import glob, os


dataset_path = '/path/to/Dataset/images'

# 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(dataset_path, "*.jpg")):
    title, ext = os.path.splitext(os.path.basename(pathAndFilename))

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

- Modify Cfg for our dataset

  We will need to modify the YOLOv3 tiny model (yolov3-tiny.cfg (https://github.com/pjreddie/darknet/blob/master/cfg/yolov3-tiny.cfg)) to train our custom detector. This modification includes:

  1. Uncomment the lines 5,6, and 7 and change the training batch to 64 and subdivisions to 2.
  2. Change the number of filters for convolutional layer "[convolution]" just before every yolo output "[yolo]" such that the number of filters= #anchors x (5 + #ofclasses)= 3x(5+1)= 18. The number 5 is the count of parameters center_x, center_y, width, height, and objectness Score. So, change the lines 127 and 171 to "filters=18".
  3. For every yolo layer [yolo] change the number of classes to 1 as in lines 135 and 177.

  Other files are needed to be created as "objects.names" which its name implies that it contains names of classes, and also the file "training.data" which contains parameters needed for training as described in the next table.

File Name	Description	Sample
objects.names	It contains the names of the classes. Also, the line number represents the object id in the annotations files.	`Rubiks Cube`

trainer.data	It contains :
	1. Number of classes.
	2. Locations of train.txt and test.txt files relative to the darknet main directory.
	3. Location of objects.names file relative to the darknet main directory.
	4. Location of the backup folder for saving the weights of training process, it is also relative to the darknet main directory.
		`classes= 1`
		`train  = custom/train.txt`
		`valid  = custom/test.txt`
		`names = custom/objects.names`
		`backup = backup/`

File Name	Description	Sample
yolov3-tiny.cfg	It contains the training parameters as batch size, learning rate, etc., and also the architecture of the network as number of layer, filters, type of activation function, etc.	

Download Pretrained Convolutional Weights

The main idea behind making custom object detection or even custom classification model is **Transfer Learning** which means reusing an efficient pre-trained model such as VGG, Inception, or Resnet as a starting point in another task. For training YOLOv3 we use convolutional weights that are pre-trained on Imagenet. We use weights from the darknet53 (https://pjreddie.com/darknet/imagenet/#darknet53) model. You can just download the weights for the convolutional layers here (76 MB) (https://pjreddie.com/media/files/darknet53.conv.74) and put it in the main directory of the darknet.

Start training

Before starting the training process we create a folder **"custom"** in the main directory of the darknet. Then we copy the files **train.txt, test.txt, objects.names, yolov3-tiny.cfg, and trainer.data inside the "custom" (https://github.com/tahaemara/yolo-custom-object-detector/tree/master/python/custom) folder** . After that, we start training via executing this command from the terminal

```
./darknet detector train custom/trainer.data custom/yolov3-tiny.cfg darknet53.conv.74
```

Notes

1. Weights will be saved in the backup folder every 100 iterations till 900 and then every 10000.
2. Kill the training process once the average loss is less than 0.06, or once the avg value no longer increases.

```
Region 23 Avg IOU: 0.880141, Class: 0.999959, Obj: 0.999842, No Obj: 0.000690, .5R: 1.000000, .75R: 1.000000,  count: 1498591: 0.033501, 0.063296 avg, 0.000010 rate, 0.004106 seconds, 498591 ima
Loaded: 0.003061 seconds
```

Making the Real-time detector

Installing OpenCV using Anaconda

1. Create conda environment with this command

```
conda create --name opencv python=3.5
```

2. Activate opencv environment with this command

```
source activate opencv
```

3. Install OpenCV with this command

```
conda install -c conda-forge opencv
```

or this one

```
conda install -c conda-forge/label/broken opencv
```

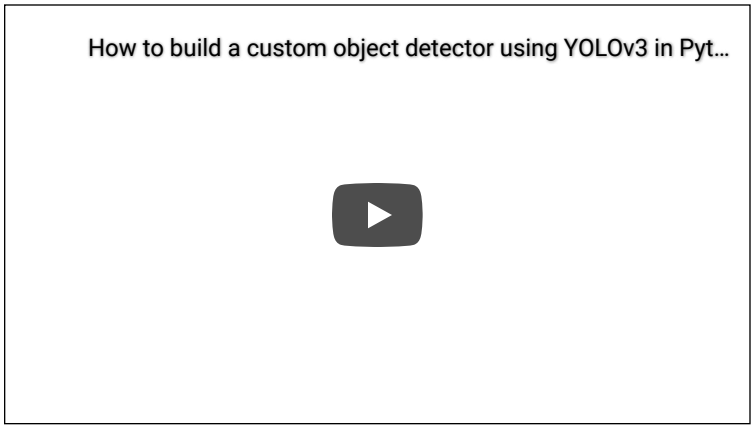
Real-time detector code

After installing OpenCV, just run the following code [2] on opencv environment by executing this command “python yolo_opencv.py -c /path/to/yolov3-tiny.cfg -w /path/to/yolov3-tiny_finally.weights -cl /path/to/objects.names”.

```
import cv2
import argparse
import numpy as np

ap = argparse.ArgumentParser()
ap.add_argument('-c', '--config',
                help = 'path to yolo config file', default='/path/to/yolov3-tiny.cfg')
ap.add_argument('-w', '--weights',
                help = 'path to yolo pre-trained weights', default='/path/to/yolov3-tiny_finally.weights')
ap.add_argument('-cl', '--classes',
                help = 'path to text file containing class names',default='/path/to/objects.names')
args = ap.parse_args()


# Get names of output layers, output for YOLOv3 is ['yolo_16', 'yolo_23']
def getOutputsNames(net):
    layersNames = net.getLayerNames()
    return [layersNames[i[0] - 1] for i in net.getUnconnectedOutLayers()]


# Darw a rectangle surrounding the object and its class name
def draw_pred(img, class_id, confidence, x, y, x_plus_w, y_plus_h):

    label = str(classes[class_id])

    color = COLORS[class_id]

    cv2.rectangle(img, (x,y), (x_plus_w,y_plus_h), color, 2)

    cv2.putText(img, label, (x-10,y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

# Define a window to show the cam stream on it
window_title= "Rubiks Detector"
cv2.namedWindow(window_title, cv2.WINDOW_NORMAL)


# Load names classes
classes = None
with open(args.classes, 'r') as f:
    classes = [line.strip() for line in f.readlines()]
print(classes)

#Generate color for each class randomly
COLORS = np.random.uniform(0, 255, size=(len(classes), 3))

# Define network from configuration file and load the weights from the given weights file
net = cv2.dnn.readNet(args.weights,args.config)

# Define video capture for default cam
cap = cv2.VideoCapture(0)

while cv2.waitKey(1) < 0:

    hasframe, image = cap.read()
    #image=cv2.resize(image, (620, 480))

    blob = cv2.dnn.blobFromImage(image, 1.0/255.0, (416,416), [0,0,0], True, crop=False)
    Width = image.shape[1]
    Height = image.shape[0]
    net.setInput(blob)
```

```
outs = net.forward(getOutputsNames(net))

class_ids = []
confidences = []
boxes = []
conf_threshold = 0.5
nms_threshold = 0.4


#print(len(outs))

# In case of tiny YOLOv3 we have 2 output(outs) from 2 different scales [3 bounding box per each scale]
# For normal normal YOLOv3 we have 3 output(outs) from 3 different scales [3 bounding box per each scale]

# For tiny YOLOv3, the first output will be 507x6 = 13x13x18
# 18=3*(4+1+1) 4 boundingbox offsets, 1 objectness prediction, and 1 class score.
# and the second output will be = 2028x6=26x26x18 (18=3*6)

for out in outs:
    #print(out.shape)
    for detection in out:

    #each detection  has the form like this [center_x center_y width height obj_score class_1_score class_2_score ..]
        scores = detection[5:]#classes scores starts from index 5
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:
            center_x = int(detection[0] * Width)
            center_y = int(detection[1] * Height)
            w = int(detection[2] * Width)
            h = int(detection[3] * Height)
            x = center_x - w / 2
            y = center_y - h / 2
            class_ids.append(class_id)
            confidences.append(float(confidence))
            boxes.append([x, y, w, h])

# apply  non-maximum suppression algorithm on the bounding boxes
indices = cv2.dnn.NMSBoxes(boxes, confidences, conf_threshold, nms_threshold)

for i in indices:
    i = i[0]
    box = boxes[i]
    x = box[0]
    y = box[1]
    w = box[2]
    h = box[3]
    draw_pred(image, class_ids[i], confidences[i], round(x), round(y), round(x+w), round(y+h))

# Put efficiency information.
t, _ = net.getPerfProfile()
label = 'Inference time: %.2f ms' % (t * 1000.0 / cv2.getTickFrequency())
cv2.putText(image, label, (0, 15), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0))

cv2.imshow(window_title, image)
```

Building a custom object detector using Yolo



Further Reading

- What's new in YOLO v3? (https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b).
- YOLOv3: An Incremental Improvement (https://pjreddie.com/media/files/papers/YOLOv3.pdf)
- How to implement a YOLO (v3) object detector from scratch in PyTorch: Part 1 (https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/).

References

- YOLO unity integration's training utils script from Here (https://github.com/tran-brian/yolo-unity-integration/blob/master/Training%20scripts/process.py).
- OpenCV object detection dnn example from Here (https://github.com/opencv/opencv/blob/3.4/samples/dnn/object_detection.py).

Subscribe to Our Mailing List

Email address

Subscribe

Share:

Like (https://facebook.com/sharer/sharer.php?u=http%3A%2F%2Femaraic.com%2Fblog%2Fyolov3-custom-object-detector)

Tweet (https://twitter.com/share?url=http%3A%2F%2Femaraic.com%2Fblog%2Fyolov3-custom-object-detector&text=In%20this%20post%2C%20we%20will%20learn%20how%20to%20train%20YOLOv3%20on%20a%20custom%20dataset%20using%20the%20Darknet%20framework%20and%20also%20how%20to%20use%20the%20generated%

+1 (https://plus.google.com/share?url=http%3A%2F%2Femaraic.com%2Fblog%2Fyolov3-custom-object-detector)

Share (https://www.linkedin.com/shareArticle?mini=true&url=http%3A%2F%2Femaraic.com%2Fblog%2Fyolov3-custom-object-detector)

Pin it (https://pinterest.com/pin/create/bookmarklet/?&url=http%3A%2F%2Femaraic.com%2Fblog%2Fyolov3-custom-object-detector&description=In%20this%20post%2C%20we%20will%20learn%20how%20to%20train%20YOLOv3%20on%20a%20custom%20dataset%20using%20the%20Darknet%20framework%20and%20also%20how%20to%20use%20the%20gene

E-mail (mailto:?subject=In%20this%20post%2C%20we%20will%20learn%20how%20to%20train%20YOLOv3%20on%20a%20custom%20dataset%20using%20the%20Darknet%20framework%20and%20also%20how%20to%20use%20the%20generated%20wei custom-object-detector)

72 Comments Emaraic

Login

Recommend 9 Tweet Share

Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

evolver • 4 months ago • edited

Hi Taha,

I trained Yolo over my 147 images over 1 class. After training I was running Real time Detector code to open webcam and test it but the python script threw this error when I tried executing. Please help.

Image size of training images from my webcam : 960 * 540

Traceback (most recent call last):
File "yolo_opencv.py", line 47, in <module>
net = cv2.dnn.readNet(args.weights,args.config)
cv2.error: OpenCV(4.0.0) /io/opencv/modules/dnn/src/darknet/darknet_io.cpp:507: error: (-215:Assertion failed) separator_index < line.size() in function 'ReadDarknetFromCfgStream'

Edit 1: I changed my image size to 416*416 and re-annotated it but the error remains the same!.

Thanks

2 • Reply • Share ›

Matthew Francis Skipworth ➚ evolver • 4 months ago

evolver,
I had this issue as well. What ended up fixing it for me was re-commenting lines 5, 6, and 7 in the .cfg file once you are done training. Hope this helps.

1 • Reply • Share ›

pimpwhippa ➚ Matthew Francis Skipworth • 4 months ago

hi Matthew,

thanks for sharing. Your solution does get rid of the error at line 47 for me. Instead, now I get another error at line 57! at least a bit of advancing!

• Reply • Share ›

pimpwhippa ➚ evolver • 4 months ago

hi Taha,

I have exactly the same error

File "custom/yolo_opencv.py", line 47, in <module>
net = cv2.dnn.readNet(args.weights,args.config)
cv2.error: OpenCV(3.4.4) /darknet/opencv/opencv-3.4.4/modules/dnn/src/darknet/darknet_io.cpp:507: error: (-215:Assertion failed) separator_index < line.size() in function 'ReadDarknetFromCfgStream'

you have any idea what's wrong? what about you Evolver?
Thanks

Reply • Share ›

Taha Emara Mod ➚ pimpwhippa • 4 months ago

Did you set the number of anchors to 6?. Tiny yolo uses 6 anchors instead of 9 anchors in case of noraml Yolo.
https://github.com/tahaemar...

Reply • Share ›

Show more replies

Fred Raducki • a month ago

Hi. your code with tiny yolo works like a charm for me BIT
what changes needs to be done on this script to use regular yolo instead of tiny yolo ?
By giving to the script : default='yolov3.cfg' and default='yolov3.weights' located in the proper folder it doesnt work out of the box.
As you mention in the code comment the regular yolo results have more dimensions in the array.
any help would be welcome. regards.

Reply • Share ›

zaitoun • a month ago

Hi,
thanks for your tutorial it is really informative.

I have a question about the data set how much Images do I need to get high accuracy of the detection object. btw I have one class single object.

100, 500 , 3000, 10,000.

thanks in advance

Reply • Share ›

New Bie to SQL • a month ago

Can I simply use Juypter Notebook to perform the operation?

Reply • Share ›

Abhinav Pradeep • a month ago

Hey Taha,
I trained the detector for 3 classes, I wanted to know if there was a source code for implementing this on a single image instead of a webcam feed. Currenlty the detector is unable to detect the images that I've trained it with. (No error messages)

Reply • Share ›

Maxime Lohya • a month ago • edited

Hello Taha,

Super post, well detailed and easy to follow. Yet I have an issue with the training part as the terminal displays this:
Loading weights from darknet.conv.74...Done!
Learning Rate: 0.001, Momentum: 0.9, Decay: 0.0005
Resizing
608

Floating point execution

I saw the error has already been encountered because of png images but mine are all jpg...

Thanks for your help!

Reply • Share ›

Taha Emara Mod → Maxime Lohya • a month ago

You are welcome,
Try to change all jpg extension to png -the extension of you files- in this script https://github.com/tahaemar...

Reply • Share ›

Maxime Lohya → Taha Emara • a month ago

That does not change the results, sadly...

Reply • Share ›

Mostafiz Hossain • 2 months ago

Hello Taha,

Amazing blog indeed. Thank you for explaining it, really helped.

Can you please explain what is 'yolov3-tiny_finally.weights' ? will it be automatically save once I start the training processes?

Cheers,

Reply • Share ›

Taha Emara Mod → Mostafiz Hossain • a month ago

Thanks. "yolov3-tiny_finally.weights" is a file contains the weights of network after training or fine tuning. Yes it will be save automatically at each hundredth iteration until iteration 900. After that, the next set of saved weights will be at iteration 10,000, 20,000, and so on.

Reply • Share ›

okok • 2 months ago

You can share with me file yolov3-tiny_finally.weights

Reply • Share ›

Taha Emara Mod → okok • 2 months ago

https://drive.google.com/fi...

Reply • Share ›

Norhan • 2 months ago

Hi Taha,

Sorry for bothering, really appreciate your help. Is there a way i can run "yolo_opencv.py" without using anaconda? i did sudo pip install opencv but then i got that error

"blob = cv2.dnn.blobFromImage(image, 1.0/255.0, (416,416), [0,0,0], True, crop=False)
cv2.error: OpenCV(4.0.0) /io/opencv/modules/imgproc/src/resize.cpp:3784: error: (-215:Assertion failed) !ssize.empty() in function 'resize'"

Reply • Share ›

Taha Emara Mod → Norhan • 2 months ago

This error is due to having image with size zero, check the size of the image before this statement. Check this solution to process only the valid frame

Reply • Share ›

Norhan • 3 months ago

Hi Taha,

For yolo training can i use drone-net.weight (it's a trained weight file on github for drones) instead of the darknet53.conv?
So after downloading the file the command will be
./darknet detector train custom/trainer.data custom/yolov3-tiny.cfg drone-net.weights

i tried to run this but it seems that no backup files are being generated.
Reply • Share ›

Norhan ➚ Norhan • 3 months ago
i would really really appreciate if you answer my question
Reply • Share ›

Taha Emara Mod ➚ Norhan • 3 months ago
It is better to use darknet53 weights as it was trained on imagenet which is larger than the drones dataset. I realy don't know if it works, but if it works, it will be better to use darknet53 weights.
Reply • Share ›

Norhan • 3 months ago
Hey taha! This is an excellent tutorial! I just have a question, do you can if i can run two trainings at the same time (different terminals and different darknet folders)?
Reply • Share ›

Taha Emara Mod ➚ Norhan • 3 months ago
Of course, yes.
Reply • Share ›

Bruce Shaohan Wang • 3 months ago
hey, what is the file type for object.names and trainer.data?
i keep getting the message ofr . : couldn't open file:custom/trainer.data
Reply • Share ›

Bruce Shaohan Wang ➚ Bruce Shaohan Wang • 3 months ago
and this error when try to use yoloopencv :FileNotFoundError: [Errno 2] No such file or directory: '/path/to/objects.names'
Reply • Share ›

Taha Emara Mod ➚ Bruce Shaohan Wang • 3 months ago
try to replace the default path from "path /path/to/objects.names" to the real path such as "/home/user/darknet/custom/objects.names"
Reply • Share ›

Show more replies

Sburns • 3 months ago
Thank you for the great tutorial! Could you explain how to update the yolov3 model to include the new weights found from the training step and how the class object file is also updated to yolov3? I want to do this without using the real time web cam example you gave!
Reply • Share ›

Taha Emara Mod ➚ Sburns • 3 months ago
I do not understand you question.
Reply • Share ›

New_User • 3 months ago
I was implementing object detection of monitor screens using coco dataset. I have referred (https://github.com/AlexeyAB..., https://github.com/pjreddie... and also learn opencv and i got the training and testing done but was not accurate as it was overfitting and hence decided to setup threshold level for further processing. I have then come across your tutorial which i found really useful and have used the python

done but was not accurate as it was overfitting and hence decided to setup threshold level for further processing. I have then come across your tutorial which I found really useful and have used the python code. When i run this, my camera has to be at far distance (is there a way to zoom out)??

Reply • Share ›

Taha Emara Mod ➚ New_User • 3 months ago

Try to collect training data that has the objects you want to detect with the same (or nearly) scale, in which, it will be appeared at testing time. On other words, don't expect that when you train a model with images contain small cat to work correctly on big cats, your data must contain all cases -scales- of the object, except it works for youas sometimes it may work.

Reply • Share ›

New_User ➚ Taha Emara • 3 months ago

ok..I will also try to add new data and then try

Can u suggest something on camera settings?

Reply • Share ›

Show more replies

Rabee • 3 months ago

Hi Taha
Thank You very much for the tutorial , I have done the same steps you mentioned in the tutorial ,but did not detect any thing , I even used your dataset and show it one of the images it was trained on .

thank you again .

Reply • Share ›

Taha Emara Mod ➚ Rabee • 3 months ago

You are welcome.
Did you get any error messgaes?

Reply • Share ›

Lek Kurdthongmee • 4 months ago

Hi Taha,

I have got this error:

Loading weights from darknet53.conv.74...Done!
Learning Rate: 0.001, Momentum: 0.9, Decay: 0.0005
Resizing
608
Floating point exception: 8

Any idea?

Reply • Share ›

pimpwhippa ➚ Lek Kurdthongmee • 4 months ago

Try to increase the subdivision here in yolov3-tiny.cfg file value from 16 to 32 or 64.

i just copy the answer of Taha below for you. sawaddee.

Reply • Share ›

Lek Kurdthongmee ➚ pimpwhippa • 4 months ago

Previously, I followed the answer without success. Finally, I found the cause of this error. The SplitTrainAndTest.py only supports images with jpg format. In my case that caused failure during the training, I used images with png format. This created a blank train and test set file (train.txt and test.txt). Currently, I switch to images with jpg format and everything seems to work correctly.

Thanks.

Reply • Share ›

wang sherry • 4 months ago

Thanks for this tutorial,and I follow these step by step.I encounter a problem,the avg loss is about 30, it didnot decrease anymore. I lower the learning rate from 0.001 to 0.0001,it not use Can you let know what may be the problem.

Reply • Share ›

Taha Emara Mod ➚ wang sherry • 4 months ago

Try to increase the dataset and check this issue about batch and subdivision configuration https://github.com/pjreddie... .

Reply • Share ›

wang sherry ➚ Taha Emara • 4 months ago • edited

thank you for your advice,my dataset number is 250. I use CPU,and make with opencv=0 and cuda=0, the batch=2 and subdivision=1,the value of batch and subdivision cannot be increased. this must be done on gpu?

Reply • Share ›

Gordon Keil • 5 months ago

Can this tutorial be done on Windows 10?

Reply • Share ›

Taha Emara Mod ➚ Gordon Keil • 4 months ago

Take a look at this repo https://github.com/AlexeyAB

Reply • Share ›

Gordon Keil ➚ Taha Emara • 4 months ago

So after installing AlexeyAB darknet onto my computer, does all of your code go into the darknet folder that I have installed?

Reply • Share ›

Show more replies

Евгений • 5 months ago • edited

Hi, Taha. Thanks for this tutorial. I want to configure the model to work with only one class. At your github I see what lables.txt looks like **0** *0.780048 0.355769 0.430288 0.610577*. But my lables.txt looks like **15** *0.415000 0.408405 0.150000 0.256466* and at folder lables also file appears **classes.txt** with list of default LabelImg classes. My class is listed in the 16th position, so instead of 0, I have 15 at the start of lables.txt. Can you tell me how to remove these classes and leave only mine? Or I can leave it and everything will work?

Reply • Share ›

Taha Emara Mod ➚ Евгений • 5 months ago

Thanks. About this problem, you can remove classes which exist in this file ../labelimg-master/data/predefined_classes.txt and then restart the labelimg.

Reply • Share ›

Евгений ➚ Taha Emara • 5 months ago

Thanks so much, it works!

Reply • Share ›

Show more replies

Jain Rahul • 5 months ago

Hi,
I trained it for 11 classes with yolo-spp weights and now using the same script for prediction and apparently it is not giving any output. Can you let know what may be the problem.

Thanks

Reply • Share ›

Taha Emara Mod ➚ Jain Rahul • 5 months ago

The common case of this problem is the mismatch between objects in images and their corresponding boxes in text files.
I recommend you to check this issue https://github.com/pjreddie....

Reply • Share ›

Antonio • 5 months ago

Hi when training only one class my output on Class: 0.999959(your example) is always 0.000000 even after 50.000 training steps, should I train more or probably I did something wrong? Thanks in advance.

Reply • Share ›

Taha Emara Mod ➚ Antonio • 5 months ago

Hello.
Class value is the average of the probabilities of the true positives, so this means that you may have an issue in your data. Try to check the match between objects in images and their corresponding boxes in text files.

Reply • Share ›

This comment was deleted.

Load more comments

ALSO ON **EMARAIC**

Emaraic - Realtime Driver Drowsiness Detection (Sleep Detection)

19 comments • 2 years ago

Avatar **Jihen Hammedi** — in fact I solved the problem with "-lblas" but another undefined symbol of "dgesdd" appears, I tried with openblas but no result is obtained ... in fact …

End to End Voice Encryption between Raspberry pi and PC using AES

5 comments • 3 years ago

Avatar **matrix89** — Hi, if i want to use the sound card both for mic and speaker, how can i do it? It works (for speaker) only on integrated soundcard

Emaraic - How to use Weka in your Java code ?

1 comment • 2 years ago

Avatar **Shounak Ray** — Where exactly are you getting the file paths from for the model? Any why is the destination model location configured as a .bin instead of a .model?

Emaraic - Object recognition using TensorFlow and Java

15 comments • 2 years ago

Avatar **Rafi** — I'm trying to load my own model (output_graph.pb and output_labels.txt) that i produced from train flower_photos using python as mention in tensorflow …

Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy

Subscribe

Email address

Subscribe

Recent Posts

(/blog/yolov3-custom-object-detector)
(/blog/yolov3-custom-object-detector)
(/blog/yolov3-custom-object-detector)How to build a custom object detector using YOLOv3 in Python (/blog/yolov3-custom-object-detector)
(/blog/yolov3-custom-object-detector)

(/blog/yolo-custom-object-detector)
(/blog/yolo-custom-object-detector)
(/blog/yolo-custom-object-detector)How to build a custom object detector using Yolo (/blog/yolo-custom-object-detector)
(/blog/yolo-custom-object-detector)

(/blog/realtime-sudoku-solver)

(/blog/realtime-sudoku-solver)
(/blog/realtime-sudoku-solver)Real-time Sudoku Solver (/blog/realtime-sudoku-solver)

(/blog/realtime-sudoku-solver)

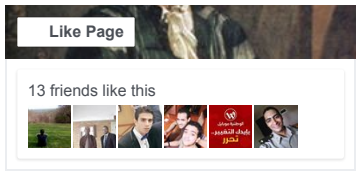
Blog Tags

Java (/categories/java) Database (/categories/database) Raspberry Pi (/categories/raspberry-pi) Internet of Things (/categories/iot) Security (/categories/security) Programming (/categories/programming)
Scheduling Algorithms (/categories/scheduling-algorithms) Machine Learning (/categories/machine-learning) Opencv (/categories/opencv) Dlib (/categories/dlib) Deeplearning4j (/categories/deeplearning4j)
Deep Learning (/categories/deeplearning) Computer Vision (/categories/computer-vision) Yolo (/categories/yolo)

Follow me

Facebook Page

Emaraic
2.1K likes



2016 © All Rights Reserved. Privacy Policy (/privacy-policy)