

# Table of contents

1. Creating appropriate dataset
2. Configure network and training
3. How to use this network

# Create dataset

for create a training dataset we need two parts , data and ground truth. The data refers to the objects or information which we should make a decision or election according to that. for example in this model our data is pictures which contain objects, we need to specify and the ground truth means true answers which uses to compare system answers and feedback if it specifies correctly or not and change itself to be more correctly.

In this problem we have pictures as training data and just we have to create the ground truth. Ground truth are save in annotation text file that we learn you how to create below.

The annotation text files are contain pictures directories and names , objects coordinates and objects classes like below:

```
"picture directory" "x_min","y_min","x_max","y_max","class number"  
"x_min1","y_min1","x_max1","y_max1","class number1"
```

Pay attention that put each picture information in a separate line and never use any spaces between numbers which you added , spaces is just using to separate directions and objects from the other objects.

The other file which you have to build is classes.txt which contain all the classes which you specified for the objects in annotation file , you have to insert each class name in a row and they should have the same priorities as the class numbers like this:

aeroplane  
bicycle  
bird  
boat  
bottle  
bus  
car  
cat  
chair  
cow  
diningtable  
dog  
horse  
motorbike  
person  
pottedplant  
sheep  
sofa  
train  
tvmonitor

(we have create a python application which help you to automatically create annotation file )

F:\yolo\VOCdevkit\VOC2007/JPEGImages/000012.jpg 156,97,351,270,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000017.jpg 185,62,279,199,14 90,78,403,336,12  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000023.jpg 9,230,245,500,1 230,220,334,500,1 2,1,117,369,14 3,2,243,462,14 225,1,334,486,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000026.jpg 90,125,337,212,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000032.jpg 104,78,375,183,0 133,88,197,123,0 195,180,213,229,14 26,189,44,238,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000033.jpg 9,107,499,263,0 421,200,482,226,0 325,188,411,223,0  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000034.jpg 116,167,360,400,18 141,153,333,229,18  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000035.jpg 1,96,191,361,14 218,98,465,318,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000036.jpg 27,79,319,344,11  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000042.jpg 263,32,500,295,18 1,36,235,299,18  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000044.jpg 1,1,370,330,8 99,101,312,213,7  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000047.jpg 15,1,459,312,6 439,170,487,230,8  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000048.jpg 58,107,291,465,2 2,1,302,500,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000061.jpg 274,11,437,279,3 184,214,281,252,3  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000064.jpg 1,23,451,500,2  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000066.jpg 209,187,228,230,14 242,182,274,259,14 269,188,295,259,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000073.jpg 121,143,375,460,15 2,154,64,459,15 270,155,375,331,3 22,143,146,500,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000077.jpg 240,1,500,333,7 2,1,306,333,7  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000078.jpg 15,75,475,412,11 94,41,437,238,11  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000083.jpg 109,33,447,309,6 380,2,435,135,14 138,14,193,115,14 1,20,75,204,14 60,48,114,156,14 32,51,83,162,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000089.jpg 20,7,183,355,14 98,215,429,374,14 332,140,455,366,14 22,51,317,291,8  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000091.jpg 125,241,195,284,6 81,248,105,268,6 91,250,126,274,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000104.jpg 403,110,500,265,19 291,75,442,333,14 2,7,283,333,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000112.jpg 70,174,277,328,11 287,162,316,198,8  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000122.jpg 36,20,430,474,7  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000129.jpg 70,202,255,500,1 74,1,272,462,14 252,19,334,487,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000133.jpg 73,18,406,500,14 135,227,338,500,12  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000134.jpg 1,115,487,330,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000138.jpg 6,56,154,285,14 145,58,283,258,14 224,54,440,332,14 363,96,500,332,14  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000140.jpg 107,146,386,300,11 2,113,500,401,17  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000141.jpg 25,53,418,399,2  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000147.jpg 33,105,65,146,8 77,114,114,149,8 146,125,175,150,8 373,121,402,153,8 435,112,465,149,8  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000153.jpg 237,147,358,191,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000154.jpg 59,76,367,266,3  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000159.jpg 234,48,286,124,14 1,16,498,333,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000161.jpg 104,34,446,390,6 68,195,121,288,6  
F:\yolo\VOCdevkit\VOC2007/JPEGImages/000162.jpg 306,227,380,299,19 196,143,309,369,14

An example of annotation file format

# Configure network and training

1. First of all , we have to convert “yolo3.weights” which you can see in yolo3 directory to a weight file compatible with keras for that you can use this command in cmd:

```
python convert.py -w yolov3.cfg yolov3.weights model_data/yolo_weights.h5
```

2. Now you have to start training but before that rename annotation.txt to train.txt and put it in directory which ‘train.py’ is stored.

Now you just need to run this command to start training:

```
python train.py --classes “classes file direction”
```

If training started correctly you have see this picture:

```
Create YOLOv3 model with 9 anchors and 20 classes.
C:\ProgramData\Anaconda3\envs\notebook\lib\site-packages\keras\engine\topology.py:3473: UserWarning: Skipping loading of weights for layer conv2d_59 due to mismatch in
shape ((1, 1, 1024, 75) vs (255, 1024, 1, 1)).
    weight_values[i].shape))
C:\ProgramData\Anaconda3\envs\notebook\lib\site-packages\keras\engine\topology.py:3473: UserWarning: Skipping loading of weights for layer conv2d_59 due to mismatch in
shape ((75,) vs (255,)).
    weight_values[i].shape))
C:\ProgramData\Anaconda3\envs\notebook\lib\site-packages\keras\engine\topology.py:3473: UserWarning: Skipping loading of weights for layer conv2d_67 due to mismatch in
shape ((1, 1, 512, 75) vs (255, 512, 1, 1)).
    weight_values[i].shape))
C:\ProgramData\Anaconda3\envs\notebook\lib\site-packages\keras\engine\topology.py:3473: UserWarning: Skipping loading of weights for layer conv2d_67 due to mismatch in
shape ((75,) vs (255,)).
    weight_values[i].shape))
C:\ProgramData\Anaconda3\envs\notebook\lib\site-packages\keras\engine\topology.py:3473: UserWarning: Skipping loading of weights for layer conv2d_75 due to mismatch in
shape ((1, 1, 256, 75) vs (255, 256, 1, 1)).
    weight_values[i].shape))
C:\ProgramData\Anaconda3\envs\notebook\lib\site-packages\keras\engine\topology.py:3473: UserWarning: Skipping loading of weights for layer conv2d_75 due to mismatch in
shape ((75,) vs (255,)).
    weight_values[i].shape))
Load weights model_data/yolo_weights.h5.
Freeze the first 249 layers of total 252 layers.
```

# How to use network

To use this network you just simply run this command:

```
python yolo_video.py --image
```

and you have see this:

```
Using TensorFlow backend.  
Image detection mode  
Ignoring remaining command line arguments: ./path2your_video,  
2019-02-26 13:35:11.200233: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX  
model_data/yolo.h5 model, anchors, and classes loaded.  
Input image filename:
```

Now just insert a picture you want to check direction in this format :

D://picture.jpg

After a while it shows you the result.

Have a good time!