this is darkflow implementation of yolo

and model is downloaded from github link- https://github.com/thtrieu/darkflow

overview of files and folders-

folder bin contain pretrained weight of some models downloaded from internet.

you can download weights from here also - https://pjreddie.com/darknet/yolo

folder cfg contain different congiguration files of yolo.

folder ckpt contain the weights of our trained model.

folder dataset contain our data

predict\_img.py file is there to predict the objects in an input object based on our trained model.

you may need to change the few things like path of the input image,

and name of model, load in option segment.

process\_video.py can be used to detect objects in a video which is not relevent to this project.

commands-

to train the model.

python flow --model cfg/tiny-yolo-voc-12c.cfg --load bin/tiny-yolo-voc.weights --train --annotation dataset/train\_annotation --dataset dataset/train\_images --gpu 1.0

and if you want use any specific trainer like adam you can add in it, --trainer adam

training saves all the weights learnt in the ckpt folder

so when you want test the model just use the model which you performed training and checkpoint number(profile number) saved in ckpt folder

python flow --imgdir dataset/test\_images --model cfg/tiny-yolo-voc-12c.cfg --load ckpt/4000 --gpu 1.0

before running this command create a folder 'out' in your test\_images folder because bydefault it will store the result there in the image format.

add the --json command if you want the output in json format instead the images.

inside the dataset folder.....

train\_images contain training images

train\_annotation contain training annotation

train\_a contain some annotation which need to preprocess in which name of object's name should be convert from

small sofa to small\_sofa i.e. space should be replcae by \_ or alternatively change name of objects given to model.

train\_i contain the sorrosponding images of such annotation.

test\_annotation and test\_images contain for testing purpose.

test\_annotation is for checking accuracy.

mAP-master is used for measuring accuracy.

move the test\_annotation file into ground truth folder.

go to directory extra and run command - python convert\_gt\_xml.py

move the json files received during the testing of model from out folder in test\_images to predicted folder

in mAP-master. and run command - python convert\_pred\_darkflow\_json.py

then come to the extra folder and run command - python main.py

it will predict accuracy.

for more understanding visit- https://github.com/thtrieu/darkflow