Requirements

```
sudo apt update
sudo apt install python3-pip
sudo apt install libopencv-dev

#INSTALAR cuda-toolkit-----
#Base Installer
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
/cuda-ubuntu2204.pinsudo mv cuda-ubuntu2204.pin
/etc/apt/preferences.d/cuda-repository-pin-600wget
https://developer.download.nvidia.com/compute/cuda/12.5.0/local_installers
/cuda-repo-ubuntu2204-12-5-local_12.5.0-555.42.02-1_amd64.debsudo dpkg -i
cuda-repo-ubuntu2204-12-5-local_12.5.0-555.42.02-1_amd64.debsudo cp
/var/cuda-repo-ubuntu2204-12-5-local/cuda-*-keyring.gpg
/usr/share/keyrings/sudo apt-get updatesudo apt-get -y install cuda-
toolkit-12-5

#Driver Installer
sudo apt-get install -y nvidia-driver-555-open
sudo apt-get install -y cuda-drivers-555

#INSTALAR cuDNN
#Base Installer
wget
https://developer.download.nvidia.com/compute/cudnn/9.2.0/local_installers
/cudnn-local-repo-ubuntu2204-9.2.0_1.0-1_amd64.deb
sudo dpkg -i cudnn-local-repo-ubuntu2204-9.2.0_1.0-1_amd64.deb
sudo cp /var/cudnn-local-repo-ubuntu2204-9.2.0/cudnn-*-keyring.gpg
/usr/share/keyrings/
sudo apt-get update
sudo apt-get -y install cudnn
```

```
pip3 install tensorflow
```

Usar a Open Images Dataset

Usando o OIDv4 Toolkit

Clonar ferramenta OIDv4

```
git clone https://github.com/EscVM/OIDv4_ToolKit.git
```

```
cd OIDv4_ToolKit/
```

Instalar dependências

```
pip3 install -r requirements.txt
```

Baixar imagens de treino pela ferramenta

Baixar imagens de treino

```
python3 main.py downloader --classes Box Coffee_cup Computer_mouse --
type_csv train --limit 500 --multiclasses 1
```

Obs: Se apresentar :

```
    [INFO] | Downloading train images.
    [INFO] | [INFO] Found 18525 online images for train.
    [INFO] | Limiting to 500 images.
    [INFO] | Download of 500 images in train.
sh: 1: aws: not found
  0%|                                                                                |
0/500 [00:00<?, ?it/s]sh: 1: aws: not found
sh: 1: aws: not found
sh: 1: aws: not found
sh: 1: aws: not found
sh: 1: aws: not found
sh: 1: aws: not found
sh: 1: aws: not found
sh: 1: aws: not found
```

Instale a AWS CLI:

```
sudo apt-get update
sudo apt-get install awscli
```

Verifique a instalação:

```
aws --version
```

Baixar imagens de teste

```
python3 main.py downloader --classes Box Coffee_cup Computer_mouse --
type_csv test --limit 100 --multiclasses 1
```

Converter arquivos de anotações

Mudar a classes no arquivo txt

/OIDv4_ToolKit/classes.txt

```
cat classes.txt
echo -e "Box\nCoffee cup\nComputer mouse" > classes.txt
```

Baixar repositório de ferramentas de coversão de anotações

```
git clone -n
https://github.com/Hemilibeatriz/TreinamentoCustomizadoYOLO.git
cd TreinamentoCustomizadoYOLO/
```

```
git checkout HEAD converter_anotacoes.py
mv converter_anotacoes.py ../
```

Edição dos arquivos de configuração

```
git clone https://github.com/AlexeyAB/darknet
cd darknet/
```

Configurar Makefile

```
nano Makefile
```

Certifique-se de que as seguintes opções estão definidas como `1`:

```
GPU=1
CUDNN=1
OPENCV=1
```

Compilar o darknet

```
make clean
make -j8
```

Modificação do .cfg

```
cp cfg/yolov4.cfg ../yolov4_custom.cfg
```

yolov4_custom.cfg

```
[net]
batch=64
subdivisions=64
# Training
#width=512
#height=512
width=608
height=608
channels=3
momentum=0.949
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.0013
burn_in=1000
#2000 * classes =  2000 8 3 = 6000
max_batches = 6000
policy=steps
#80%/90% do valor do max_batches
steps=4800,5400
scales=.1,.1

```

```
[convolutional]
size=1
stride=1
pad=1
#(classes + 5) * 3
filters=24
activation=linear


[yolo]
mask = 0,1,2
anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146,
classes=3
num=9
jitter=.3
ignore_thresh = .7
```

in /darknet

```
touch obj.names
touch obj.data
```

depois

```
touch obj.names
touch obj.data
cp obj.* ../
```

Gerar arquivos de train.txt e test.txt

```
cd OIDv4_ToolKit/
mkdir data
cp -r OID/Dataset/train/Box_Coffee\ cup_Computer\ mouse/ ./data/obj
cp -r OID/Dataset/test/Box_Coffee\ cup_Computer\ mouse/ ./data/valid
```

Baixar ferramenta de geração de treino e teste

```
cd TreinamentoCustomizadoYOLO/
git checkout HEAD gera_train.py
git checkout HEAD gera_test.py
mv gera_t* ../
```

Executar ferramenta de geração de treino e teste

```
python3 gera_train.py
python3 gera_teste.py

python3 gera_train.py
python3 gera_test.py
```

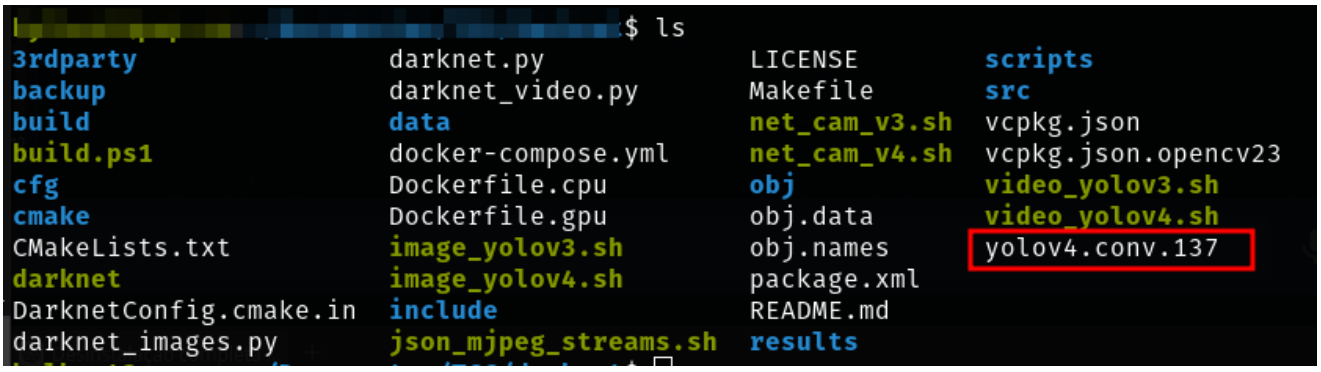
Treinamento YOLO

in /darknet

```
cp ../yolov4_custom.cfg ./cfg/
cp ../obj.* ./data/
cp ../t* ./data/
cp -r ../OIDv4_ToolKit/data/obj/ ./data/
cp -r ../OIDv4_ToolKit/data/valid/ ./data/
```

Baixar pesos pré treinados das camadas convolucionais

```
wget
https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_opti
mal/yolov4.conv.137
```




```
./darknet detector train data/obj.data cfg/yolov4_custom.cfg
yolov4.conv.137 -map
```