Treinamento da Rede Yolo com Imagens Próprias

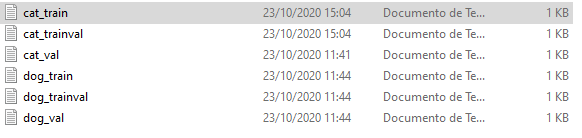
1 – Fazer as anotações nas imagens usando o labelImg

2 - Criar um diretório raiz para colocar as imagens e suas anotações

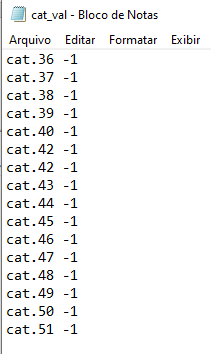
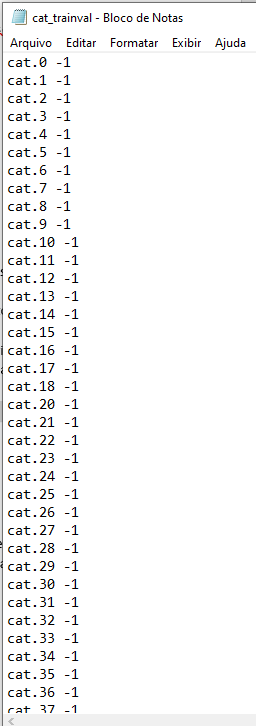
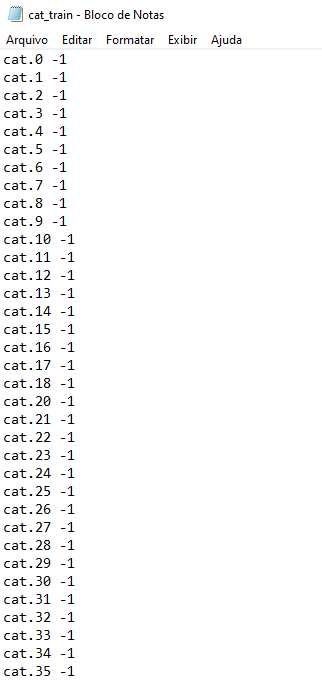
Ex: dataset\_cao\_gato

3 – Dentro do diretório dataset\_cao\_gato, temos que criar 3 diretórios:

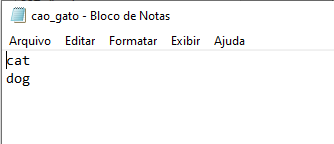
* Um com o nome Annotations, onde serão colocados todos os arquivos .xml gerado na etapa de anotações das imagens no LabelImg
* Um com nome JPEGImages, onde deverão ser colocadas todas as imagens utilizadas
* Um diretório com nome Imagesets e dentro desse diretório, criar outro com o nome Main: Dentro desse diretório, criar 3 arquivos .txt para cada classe conforme figura abaixo:



Onde dentro de cada arquivo você deverá colocar o nome de cada imagem que você separou para treinar, validar e no trainval, colocar todas de acordo com a imagem abaixo:

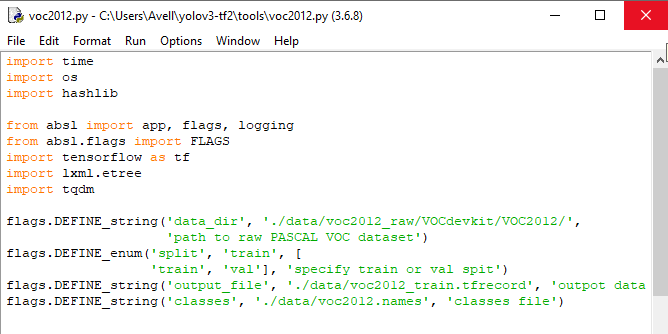


Criar um arquivo txt com o nome de cada classe que irá usar no treinamento e salvá-lo com extensão .NAMES e colocar dentro do diretório data



4 – Adaptar o dataset

Pegar o arquivo voc2012.py e adaptá-lo para o seu novo dataset



Mudar o endereçamentos dos diretórios de acordo com o dataset que você criou e salvá-lo com outro nome:

5 – Rodar a linha de código abaixo, tanto para train, quanto para val:

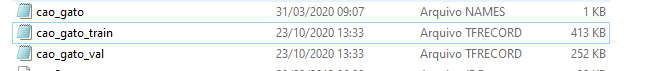
Train:

python tools/cao\_gato.py --data\_dir ./data/dataset\_cao\_gato --split train --output\_file ./data/cao\_gato\_train.tfrecord

Val:

python tools/cao\_gato.py --data\_dir ./data/dataset\_cao\_gato --split val --output\_file ./data/cao\_gato\_val.tfrecord

Esses commandos irá criar dois arquivos com extensão .tfrecord no diretório data



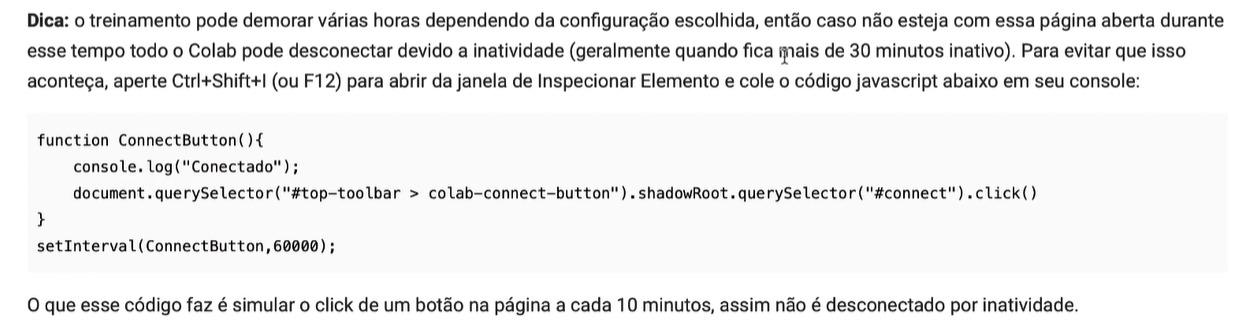
6 – Treinar a rede de acordo com o seu novo dataset com o comando abaixo

python train.py --dataset ./data/cao\_gato\_train.tfrecord --val\_dataset ./data/cao\_gato\_val.tfrecord --classes ./data/cao\_gato.names --num\_classes 2 --mode fit --transfer darknet --batch\_size 16 --epochs 10 --weights ./checkpoints/yolov3.tf --weights\_num\_classes 80

7 – Detectar as novas imagens com o seguinte comando:

python detect.py --classes ./data/cao\_gato.names --num\_classes 2 --weights ./checkpoints/yolov3\_train\_10.tf --image ./data/cat0.jpg

obs.: Fiquem atento com relação ao apontamento de diretórios.



Ou

function ConnectButton(){

document.querySelector("#top-toolbar > colab-connect-button").shadowRoot.querySelector("#connect-icon").click();

}

setInterval(ConnectButton, 60000);