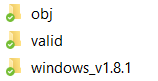
Treinamento da Yolo com o Próprio Dataset

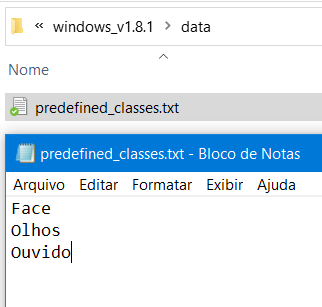
Criando o Dataset

1 – Criar 2 pastas contendo as classes que irá utilizar no treinamento, uma com o nome de **obj** e outra com o nome **valid** (esses nomes são criados de acordo com a sua escolha).

* 1. Coloque na pasta **obj** os arquivos de treinamento, na pasta **valid** coloque os de validação.

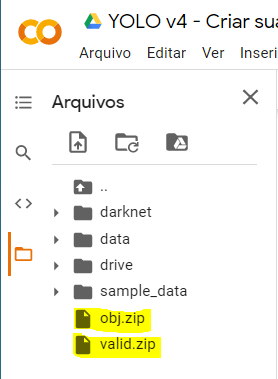


* 1. Abra a pasta **windows\_v1.8.1**. Dentro dela há uma pasta chamada **data**, abra o arquivo **predefined\_classes.txt** e coloco os nomes das classes que deseja configurar.



* 1. Execute o arquivo **labelImg.exe**, clique em **Open Dir** e encontre o diretório **obj**. As imagens armazenadas na pasta aparecerão listadas no programa para serem inseridas as bounding boxes.
  2. No **labelImg.exe**, clique em **PascalVOC** para mudar o formato para **YOLO.**
  3. Na sequência, clique em **Create\nRectBox** para desenhar as bounding boxes. Após desenhar os retângulos, clique em **Save** para salvar o arquivo .txt com as labels na pasta da imagem.
  4. Clique em **Next Image** para avançar para a próxima imagem.
  5. Repetir os itens acima para os arquivos da pasta **valid** e crie um arquivo .zip para ambos.

1. – Para iniciar a preparação dos dados, arraste os arquivos **obj.zip** e **valid.zip** para o explorador de **Arquivos** do ambiente Colab.



Iniciando o treinamento

3 – Zipar as pastas **obj** e **valid** e colocar dentro de um diretório que você irá armazenar todas as rotinas necessárias para o treinamento.

4 – Editar o arquivo de configuração da rede (.cfg)

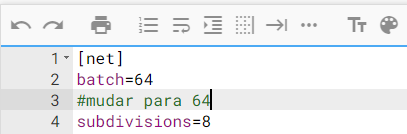
4.1 – Fazer o git clone

4.2 – montar o framework: !make

4.3 – fazer uma cópia do yolov4.cfg para o diretório que criamos no colab para fazer o treinamento e renomeá-lo para yolov4\_custom.cfg

4.4 – abrir o arquivo yolov4\_custom.cfg com o editor de texto do colab e editá-lo de acordo com as suas características de treinamento (Faremos para o cao-gato)

a) Mudar o subdivision para 64 => indica consumo de memória, se colocar um valor muito baixo ele vai consumir mais memória e pode ser que ele não execute no Colab. Na primeira vez que fiz o teste, não troquei este valor e ele travou



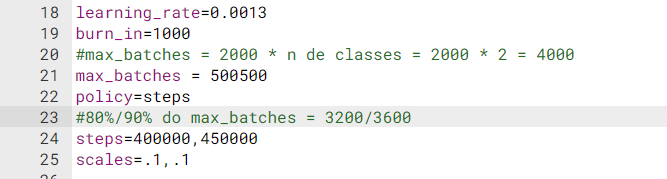
b) Mudar o max\_batches (número total de registros de cada batch) tem um valor alto de acordo com a base completa, pois a mesma tem muitas classes. Pela documentação da Yolo, deve-se usar a seguinte relação:

max\_batches = 2000 \* nº de classes

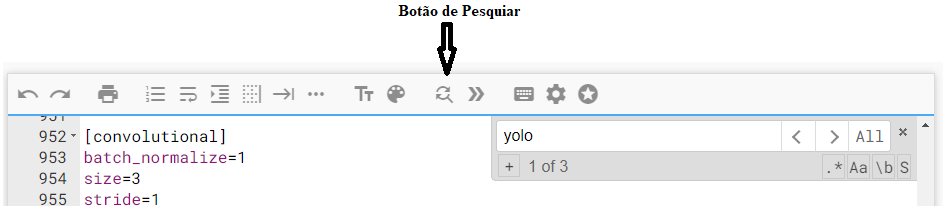
para o nosso caso, como temos 2 classes,

max\_batches = 2000 \* 2 = 4000

c) Mudar o número de steps: Usar como parâmetro 80% e 90% do max\_batches (tudo isso foi baseado na documentação da Yolo)



4.5 – Clicar no botão pesquisar e digitar yolo, vai aparecer 3 valores:



em cada um você deverá trocar o número de classes (está 80 devido ao número de classes original) deve-se passar para o número de classes que você irá treinar, nesse caso colocaremos 2.

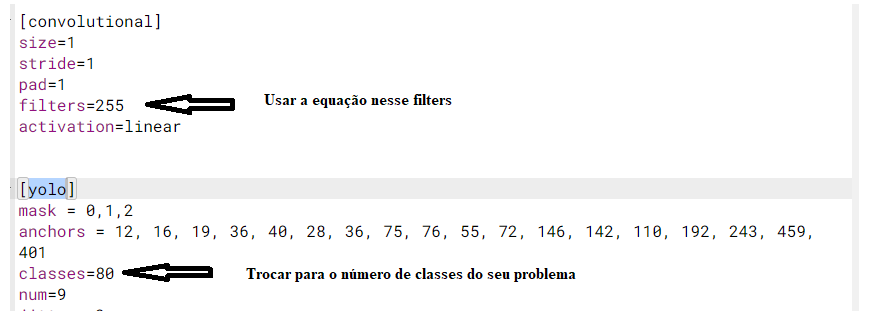
Mudar também o parâmetro filters logo acima usando a seguinte relação:

filter = (nº de classes + 5) \* 3

no nosso caso:

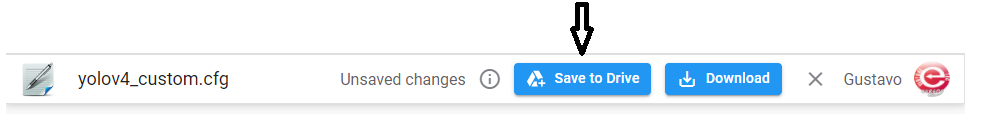
filter = (2 + 5) \* 3 = 21

fazer isso apenas no filter da última camada de convolução.

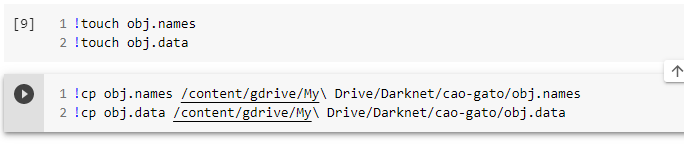


Clicar na seta para localizar o outro Yolo e repetir os passos acima.

Para finalizar, clicar no botão Save to Drive

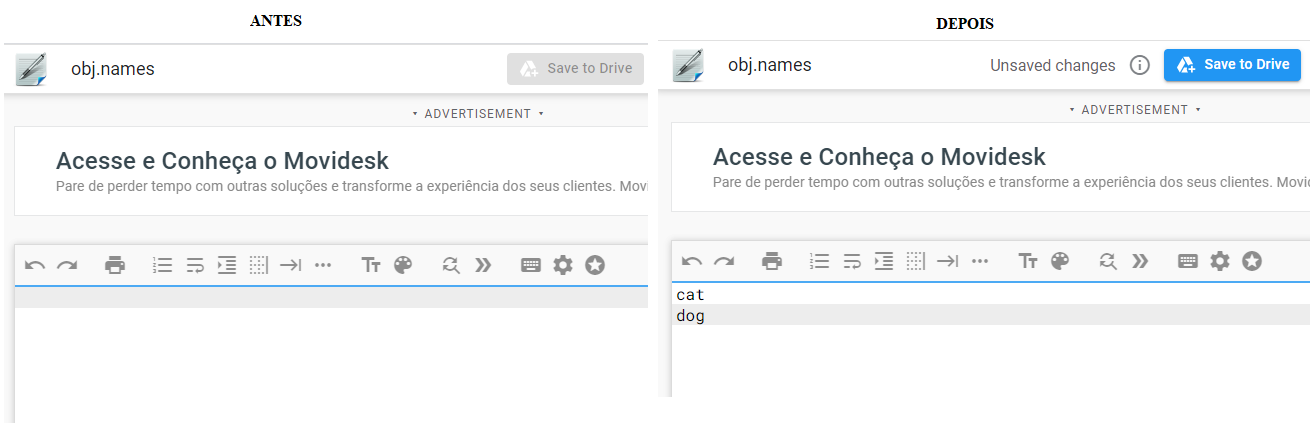


5 – Criar dois arquivos obj.names e obj.data e salvá-lo dentro da pasta do Colab, onde está salvando tudo para realizar o treinamento:

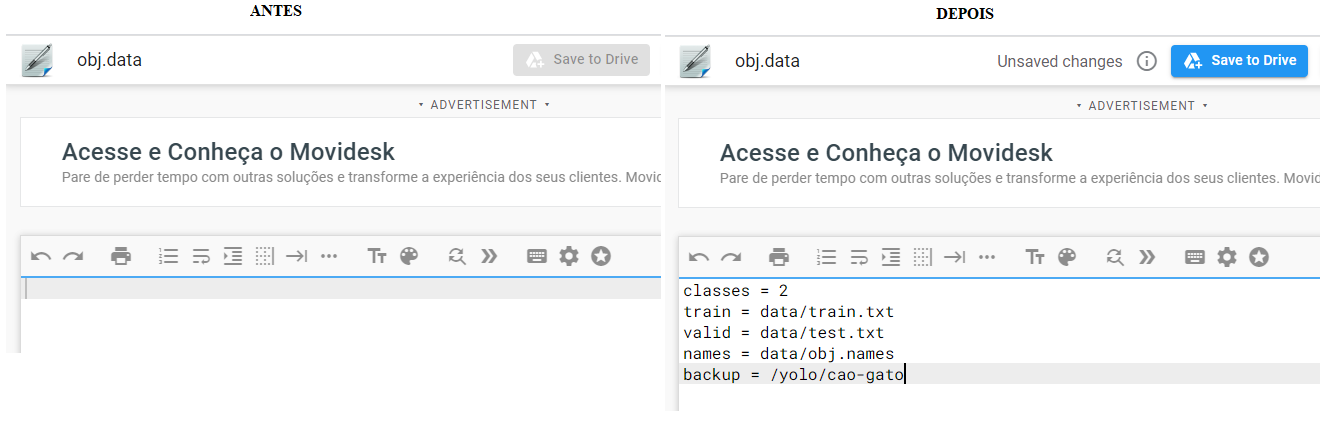


A figura acima mostra como criar e salvar esses dois arquivos dentro da nossa pasta cao-gato, onde estamos salvando todos os arquivos, porém, eles estão vazios, temos que editá-los de acordo com a característica de cada um:

obj.names => um arquivo txt que deverá ter o nome das classes utilizadas no treinamento



Obj.data => informações sobre o dataset e o armazenamento das imagens



classes =

train = data/train.txt

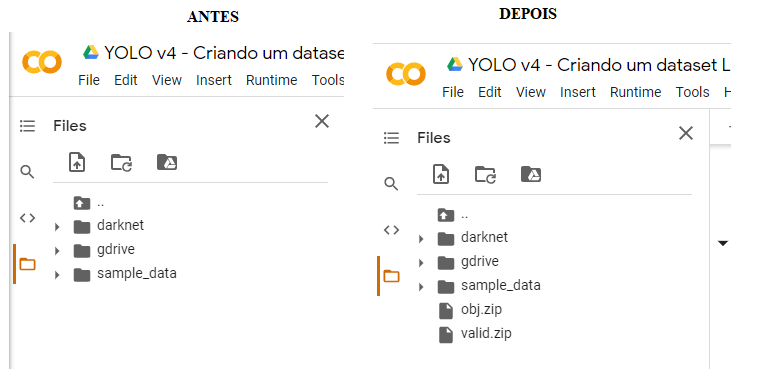
valid = data/test.txt

names = data/obj.names

backup = /yolo/Dataset

6 – Gerar dois arquivos: train.txt e tes.txt

6.1 – Copiar os dois arquivos obj.zip e valid.zip para a pasta raiz do colab

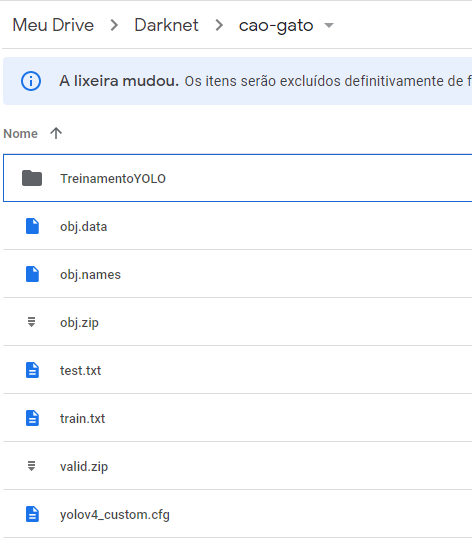


6.2 – Descompactar os dois arquivos

6.3 – Utilizar os arquivos gerar\_train.py e gerar\_test.py para criar os dois arquivo .txt que irão conter o nome de cada imagen

6.4 – Copiar os dois arquivos para o diretório que estamos salvando todos os arquivos

Por fim, o diretório deverá conter todos esses arquivos:



Essa primeira parte foi a preparação do dataset para ser utilizado no treinamento da rede Yolo.