Detecção de Objetos Atraves da Biblioteca Mask RCNN

Kaio Takeshi Arakawaa Dos Santos  
Departamento de Ciência Da Computação  
IFC-VideiraVideira  
Kaioarakawa@gmail.com

*Resumo*—Nesse presente artigo será demostrado a forma de detecção de objetos por meio da Mask RCNN, um modelo criado em 2017 para auxiliar a detecção de objetos e treinamentos de rede neurais na área.

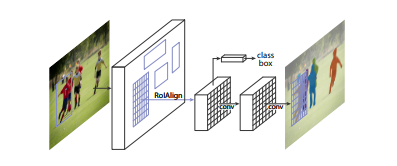
Keywords—Mask RCNN, modelo, rede neural.

# Introdução

Nos últimos anos, algoritmos baseados em redes neurais convolucionais (CNNs) levaram a avanços dramáticos no estado da arte para problemas fundamentais em visão computacional, como detecção de objetos, localização de objetos, segmentação semântica e segmentação de instâncias de objetos [1].

A detecção de objetos é uma tarefa muito popular na comunidade de pesquisa de visão computacional e uma ferramenta útil para muitas aplicações do mundo real. Muitos desses aplicativos estão na área da robótica. Um robô auxiliado com informações dos objetos ao seu redor estará mais bem equipado para ter interações bem-sucedidas com seu ambiente [2].

Tradicionalmente, os métodos de detecção de objetos são projetados e avaliado para a métrica de precisão média (mAP). O mAP classifica a saída das caixas delimitadoras por um sistema de detecção com base na pontuação de cada caixa. Os valores brutos deas pontuações são difíceis de interpretar, exceto que uma pontuação mais alta é significa que o sistema está mais confiável. O sistema de classificação geralmente incentiva os detectores a produzirem muitas detecções para cada imagem, já que normalmente isso só aumentará sua pontuação mAP [1].

JEREMIAH W. JOHNSON

1. O modelo Mask-RCNN.

# O Modelo Mask-RCNN

O modelo Mask-RCNN foi desenvolvido em 2017 e estende o modelo Faster-RCNN para segmentação semântica, localização de objetos e segmentação de instância de objeto de imagens naturais. Mask-RCNN é descrito pelos autores como fornecendo uma "estrutura simples, flexível e geral para o objeto segmentação de instância“ [3].

Mask-RCNN foi usado para superar todas as entradas monomodelo existentes em todas as tarefas no Desafio COCO [1].

Ele é um método de aprendizagem profunda muito popular para detecção de objeto e segmentação de instância que alcançou resultados de última geração no conjunto de dados MSCOCO quando publicados [3].

Embora alguns detectores tenham se sobressaído ao MaskRCNN no desempenho do mAP, eles o fizeram por apenas um poucos pontos e geralmente os mesmos são baseados na arquitetura Mask-RCNN [3].

O Mask-RCNN é um pipeline de reconhecimento de dois estágios. Em seu primeiro estágio, os recursos são extraídos da imagem usando um rede neural convencional de backbone e classe propostas de regiões agnósticas são previstas. Essas propostas são então refinados e classificados no segundo estágio, para se tornarem caixas delimitadoras rotuladas para detecção de objeto ou máscaras de segmentação para a tarefa de segmentação de instância [4].

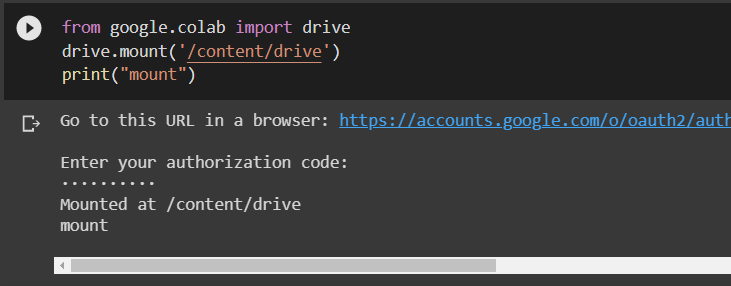
Os autores fornecem modelos treinados para detecção de objetos e segmentação de instâncias no MSCOCO, usando redes de backbone que são pré-treinadas no conjunto de dados de classificação [5].

# desenvolvimento dO mask-RCNN

Esta é uma implementação do Mask R-CNN no Python, Keras e TensorFlow. O modelo gera caixas delimitadoras e máscaras de segmentação para cada instância de um objeto na imagem. É baseado em Feature Pyramid Network (FPN) e um backbone ResNet101 [3].

Com o uso do Google colaboratory foi criado uma rede neural para a detecção de objetos através de uma imagem retirada de uma câmera local. Em base ele visualiza cada etapa da Rede de Proposta de Região do primeiro estágio e exibe âncoras positivas e negativas junto com o refinamento da caixa de âncora [2].

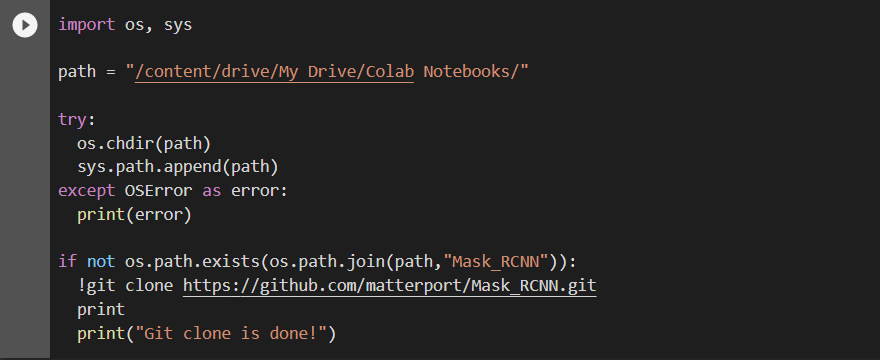
Na primeira etapa foi feito um código para montar o drive ao projeto para com isso conseguir fazer a segunda parte.

O autor 

1. Mount do drive no projeto.

Como já dito anteriormente o essa segunda etapa necessita da primeira etapa fazendo para conseguir salvar o repositório no drive local, nesse repositório se encontra funções necessárias para a execução da detecção de objetos pela rede neural mas não apenas isso nele também já tem os dados da rede neural treinada facilitando um pouco o processo [5].

O autor

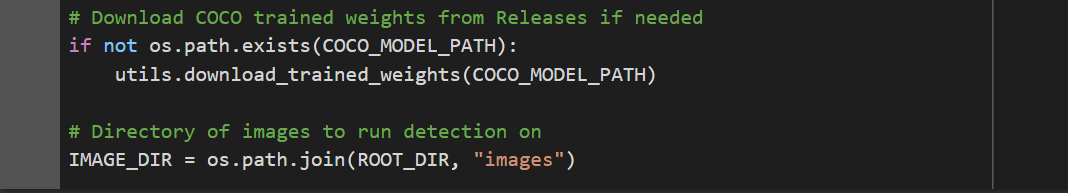
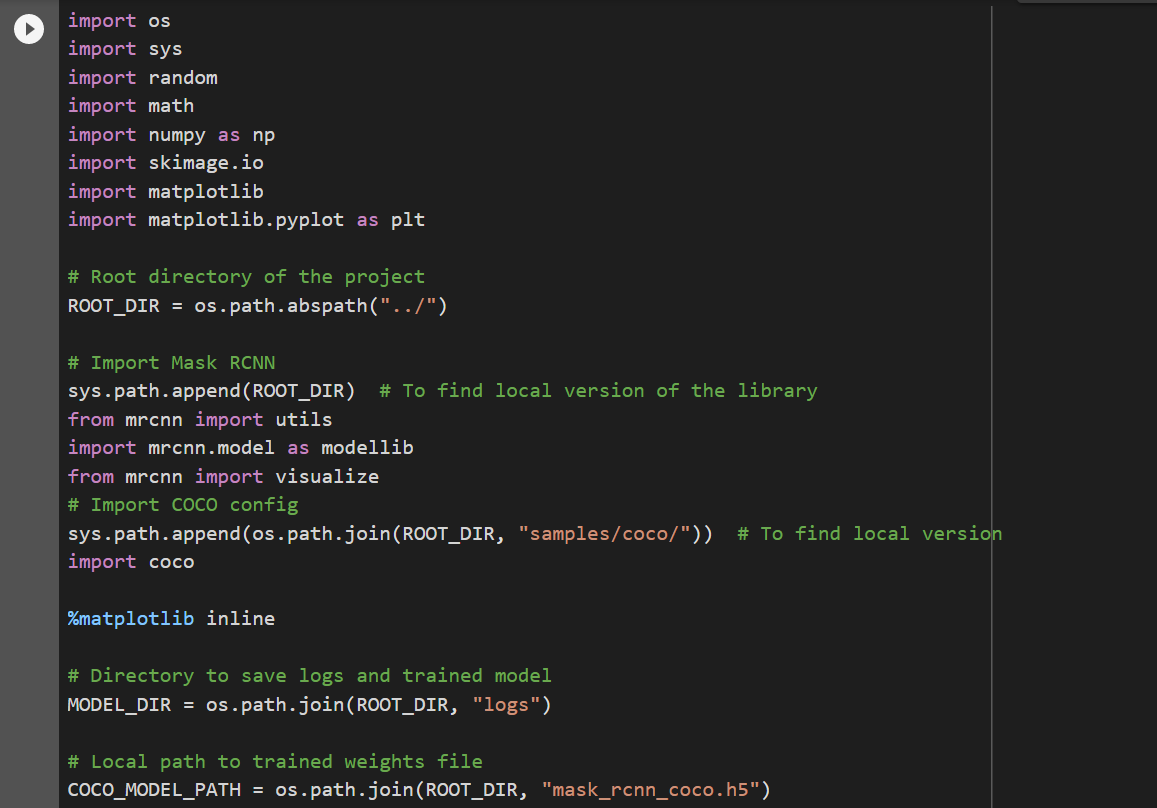


1. Clone do mask\_RCNN.

Nessa etapa é feito a maioria das importações necessárias, como já dito é necessário do Keras e do Tenserflow, porém existem outras que também são necessárias como podemos ver pela imagem abaixo [4].

Essa etapa não é utilizada apenas para a importação de bibliotecas, mas também para a retirada dos diretórios de treinos e também dos dados do treino para a rede neural. Podemos ver que temos uma importação da mask\_rcnn\_coco dela que vem os dados de treinamento.

O autor

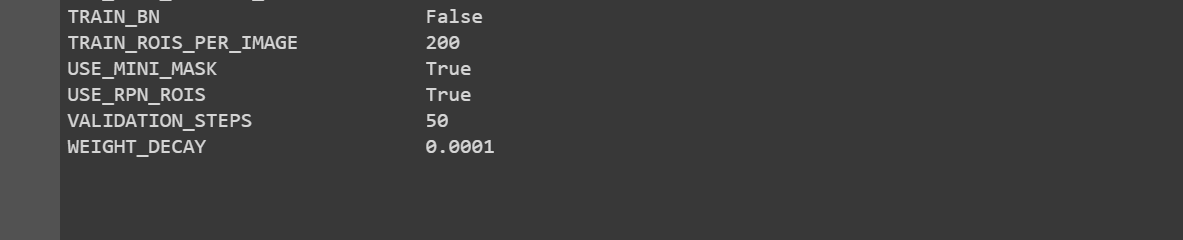
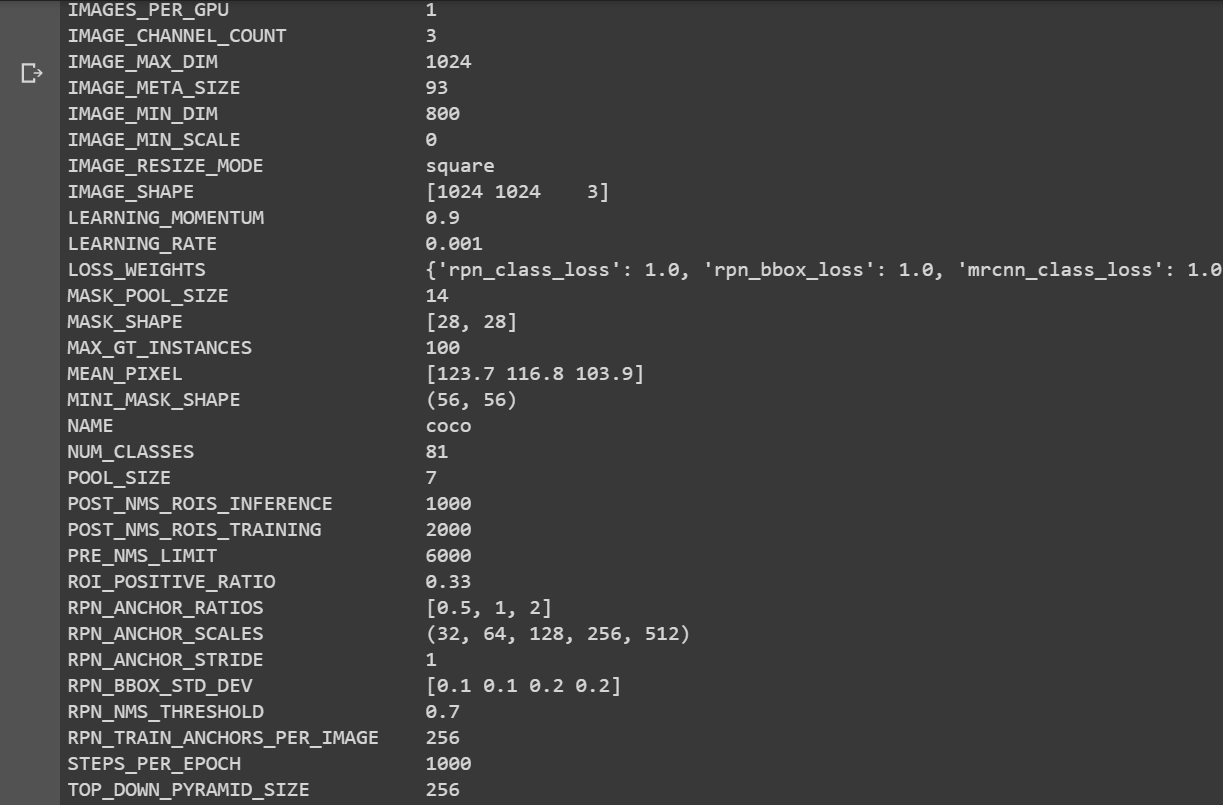
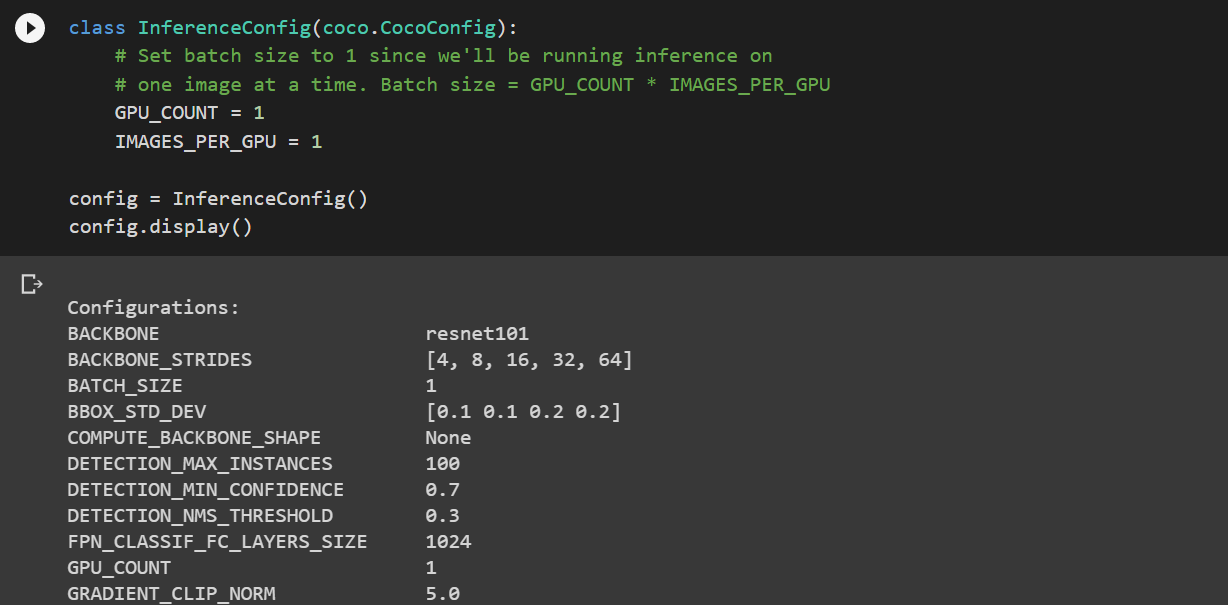


1. Configurações iniciais.

Na próxima imagem temos apenas uma área de configuração, onde está sendo usando um modelo treinado no conjunto de dados MS-COCO. As configurações deste modelo estão na classe CocoConfig em coco.py.

Para inferir, foi modificado um pouco as configurações para se adequar à tarefa. Para fazer isso, uma subclasse será criada da classe CocoConfig e substituída com os atributos necessários, como podemos ver na imagem, onde no console é retornado a config feita. Um ponto que podemos destacar é o backbone e nossa batch size.

O autor

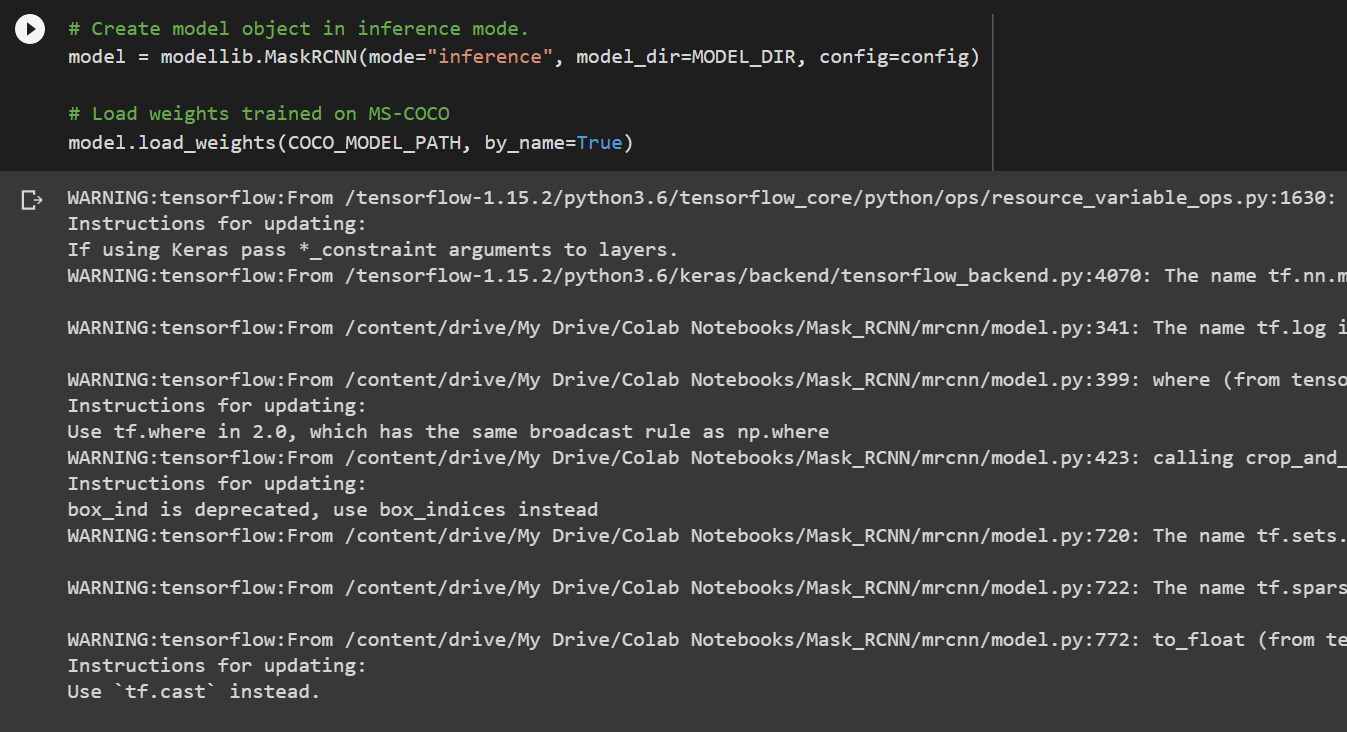


1. Configurações na MS-COCO.

Como próximo passo temos a criação de um modelo. Esse modelo classifica objetos e retorna IDs de classe, que são valores inteiros que identificam cada classe. Alguns conjuntos de dados atribuem valores inteiros às suas classes e outros não. Por exemplo, no conjunto de dados MS-COCO, a classe 'pessoa' é 1 e 'urso de pelúcia' é 88. Os IDs são frequentemente sequenciais, mas nem sempre. O conjunto de dados COCO, por exemplo, tem classes associadas aos IDs de classe 70 e 72, mas não 71.

Para melhorar a consistência e dar suporte ao treinamento em dados de várias fontes ao mesmo tempo, a classe Dataset atribui seus próprios IDs inteiros sequenciais a cada classe. Por exemplo, se você carregar o dataset COCO usando essa Dataset, a classe 'pessoa' obterá classe ID = 1 (assim como COCO) e a classe 'ursinho de pelúcia' é 78 (diferente de COCO).

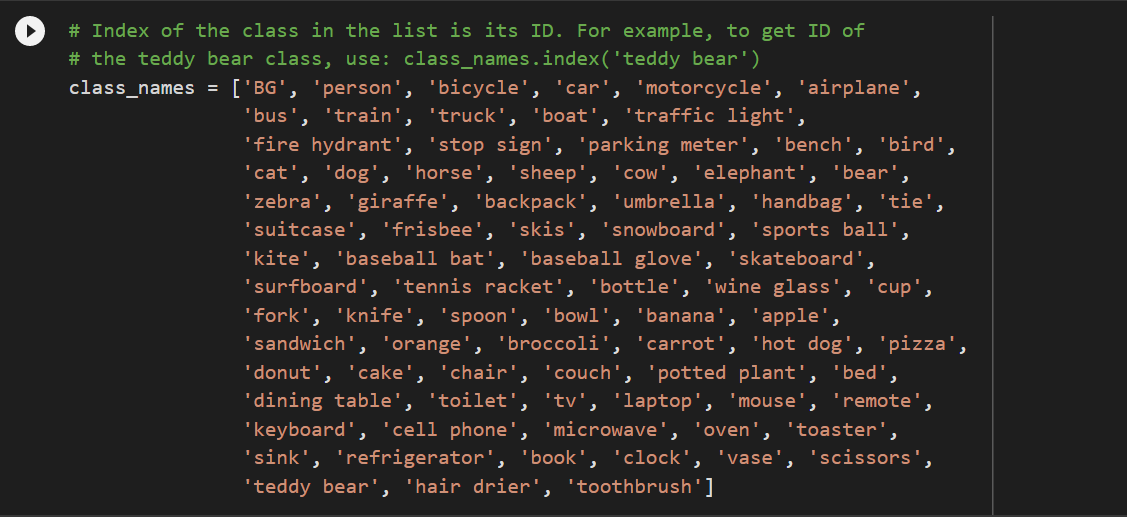
O autor



1. Criação do modelo de objetos.

Para complementar o texto anterior temos a classe criada para não ser necessário baixar conjunto de dados COCO.

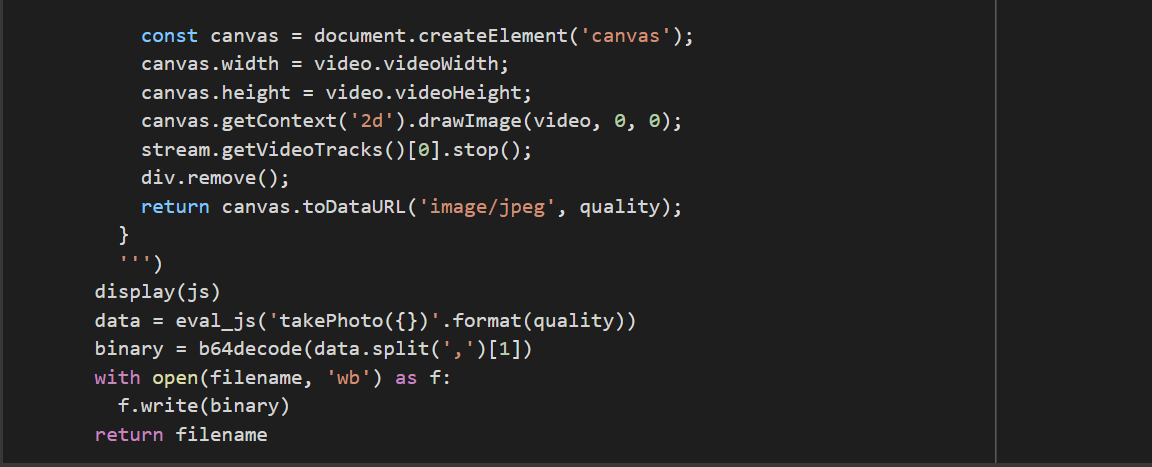
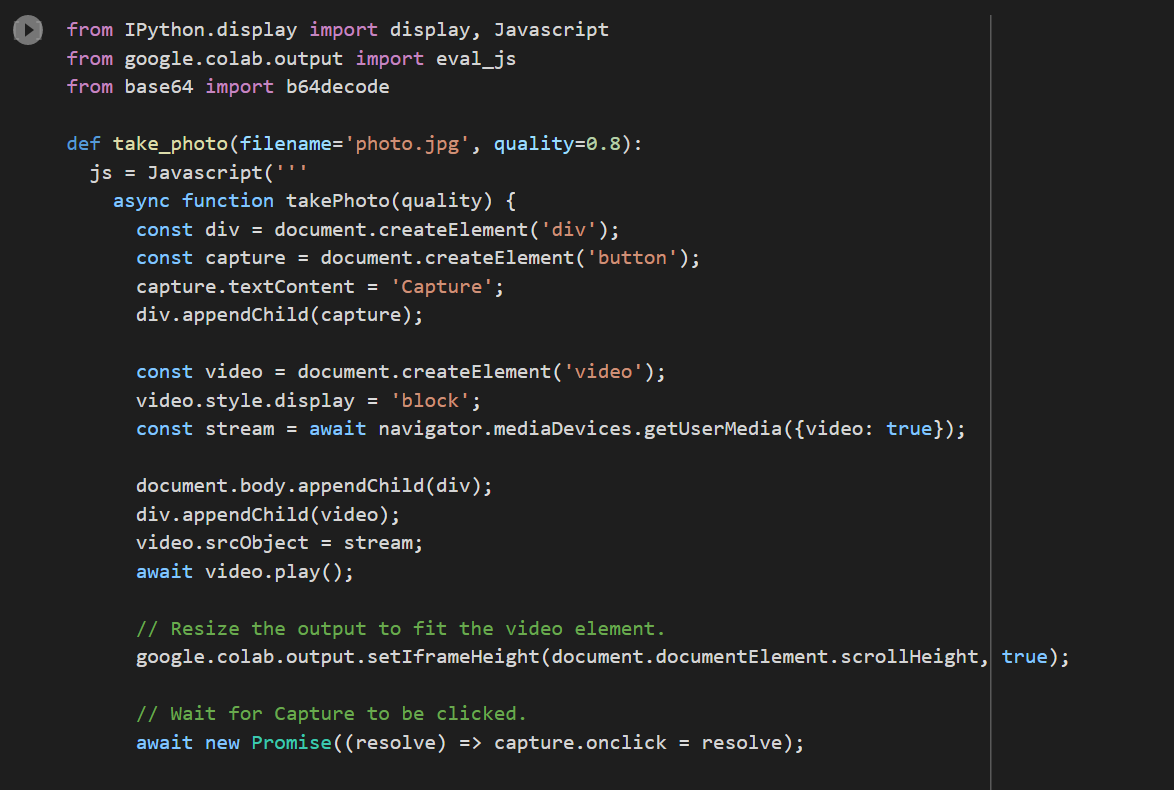
O autor



1. Class names com nome e ids para os objetos.

Após todo o treinamento e configuração chegamos a parte de captura da imagem, nessa área estamos importante mais algumas bibliotecas a fim de criar uma função chamada take\_photo para fazer a captura da imagem e logo depois modifica-la de forma que a imagem fique com as configurações ideias.

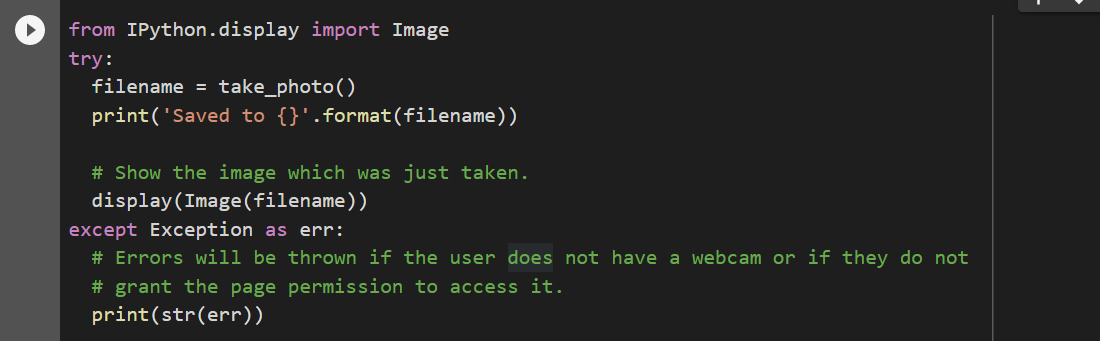
O autor



1. Criação da funçao take\_photo.

Por fim chamamos a função take\_photo para fazer a captura da imagem, como já configuramos, na área do console aparece um display para fazer a captura do jeito que o usuário necessitar.

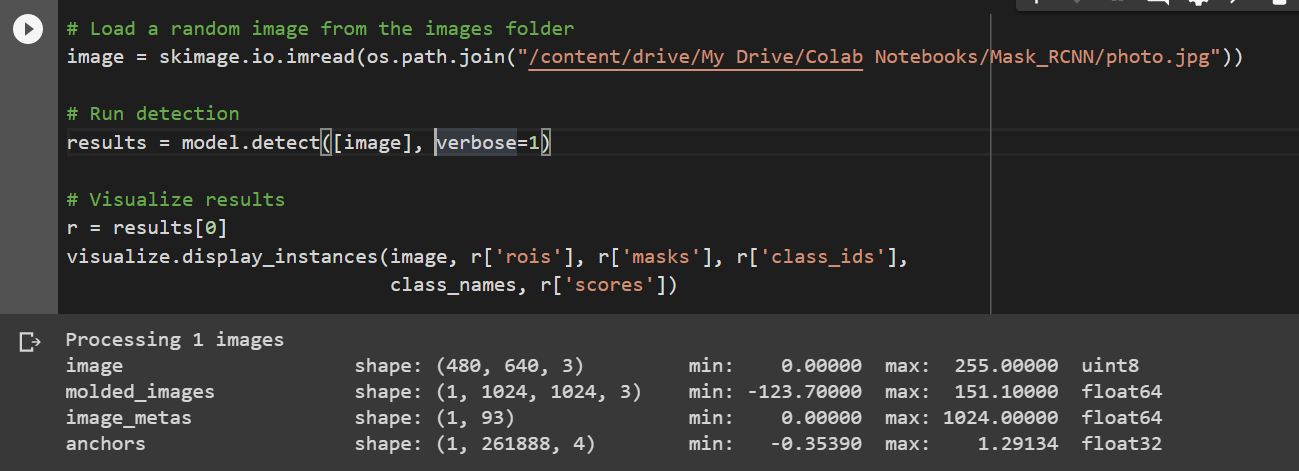
O autor



1. Utilização da função take\_photo.

Após a retirada da imagem só resta um ultimo passo que seria a chamada da rede neural para detectar os objetos presentes na mesma. Passando os dados de treinamento e a imagem.

O autor



1. Chamada da rede neural para detecção de objetos.

E como resultado temos a imagem abaixo, demostrando os objetos de forma quadriculada com os nomes do mesmo e a porcentagem de acerto ao lado.

O autor



1. Imagem final da detecção de objetos.

##### Conclusão

Neste artigo, demonstramos que o modelo Mask-RCNN, projetado principalmente com a detecção de objetos, localização de objetos e segmentação de instâncias de imagens naturais em mente, pode ser usado para produzir resultados de alta qualidade para os desafios apresentados no dia a dia.

Existem várias tarefas semelhantes na análise de imagens que podem ser atribuídas a esse método, um deles seria a segmentação medica para a detecção de moléculas anômalas por exemplo. Um ponto seria a detecção da corona vírus através de imagem.

Por fim a Mask-RCNN tem muito futuro e algumas melhorias que podem ser feitas para a melhor detecção como o bug na imagem acima onde ela detectou duas pessoas sendo que era apenas o usuário com o dedo levantado. Um ponto para determinar isso seria com mais treinamento para a rede neural entender melhor as áreas de contato e conseguir distinguir os falsos positivos.

##### References

1. HILL, Phil Ammirato Unc-Chapel. **A Mask-RCNN Baseline for Probabilistic Object Detection**. 2019. Disponível em: https://arxiv.org/pdf/1908.03621.pdf. Acesso em: 13 ago. 2020.
2. HE, Peilin. **A Value Recognition Algorithm for Pointer Meter Based on Improved Mask-RCNN**. Disponível em: https://ieeexplore.ieee.org/abstract/document/8836852. Acesso em: 13 ago. 2020.
3. JOHNSON, Jeremiah W.. **ADAPTING MASK-RCNN FOR AUTOMATIC NUCLEUS SEGMENTATION**. 2018. 7 f. Tese (Doutorado) - University Of New Hampshire, Manchester, 2018.G.
4. MATTERPORT. **Mask rcnn**. Disponível em: https://colab.research.google.com/github/tensorflow/tpu/blob/master/models/official/mask\_rcnn/mask\_rcnn\_demo.ipynb. Acesso em: 13 ago. 2020
5. PANELLINZUO, Author Links Open Overlay. **A robust approach to reading recognition of pointer meters based on improved mask-RCNN**. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S092523122030076X. Acesso em: 13 ago. 2020.