

Lab 6 - BCC406

REDES NEURAIS E APRENDIZAGEM EM PROFUNDIDADE

Detecção e Segmentação de objetos

Prof. Eduardo e Prof. Pedro

Objetivos:

- Detecção de objetos

Data da entrega : 25/02

- Este notebook é baseado em tensorflow e Keras.
- Execute todo notebook e salve tudo em um PDF **nomeado** como "NomeSobrenome-LabX.pdf"
- Envie o PDF via google [FORM](#)

✓ Detecção de Objetos (100pt)

Execute o tutorial do [link](#). Faça um teste com os seguintes modelos:

- EfficientDet D0 512x512
- SSD MobileNet V2 FPNLite 320x320
- SSD ResNet50 V1 FPN 640x640 (RetinaNet50)
- Faster R-CNN ResNet50 V1 640x640
- Mask R-CNN Inception ResNet V2 1024x1024

Teste com imagens de:

- Praia ([link](#))
- Cachorros ([link](#))
- Pássaros ([link](#))

Deixe disponível aqui somente as saídas dos modelos que você testou. Não é necessário copiar todo tutorial para cá.

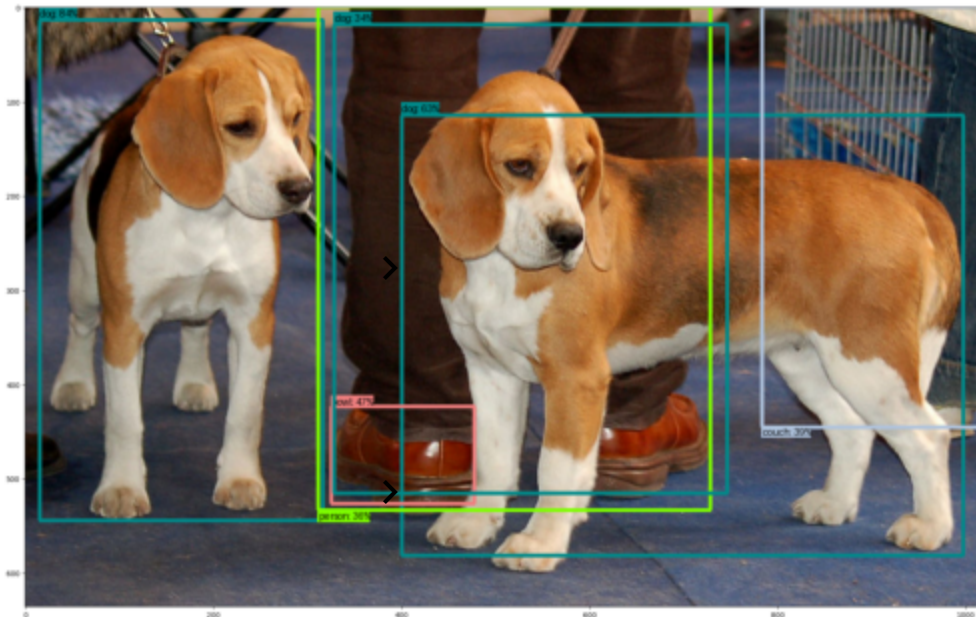
```
1 import matplotlib.pyplot as plt
2
```

```
3 def plot_images(model):
4     targets = ["beach", "dog", "bird"]
5
6     for target in targets:
7         img_path = target + "_" + model + ".png"
8         imagem = plt.imread(img_path)
9
10        # Exibe a imagem
11        plt.imshow(imagem)
12        plt.axis('off') # Remove os eixos (opcional)
13        plt.show()
14
```

✓ Modelo EfficientDet D0 512x512 (10pt)

```
1 plot_images("efficientDet")
```





Modelo SSD MobileNet V2 FPN Lite 320x320 (10pt)

[] ↳ 1 célula oculta

Modelo SSD ResNet50 V1 FPN 640x640 - RetinaNet50 (10pt)

[] ↳ 1 célula oculta

Modelo Faster R-CNN ResNet50 V1 640x640 (10pt)

[] ↳ 1 célula oculta

Modelo Mask R-CNN Inception ResNet V2 1024x1024 (10pt)

[] ↳ 1 célula oculta

✓ Entendendo os resultados (20pt)

ToDo: O que você pode descrever sobre os resultados obtidos?

Todas as imagens possuem um tamanho muito maior do que a entrada de qualquer um dos modelos. Todas

EfficientDet: A imagem dos passáros possui uma confiança baixa em algumas predições, enquanto alg

SSD Mobilenet: Apresentou péssimos resultados, fazendo poucas predições na praia, identificou vár

SSD Resnet: Possui os mesmos problemas da versão com a Mobilenet

Faster R-CNN: Para a imagem da praia onde os objetos são pequenos, foi possível detectar muito be

Mask R-CNN: Apresentou os melhores resultados, porém não identificou um único passáro e duas pess

Pode-se levantar a hipótese de que, devido ao maior tamanho de entrada aceito pela Mask R-CNN, as

✓ Custo computacional (30pt)

Compute o custo computacional (tempo de inferência) de cada modelo acima

Dica : Use o método "default_timer" da biblioteca "timeit"

```
1 import timeit
2
3 def make_inferences():
4     models = [
5         'EfficientDet D0 512x512',
6         'SSD MobileNet V2 FPNLite 320x320',
7         'SSD ResNet50 V1 FPN 640x640 (RetinaNet50)',
8         'Faster R-CNN ResNet50 V1 640x640',
9         'Mask R-CNN Inception ResNet V2 1024x1024'
10    ]
11
12    for model in models:
13        model_handle = ALL_MODELS[model]
14        hub_model = hub.load(model_handle)
15
16        image_path = IMAGES_FOR_TEST["Beach"]
17        image_np = load_image_into_numpy_array(image_path)
18
```

```
19     # Fazendo a inferência
20     inicio = timeit.default_timer()
21
22     hub_model(image_np)
23
24     fim = timeit.default_timer()
25     print (f'{model}: duracao: %f' % (fim - inicio))
26
27 make_inferences()
```

- EfficientDet D0 512x512: duracao: 4.920923
- SSD MobileNet V2 FPNLite 320x320: duracao: 3.633905
- SSD ResNet50 V1 FPN 640x640 (RetinaNet50): duracao: 3.893012
- Faster R-CNN ResNet50 V1 640x640: duracao: 6.880522
- Mask R-CNN Inception ResNet V2 1024x1024: duracao: 6.499287

Os códigos foram executados no Notebook do tf2_object_detection

✓ Detectando objetos com dados próprios (Opcional / 20 Pontos Extra)

Caso você queira usar as técnicas de detecção de objetos em uma base de dados própria, siga o tutorial do [link](#). Você também pode se basear no trabalho do [link](#).

Relate sua experiência e anexe aqui os resultados.

