

Lab 5 - BCC406

REDES NEURAIS E APRENDIZAGEM EM PROFUNDIDADE

ConvNets para classificação e localização

Prof. Eduardo e Prof. Pedro

Objetivos:

- Parte I : Detecção de objetos
- Parte II : Segmentação de imagens na [Oxford Pet Dataset](#)

Data da entrega : --/--

- Complete o código (marcado com ToDo) e quando requisitado, escreva textos diretamente nos notebooks. Onde tiver *None*, substitua pelo seu código.
- Execute todo notebook e salve tudo em um PDF **nomeado** como "NomeSobrenome-Lab4.pdf"
- Envie o PDF via google [FORM](#)

Este notebook é baseado em tensorflow e Keras.

▼ Parte I - Detecção de Objetos (60pt)

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
- Cachorros
- Pássaros

▼ ToDo : Custo computacional (30pt)

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

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

```
# This Colab requires TF 2.5.
!pip install -U "tensorflow>=2.5"

Requirement already satisfied: tensorflow>=2.5 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: tf-estimator-nightly==2.8.0.dev2021122109 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: tensorboard<2.9,>=2.8 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: tensorflow-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: urllib3!=1.25.0,!>=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages/tensorflow/_api/python/framework/dtypes.py
```

```
import os
import pathlib

import matplotlib
import matplotlib.pyplot as plt
```

```

import io
import scipy.misc
import numpy as np
from six import BytesIO
from PIL import Image, ImageDraw, ImageFont
from six.moves.urllib.request import urlopen

import tensorflow as tf
import tensorflow_hub as hub

tf.get_logger().setLevel('ERROR')

```

@title Run this!!

Run this!!



```

def load_image_into_numpy_array(path):
    """Load an image from file into a numpy array.

    Puts image into numpy array to feed into tensorflow graph.
    Note that by convention we put it into a numpy array with shape
    (height, width, channels), where channels=3 for RGB.

Args:
    path: the file path to the image

Returns:
    uint8 numpy array with shape (img_height, img_width, 3)
"""
image = None
if(path.startswith('http')):
    response = urlopen(path)
    image_data = response.read()
    image_data = BytesIO(image_data)
    image = Image.open(image_data)
else:
    image_data = tf.io.gfile.GFile(path, 'rb').read()
    image = Image.open(BytesIO(image_data))

(im_width, im_height) = image.size
return np.array(image.getdata()).reshape(
    (1, im_height, im_width, 3)).astype(np.uint8)

```

```

ALL_MODELS = {
'CenterNet HourGlass104 512x512' : 'https://tfhub.dev/tensorflow/centernet/hourglass_512x5
'CenterNet HourGlass104 Keypoints 512x512' : 'https://tfhub.dev/tensorflow/centernet/hourg
'CenterNet HourGlass104 1024x1024' : 'https://tfhub.dev/tensorflow/centernet/hourglass_102
'CenterNet HourGlass104 Keypoints 1024x1024' : 'https://tfhub.dev/tensorflow/centernet/hou
'CenterNet Resnet50 V1 FPN 512x512' : 'https://tfhub.dev/tensorflow/centernet/resnet50v1_f
'CenterNet Resnet50 V1 FPN Keypoints 512x512' : 'https://tfhub.dev/tensorflow/centernet/re
'CenterNet Resnet101 V1 FPN 512x512' : 'https://tfhub.dev/tensorflow/centernet/resnet101v1
'CenterNet Resnet50 V2 512x512' : 'https://tfhub.dev/tensorflow/centernet/resnet50v2_512x5
'CenterNet Resnet50 V2 Keypoints 512x512' : 'https://tfhub.dev/tensorflow/centernet/resnet
'EfficientDet D0 512x512' : 'https://tfhub.dev/tensorflow/efficientdet/d0/1',

```

```
'EfficientDet D1 640x640' : 'https://tfhub.dev/tensorflow/efficientdet/d1/1',
'EfficientDet D2 768x768' : 'https://tfhub.dev/tensorflow/efficientdet/d2/1',
'EfficientDet D3 896x896' : 'https://tfhub.dev/tensorflow/efficientdet/d3/1',
'EfficientDet D4 1024x1024' : 'https://tfhub.dev/tensorflow/efficientdet/d4/1',
'EfficientDet D5 1280x1280' : 'https://tfhub.dev/tensorflow/efficientdet/d5/1',
'EfficientDet D6 1280x1280' : 'https://tfhub.dev/tensorflow/efficientdet/d6/1',
'EfficientDet D7 1536x1536' : 'https://tfhub.dev/tensorflow/efficientdet/d7/1',
'SSD MobileNet v2 320x320' : 'https://tfhub.dev/tensorflow/ssd_mobilenet_v2/2',
'SSD MobileNet V1 FPN 640x640' : 'https://tfhub.dev/tensorflow/ssd_mobilenet_v1/fpn_640x64
'SSD MobileNet V2 FPNLite 320x320' : 'https://tfhub.dev/tensorflow/ssd_mobilenet_v2/fpnlit
'SSD MobileNet V2 FPNLite 640x640' : 'https://tfhub.dev/tensorflow/ssd_mobilenet_v2/fpnlit
'SSD ResNet50 V1 FPN 640x640 (RetinaNet50)' : 'https://tfhub.dev/tensorflow/retinanet/resn
'SSD ResNet50 V1 FPN 1024x1024 (RetinaNet50)' : 'https://tfhub.dev/tensorflow/retinanet/re
'SSD ResNet101 V1 FPN 640x640 (RetinaNet101)' : 'https://tfhub.dev/tensorflow/retinanet/re
'SSD ResNet101 V1 FPN 1024x1024 (RetinaNet101)' : 'https://tfhub.dev/tensorflow/retinanet/
'SSD ResNet152 V1 FPN 640x640 (RetinaNet152)' : 'https://tfhub.dev/tensorflow/retinanet/re
'SSD ResNet152 V1 FPN 1024x1024 (RetinaNet152)' : 'https://tfhub.dev/tensorflow/retinanet/
'Faster R-CNN ResNet50 V1 640x640' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet50_v1
'Faster R-CNN ResNet50 V1 1024x1024' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet50_
'Faster R-CNN ResNet50 V1 800x1333' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet50_v
'Faster R-CNN ResNet101 V1 640x640' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet101_
'Faster R-CNN ResNet101 V1 1024x1024' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet10
'Faster R-CNN ResNet101 V1 800x1333' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet101
'Faster R-CNN ResNet152 V1 640x640' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet152_
'Faster R-CNN ResNet152 V1 1024x1024' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet15
'Faster R-CNN ResNet152 V1 800x1333' : 'https://tfhub.dev/tensorflow/faster_rcnn/resnet152
'Faster R-CNN Inception ResNet V2 640x640' : 'https://tfhub.dev/tensorflow/faster_rcnn/inc
'Faster R-CNN Inception ResNet V2 1024x1024' : 'https://tfhub.dev/tensorflow/faster_rcnn/i
'Mask R-CNN Inception ResNet V2 1024x1024' : 'https://tfhub.dev/tensorflow/mask_rcnn/incep
}
```

```
IMAGES_FOR_TEST = {
    'Beach' : 'models/research/object_detection/test_images/image2.jpg',
    'Dogs' : 'models/research/object_detection/test_images/image1.jpg',
    # By Heiko Gorski, Source: https://commons.wikimedia.org/wiki/File:Naxos_Taverna.jpg
    'Naxos Taverna' : 'https://upload.wikimedia.org/wikipedia/commons/6/60/Naxos_Taverna.jpg
    # Source: https://commons.wikimedia.org/wiki/File:The_Coleoptera_of_the_British_islands_
    'Beatles' : 'https://upload.wikimedia.org/wikipedia/commons/1/1b/The_Coleoptera_of_the_B
    # By Américo Toledano, Source: https://commons.wikimedia.org/wiki/File:Biblioteca_Maim%C
    'Phones' : 'https://upload.wikimedia.org/wikipedia/commons/thumb/0/0d/Biblioteca_Maim%C3%
    # Source: https://commons.wikimedia.org/wiki/File:The_smaller_British_birds_(8053836633)
    'Birds' : 'https://upload.wikimedia.org/wikipedia/commons/0/09/The_smaller_British_birds.
}
```

```
COCO17_HUMAN_POSE_KEYPOINTS = [(0, 1),
(0, 2),
(1, 3),
(2, 4),
(0, 5),
(0, 6),
(5, 7),
(7, 9),
(6, 8),
(8, 10),
(5, 6),
```

```
(5, 11),
(6, 12),
(11, 12),
(11, 13),
(13, 15),
(12, 14),
(14, 16)]  
  
# Clone the tensorflow models repository
!git clone --depth 1 https://github.com/tensorflow/models  
  
Cloning into 'models'...
remote: Enumerating objects: 3341, done.
remote: Counting objects: 100% (3341/3341), done.
remote: Compressing objects: 100% (2775/2775), done.
remote: Total 3341 (delta 881), reused 1361 (delta 516), pack-reused 0
Receiving objects: 100% (3341/3341), 34.34 MiB | 13.62 MiB/s, done.
Resolving deltas: 100% (881/881), done.  
  
%%bash
sudo apt install -y protobuf-compiler
cd models/research/
protoc object_detection/protos/*.proto --python_out=.
cp object_detection/packages/tf2/setup.py .
python -m pip install .  
  
Collecting hdfs<3.0.0,>=2.1.0
  Downloading hdfs-2.7.0-py3-none-any.whl (34 kB)
Collecting pymongo<4.0.0,>=3.8.0
  Downloading pymongo-3.12.3-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64
Collecting orjson<4.0
  Downloading orjson-3.6.8-cp37-cp37m-manylinux_2_24_x86_64.whl (253 kB)
Collecting requests<3.0.0dev,>=2.18.0
  Downloading requests-2.27.1-py2.py3-none-any.whl (63 kB)
Collecting fastavro<2,>=0.23.6
  Downloading fastavro-1.4.11-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64
Collecting cloudpickle<3,>=2.0.0
  Downloading cloudpickle-2.0.0-py3-none-any.whl (25 kB)
Collecting dill<0.3.2,>=0.3.1.1
  Downloading dill-0.3.1.1.tar.gz (151 kB)
Requirement already satisfied: docopt in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Collecting protobuf>=3.12.0
  Downloading protobuf-3.20.1-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (1.1 MB)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: cycler>=0.10.0 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: opencv-python>=4.1.0.25 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: kiwisolver>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Collecting portalocker
  Downloading portalocker-2.4.0-py2.py3-none-any.whl (16 kB)
Collecting colorama
  Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: regex in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: scikit-learn>=0.21.3 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt)
```

```
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: tensorflow-metadata in /usr/local/lib/python3.7/dis
Requirement already satisfied: promise in /usr/local/lib/python3.7/dist-packages (-
Requirement already satisfied: attrs>=18.1.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.7/dis
Building wheels for collected packages: object-detection, py-cpuinfo, dill, avro-p
  Building wheel for object-detection (setup.py): started
  Building wheel for object-detection (setup.py): finished with status 'done'
  Created wheel for object-detection: filename=object_detection-0.1-py3-none-any.w
  Stored in directory: /tmp/pip-ephem-wheel-cache-g943e_qo/wheels/fa/a4/d2/e9a5057
  Building wheel for py-cpuinfo (setup.py): started
  Building wheel for py-cpuinfo (setup.py): finished with status 'done'
  Created wheel for py-cpuinfo: filename=py_cpuinfo-8.0.0-py3-none-any.whl size=22
  Stored in directory: /root/.cache/pip/wheels/d2/f1/1f/041add21dc9c4220157f1bd2bd
  Building wheel for dill (setup.py): started
  Building wheel for dill (setup.py): finished with status 'done'
  Created wheel for dill: filename=dill-0.3.1.1-py3-none-any.whl size=78544 sha256
  Stored in directory: /root/.cache/pip/wheels/a4/61/fd/c57e374e580aa78a45ed78d585
  Building wheel for avro-python3 (setup.py): started
  Building wheel for avro-python3 (setup.py): finished with status 'done'
  Created wheel for avro-python3: filename=avro_python3-1.10.2-py3-none-any.whl si
  Stored in directory: /root/.cache/pip/wheels/d6/e5/b1/6b151d9b535ee50aaa6ab27d14
  Building wheel for seqeval (setup.py): started
  Building wheel for seqeval (setup.py): finished with status 'done'
  Created wheel for seqeval: filename=seqeval-1.2.2-py3-none-any.whl size=16180 sh
```

```
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.utils import ops as utils_ops

%matplotlib inline

PATH_TO_LABELS = './models/research/object_detection/data/mscoco_label_map.pbtxt'
category_index = label_map_util.create_category_index_from_labelmap(PATH_TO_LABELS, use di
```

Model Selection

model display name: Faster R-CNN Inception ResNet V2 1024x1024

Mostrar código

Selected model: Faster R-CNN Inception ResNet V2 1024x1024
Model Handle at TensorFlow Hub: https://tfhub.dev/tensorflow/faster_rcnn/inception_resnet_v2_atrous_1024x1024/1

```
print('loading model...')  
hub_model = hub.load(model_handle)  
print('model loaded!')
```

loading model...
model loaded!

Image Selection (don't forget to execute the cell!)

selected_image: Dogs



flip_image_horizontally:

convert_image_to_grayscale:

[Mostrar código](#)



```
def alguma_funcao():
    results = hub_model(image_np)

# different object detection models have additional results
# all of them are explained in the documentation
    result = {key:value.numpy() for key,value in results.items()}
    print(result.keys())
```



```
#exemplo de uso da timeit
import timeit

inicio = timeit.default_timer()
alguma_funcao()
fim = timeit.default_timer()
print ('duracao: %f' % (fim - inicio))

dict_keys(['detection_multiclass_scores', 'detection_scores', 'detection_classes', 'r
duracao: 37.408004
```

◀ ▶

////// Resultados:

EfficientDet D0 512x512

```
Praia 6.990672
Cachorro 0.680395
Passaro 1.117323
```

SSD MobileNet V2 FPNLite 320x320

```
Praia 0.592833
Cachorro 4.916549
Passaro 0.656639
```

SSD ResNet50 V1 FPN 640x640 (RetinaNet50)

```
Praia 1.307339
Cachorro 1.294660
Passaro 2.545451
```

Faster R-CNN ResNet50 V1 640x640

```
Praia 4.422644  
Cachorro 9.446927  
Passaro 4.350711
```

Mask R-CNN Inception ResNet V2 1024x1024

```
Praia 37.969243  
Cachorro 37.408004  
Passaro 44.342133
```

```
# running inference
results = hub_model(image_np)

# different object detection models have additional results
# all of them are explained in the documentation
result = {key:value.numpy() for key,value in results.items()}
print(result.keys())

dict_keys(['detection_scores', 'detection_classes', 'detection_keypoint_scores', 'num
          ▶ [REDACTED] ▷
```



```
label_id_offset = 0
image_np_with_detections = image_np.copy()

# Use keypoints if available in detections
keypoints, keypoint_scores = None, None
if 'detection_keypoints' in result:
    keypoints = result['detection_keypoints'][0]
    keypoint_scores = result['detection_keypoint_scores'][0]

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_with_detections[0],
    result['detection_boxes'][0],
    (result['detection_classes'][0] + label_id_offset).astype(int),
    result['detection_scores'][0],
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=200,
    min_score_thresh=.30,
    agnostic_mode=False,
    keypoints=keypoints,
    keypoint_scores=keypoint_scores,
    keypoint_edges=COCO17_HUMAN_POSE_KEYPOINTS)

plt.figure(figsize=(24,32))
plt.imshow(image_np_with_detections[0])
plt.show()
```



▼ ToDo : YoloV3 (30pt)

Carregue o YoloV3 pré-treinado (ver [link](#)) e execute a inferência nas mesmas imagens testadas com os modelos acima. Calule o custo computacional e compare contra os modelos acima.

Qual a sua conclusão? Justifique

ToDo : 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.

▼ Part II - Segmentação (40pt)

▼ ToDo : Rodando um tutorial (15pt)

Estude o tutorial do [link](#). Rode o código e veja o resultado.

```
pip install -q git+https://github.com/tensorflow/examples.git

Building wheel for tensorflow-examples (setup.py) ... done
WARNING: Built wheel for tensorflow-examples is invalid: Metadata 1.2 mandates PEP
          Running setup.py install for tensorflow-examples ... done
DEPRECATION: tensorflow-examples was installed using the legacy 'setup.py install'
```

```
try:
    # %tensorflow_version only exists in Colab.
    %tensorflow_version 2.x
except Exception:
    pass
import tensorflow as tf

from __future__ import absolute_import, division, print_function, unicode_literals

from tensorflow_examples.models.pix2pix import pix2pix

import tensorflow_datasets as tfds
tfds.disable_progress_bar()

from IPython.display import clear_output
import matplotlib.pyplot as plt
```

```
dataset, info = tfds.load('oxford_iiit_pet:3.*.*', with_info=True)

Downloading and preparing dataset oxford_iiit_pet/3.2.0 (download: 773.52 MiB, generated: 1.06 GiB)
Shuffling and writing examples to /root/tensorflow_datasets/oxford_iiit_pet/3.2.0.info
Shuffling and writing examples to /root/tensorflow_datasets/oxford_iiit_pet/3.2.0.info
Dataset oxford_iiit_pet downloaded and prepared to /root/tensorflow_datasets/oxford_iiit_pet/3.2.0

def normalize(input_image, input_mask):
    input_image = tf.cast(input_image, tf.float32) / 255.0
    input_mask -= 1
    return input_image, input_mask

@tf.function
def load_image_train(datapoint):
    input_image = tf.image.resize(datapoint['image'], (128, 128))
    input_mask = tf.image.resize(datapoint['segmentation_mask'], (128, 128))

    if tf.random.uniform() > 0.5:
        input_image = tf.image.flip_left_right(input_image)
        input_mask = tf.image.flip_left_right(input_mask)

    input_image, input_mask = normalize(input_image, input_mask)

    return input_image, input_mask

def load_image_test(datapoint):
    input_image = tf.image.resize(datapoint['image'], (128, 128))
    input_mask = tf.image.resize(datapoint['segmentation_mask'], (128, 128))

    input_image, input_mask = normalize(input_image, input_mask)

    return input_image, input_mask

TRAIN_LENGTH = info.splits['train'].num_examples
BATCH_SIZE = 64
BUFFER_SIZE = 1000
STEPS_PER_EPOCH = TRAIN_LENGTH // BATCH_SIZE

train = dataset['train'].map(load_image_train, num_parallel_calls=tf.data.experimental.AUTOTUNE)
test = dataset['test'].map(load_image_test)

train_dataset = train.cache().shuffle(BUFFER_SIZE).batch(BATCH_SIZE).repeat()
train_dataset = train_dataset.prefetch(buffer_size=tf.data.experimental.AUTOTUNE)
test_dataset = test.batch(BATCH_SIZE)

def display(display_list):
    plt.figure(figsize=(15, 15))
```

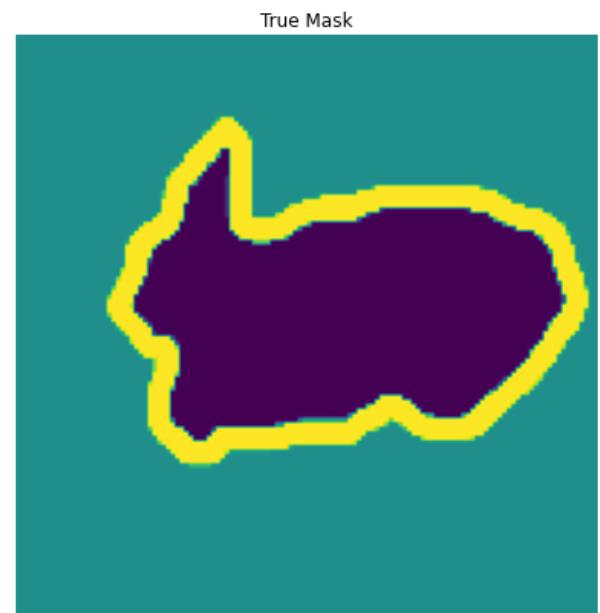
```

title = ['Input Image', 'True Mask', 'Predicted Mask']

for i in range(len(display_list)):
    plt.subplot(1, len(display_list), i+1)
    plt.title(title[i])
    plt.imshow(tf.keras.preprocessing.image.array_to_img(display_list[i]))
    plt.axis('off')
plt.show()

for image, mask in train.take(1):
    sample_image, sample_mask = image, mask
display([sample_image, sample_mask])

```



```
OUTPUT_CHANNELS = 3
```

```

base_model = tf.keras.applications.MobileNetV2(input_shape=[128, 128, 3], include_top=False)

# Use as ativações dessas camadas
layer_names = [
    'block_1_expand_relu',    # 64x64
    'block_3_expand_relu',    # 32x32
    'block_6_expand_relu',    # 16x16
    'block_13_expand_relu',   # 8x8
    'block_16_project',      # 4x4
]
layers = [base_model.get_layer(name).output for name in layer_names]

# Crie o modelo de extração de características
down_stack = tf.keras.Model(inputs=base_model.input, outputs=layers)

```

```
down_stack.trainable = False

    Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mc
9412608/9406464 [=====] - 0s 0us/step
9420800/9406464 [=====] - 0s 0us/step

[< >]

up_stack = [
    pix2pix.upsample(512, 3), # 4x4 -> 8x8
    pix2pix.upsample(256, 3), # 8x8 -> 16x16
    pix2pix.upsample(128, 3), # 16x16 -> 32x32
    pix2pix.upsample(64, 3), # 32x32 -> 64x64
]

def unet_model(output_channels):

    # Esta é a última camada do modelo
    last = tf.keras.layers.Conv2DTranspose(
        output_channels, 3, strides=2,
        padding='same', activation='softmax') #64x64 -> 128x128

    inputs = tf.keras.layers.Input(shape=[128, 128, 3])
    x = inputs

    # Downampling através do modelo
    skips = down_stack(x)
    x = skips[-1]
    skips = reversed(skips[:-1])

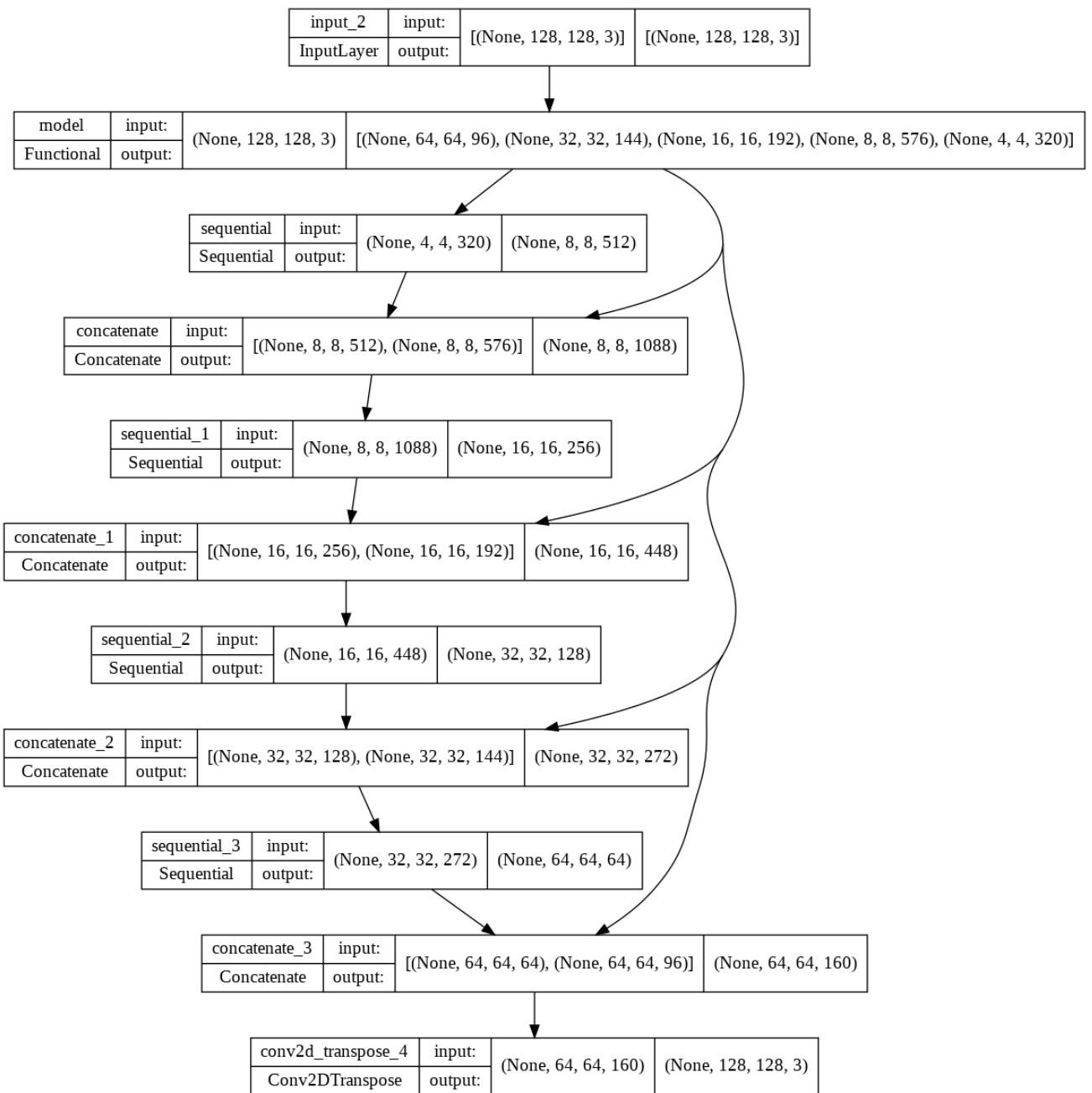
    # Upsampling e estabelecimento das conexões de salto
    for up, skip in zip(up_stack, skips):
        x = up(x)
        concat = tf.keras.layers.concatenate()
        x = concat([x, skip])

    x = last(x)

    return tf.keras.Model(inputs=inputs, outputs=x)

model = unet_model(OUTPUT_CHANNELS)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

tf.keras.utils.plot_model(model, show_shapes=True)
```



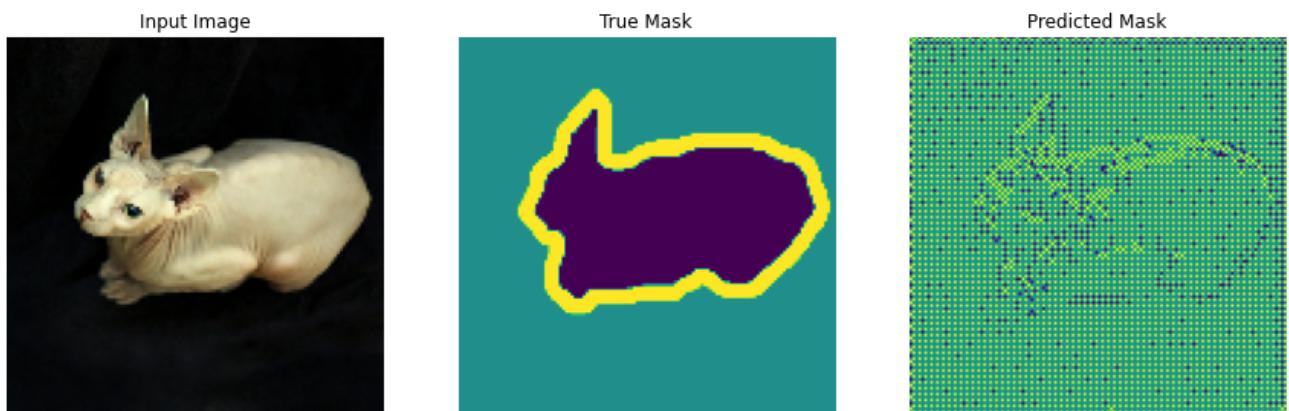
```

def create_mask(pred_mask):
    pred_mask = tf.argmax(pred_mask, axis=-1)
    pred_mask = pred_mask[..., tf.newaxis]
    return pred_mask[0]

def show_predictions(dataset=None, num=1):
    if dataset:
        for image, mask in dataset.take(num):
            pred_mask = model.predict(image)
            display([image[0], mask[0], create_mask(pred_mask)])
    else:
        display([sample_image, sample_mask,
                 create_mask(model.predict(sample_image[tf.newaxis, ...]))])

show_predictions()

```



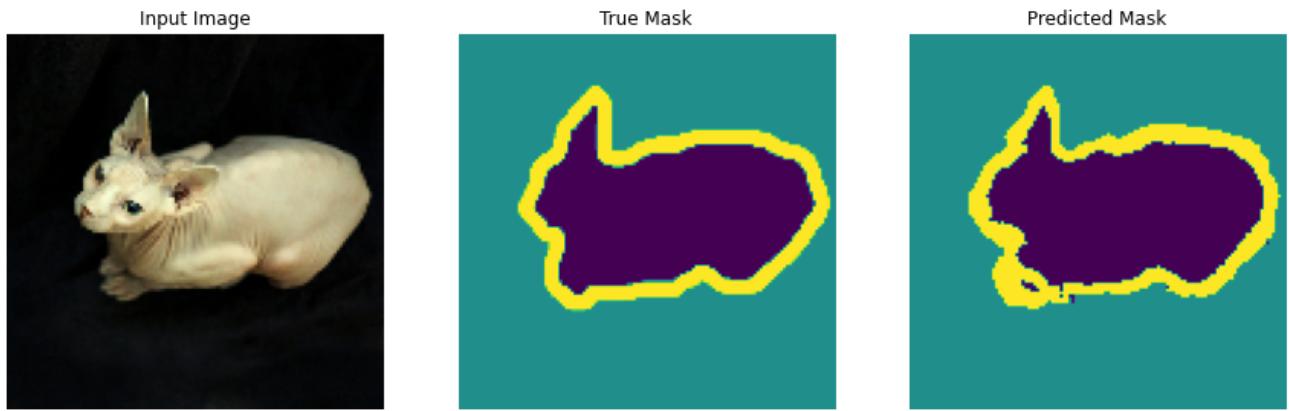
```

class DisplayCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        clear_output(wait=True)
        show_predictions()
        print ('\nSample Prediction after epoch {} \n'.format(epoch+1))

EPOCHS = 20
VAL_SUBSPLITS = 5
VALIDATION_STEPS = info.splits['test'].num_examples//BATCH_SIZE//VAL_SUBSPLITS

model_history = model.fit(train_dataset, epochs=EPOCHS,
                          steps_per_epoch=STEPS_PER_EPOCH,
                          validation_steps=VALIDATION_STEPS,
                          validation_data=test_dataset,
                          callbacks=[DisplayCallback()])

```



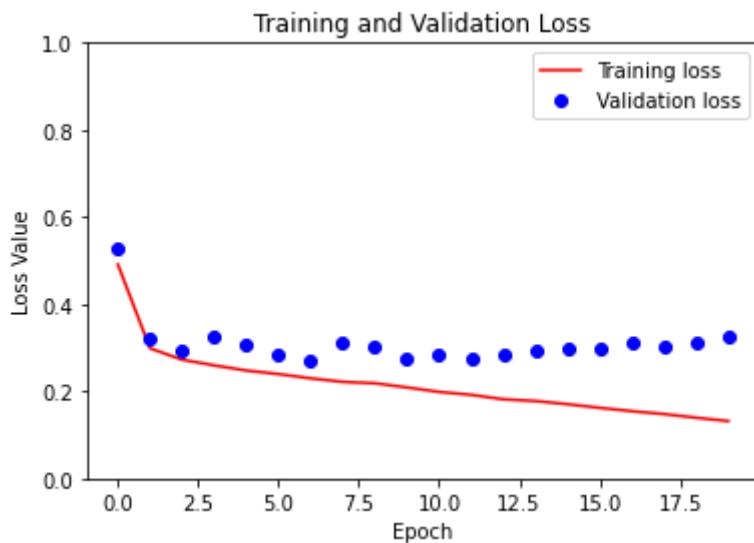
Sample Prediction after epoch 20

```
57/57 [=====] - 345s 6s/step - loss: 0.1317 - accuracy: 0.9:
```

```
loss = model_history.history['loss']
val_loss = model_history.history['val_loss']

epochs = range(EPOCHS)

plt.figure()
plt.plot(epochs, loss, 'r', label='Training loss')
plt.plot(epochs, val_loss, 'bo', label='Validation loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss Value')
plt.ylim([0, 1])
plt.legend()
plt.show()
```



```
show_predictions(test_dataset, 3)
```



▼ ToDo : Melhorando o modelo (25pt)

Use das operações que você conhece para construir uma rede melhor:

- Dropout
- Convolution2D, Dense, (várias funções de ativação como GeLu, LeakyReLu, etc)
- Flatten, GlobalAveragePooling2D, GlobalMaxPooling2D, etc.
- Use outras arquiteturas para o encoder: inception, Xception, VGG16, EfficientNet

