FootballSemanticSeg

March 3, 2023

[1]:
```python
# Common
import os
import keras
import numpy as np
import pandas as pd
import tensorflow as tf

# Data
from glob import glob
from tqdm import tqdm
import tensorflow.image as tfi
from tensorflow.keras.utils import load_img, img_to_array

# Data Visualization
import matplotlib.pyplot as plt

# Model
from tensorflow.keras.layers import add
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Layer
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import multiply
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Concatenate
from tensorflow.keras.layers import Conv2DTranspose
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.callbacks import Callback, ModelCheckpoint
from tensorflow.keras.models import Model

# Model Visualization
from tensorflow.keras.utils import plot_model
```

```
2023-03-03 01:50:45.899752: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
```

```
compiler flags.
2023-03-03 01:50:48.372280: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot
open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/Qt5/5.
15.2/gcc_64/lib:/opt/vtkqt5/lib:/opt/pcgal/lib:/opt/pcl/lib:/opt/ospray/lib:/opt
/mrpt/lib:/opt/opencv/lib:/opt/cgal/lib:/opt/boost/lib:/usr/local/vxl/lib:/usr/l
ocal/vtk8/lib:/usr/local/vtk/lib:/usr/local/opencv/lib:/usr/local/visp/lib:/usr/
local/pcl/lib:/usr/local/opencv/lib:/usr/local/mrpt/lib:/usr/local/cgal/lib:/usr
/local/boost/lib:/usr/local/cuda-12.0/lib64:/usr/local/cuda-12.0/extras/CUPTI/li
b64:/usr/local/cuda/lib64:/usr/local/cuda/lib64:/usr/local/cuda-12.0/lib64:/usr/
local/cuda-12.0/lib64:/opt/Qt5/5.15.2/gcc_64/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr
/local/jdk20/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr/local/opencv/lib:/opt/pcgal/lib
:/usr/local/vtk8/lib:/usr/local/boost/lib:/usr/local/pcl/lib:/usr/local/vtk8/lib
:/opt/pcgal/lib:/opt/Qt5/5.15.2/gcc_64/lib:/opt/Qt5/5.15.2/gcc_64/plugins/platfo
rms:/opt/Qt5/5.15.2/gcc_64/plugins/platformthemes:/usr/local/mrpt/lib:/usr/local
/opencv/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr/local/itk/lib:/opt/ParaView/lib:/opt
/vtkqt5/lib/java/vtk-Linux-x86_64:/opt/ospray/lib:/opt/vtkqt5/lib:/usr/local/jdk
20/lib:/usr/local/jdk20/lib:/opt/sofa/plugins:/opt/pcl/lib:/opt/vtkqt5/lib/opt/s
ofa/lib:/opt/pcl/lib:/opt/vtkqt5/lib:/opt/pcl/lib:/opt/vtkqt5/lib:/opt/Qt5/5.15.
2/gcc_64/lib:/opt/pcgal/lib:/usr/local/vtk8/lib:/usr/local/pcl/lib:/opt/cloudcom
pare/lib:/opt/opencv/lib:/opt/cgal/lib:/opt/mrpt/lib:/opt/Qt5/5.15.2/gcc_64/lib:
/opt/boostu/lib:/opt/pcgal/lib:/home/picox/uopt/Qt5.12/5.12.12/gcc_64/gcc_64/lib
:/usr/local/cuda-12.0/lib64:/usr/local/cuda/lib64:/opt/cuda/lib64:/home/picox/uo
pt/anaconda3/lib/:/home/picox/uopt/anaconda3/lib/python3.9/site-packages/tensorr
t/:/home/picox/uopt/anaconda3/envs/tf/lib:/home/picox/uopt/anaconda3/envs/tf/lib
/python3.9/site-packages/tensorrt/
2023-03-03 01:50:48.372303: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore above
cudart dlerror if you do not have a GPU set up on your machine.
2023-03-03 01:50:53.839534: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot
open shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/Qt5/5.
15.2/gcc_64/lib:/opt/vtkqt5/lib:/opt/pcgal/lib:/opt/pcl/lib:/opt/ospray/lib:/opt
/mrpt/lib:/opt/opencv/lib:/opt/cgal/lib:/opt/boost/lib:/usr/local/vxl/lib:/usr/l
ocal/vtk8/lib:/usr/local/vtk/lib:/usr/local/opencv/lib:/usr/local/visp/lib:/usr/
local/pcl/lib:/usr/local/opencv/lib:/usr/local/mrpt/lib:/usr/local/cgal/lib:/usr
/local/boost/lib:/usr/local/cuda-12.0/lib64:/usr/local/cuda-12.0/extras/CUPTI/li
b64:/usr/local/cuda/lib64:/usr/local/cuda/lib64:/usr/local/cuda-12.0/lib64:/usr/
local/cuda-12.0/lib64:/opt/Qt5/5.15.2/gcc_64/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr
/local/jdk20/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr/local/opencv/lib:/opt/pcgal/lib
:/usr/local/vtk8/lib:/usr/local/boost/lib:/usr/local/pcl/lib:/usr/local/vtk8/lib
:/opt/pcgal/lib:/opt/Qt5/5.15.2/gcc_64/lib:/opt/Qt5/5.15.2/gcc_64/plugins/platfo
rms:/opt/Qt5/5.15.2/gcc_64/plugins/platformthemes:/usr/local/mrpt/lib:/usr/local
/opencv/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr/local/itk/lib:/opt/ParaView/lib:/opt
/vtkqt5/lib/java/vtk-Linux-x86_64:/opt/ospray/lib:/opt/vtkqt5/lib:/usr/local/jdk
20/lib:/usr/local/jdk20/lib:/opt/sofa/plugins:/opt/pcl/lib:/opt/vtkqt5/lib/opt/s
```

```
ofa/lib:/opt/pcl/lib:/opt/vtkqt5/lib:/opt/pcl/lib:/opt/vtkqt5/lib:/opt/Qt5/5.15.
2/gcc_64/lib:/opt/pcgal/lib:/usr/local/vtk8/lib:/usr/local/pcl/lib:/opt/cloudcom
pare/lib:/opt/opencv/lib:/opt/cgal/lib:/opt/mrpt/lib:/opt/Qt5/5.15.2/gcc_64/lib:
/opt/boostu/lib:/opt/pcgal/lib:/home/picox/uopt/Qt5.12/5.12.12/gcc_64/gcc_64/lib
:/usr/local/cuda-12.0/lib64:/usr/local/cuda/lib64:/opt/cuda/lib64:/home/picox/uo
pt/anaconda3/lib/:/home/picox/uopt/anaconda3/lib/python3.9/site-packages/tensorr
t/:/home/picox/uopt/anaconda3/envs/tf/lib:/home/picox/uopt/anaconda3/envs/tf/lib
/python3.9/site-packages/tensorrt/
2023-03-03 01:50:53.839821: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: /opt/Qt5/5.15.2/gcc_64/lib:/opt/vtkqt5/lib:/opt/pcga
l/lib:/opt/pcl/lib:/opt/ospray/lib:/opt/mrpt/lib:/opt/opencv/lib:/opt/cgal/lib:/
opt/boost/lib:/usr/local/vxl/lib:/usr/local/vtk8/lib:/usr/local/vtk/lib:/usr/loc
al/opencv/lib:/usr/local/visp/lib:/usr/local/pcl/lib:/usr/local/opencv/lib:/usr/
local/mrpt/lib:/usr/local/cgal/lib:/usr/local/boost/lib:/usr/local/cuda-12.0/lib
64:/usr/local/cuda-12.0/extras/CUPTI/lib64:/usr/local/cuda/lib64:/usr/local/cuda
/lib64:/usr/local/cuda-12.0/lib64:/usr/local/cuda-12.0/lib64:/opt/Qt5/5.15.2/gcc
_64/lib:/opt/Qt5/5.15.2/gcc_64/lib:/usr/local/jdk20/lib:/opt/Qt5/5.15.2/gcc_64/l
ib:/usr/local/opencv/lib:/opt/pcgal/lib:/usr/local/vtk8/lib:/usr/local/boost/lib
:/usr/local/pcl/lib:/usr/local/vtk8/lib:/opt/pcgal/lib:/opt/Qt5/5.15.2/gcc_64/li
b:/opt/Qt5/5.15.2/gcc_64/plugins/platforms:/opt/Qt5/5.15.2/gcc_64/plugins/platfo
rmthemes:/usr/local/mrpt/lib:/usr/local/opencv/lib:/opt/Qt5/5.15.2/gcc_64/lib:/u
sr/local/itk/lib:/opt/ParaView/lib:/opt/vtkqt5/lib/java/vtk-Linux-x86_64:/opt/os
pray/lib:/opt/vtkqt5/lib:/usr/local/jdk20/lib:/usr/local/jdk20/lib:/opt/sofa/plu
gins:/opt/pcl/lib:/opt/vtkqt5/lib/opt/sofa/lib:/opt/pcl/lib:/opt/vtkqt5/lib:/opt
/pcl/lib:/opt/vtkqt5/lib:/opt/Qt5/5.15.2/gcc_64/lib:/opt/pcgal/lib:/usr/local/vt
k8/lib:/usr/local/pcl/lib:/opt/cloudcompare/lib:/opt/opencv/lib:/opt/cgal/lib:/o
pt/mrpt/lib:/opt/Qt5/5.15.2/gcc_64/lib:/opt/boostu/lib:/opt/pcgal/lib:/home/pico
x/uopt/Qt5.12/5.12.12/gcc_64/gcc_64/lib:/usr/local/cuda-12.0/lib64:/usr/local/cu
da/lib64:/opt/cuda/lib64:/home/picox/uopt/anaconda3/lib/:/home/picox/uopt/anacon
da3/lib/python3.9/site-packages/tensorrt/:/home/picox/uopt/anaconda3/envs/tf/lib
:/home/picox/uopt/anaconda3/envs/tf/lib/python3.9/site-packages/tensorrt/
2023-03-03 01:50:53.839840: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot
dlopen some TensorRT libraries. If you would like to use Nvidia GPU with
TensorRT, please make sure the missing libraries mentioned above are installed
properly.
```

[2]:
```python
# Root Path
image_paths = 'Football/images/'

# All Images and thier respective maps
all_images = glob(image_paths + "*.jpg")
all_paths = [path.replace(".jpg",".jpg___fuse.png") for path in all_images]
all_images[0]
```

[2]: 'Football/images/Frame 1 (71).jpg'

[3]:
```
def load_image(path, SIZE=256):
    image = load_img(path)
    image = tfi.resize(image, (SIZE, SIZE))
    image = img_to_array(image)
    image = tf.cast(image, tf.float32)
    image = image/255.
    return image

def load_data(image_paths, label_paths, SIZE=256):
    images, label_maps = np.zeros(shape=(len(image_paths), SIZE, SIZE, 3)), np.
 ↪zeros(shape=(len(label_paths), SIZE, SIZE, 3))
    for i, (image_path, label_path) in tqdm(enumerate(zip(image_paths,␣
 ↪label_paths)), desc="Loading"):
        image, label_map = load_image(image_path, SIZE=SIZE),␣
 ↪load_image(label_path, SIZE=SIZE)
        images[i], label_maps[i] = image, label_map
    return images, label_maps
```

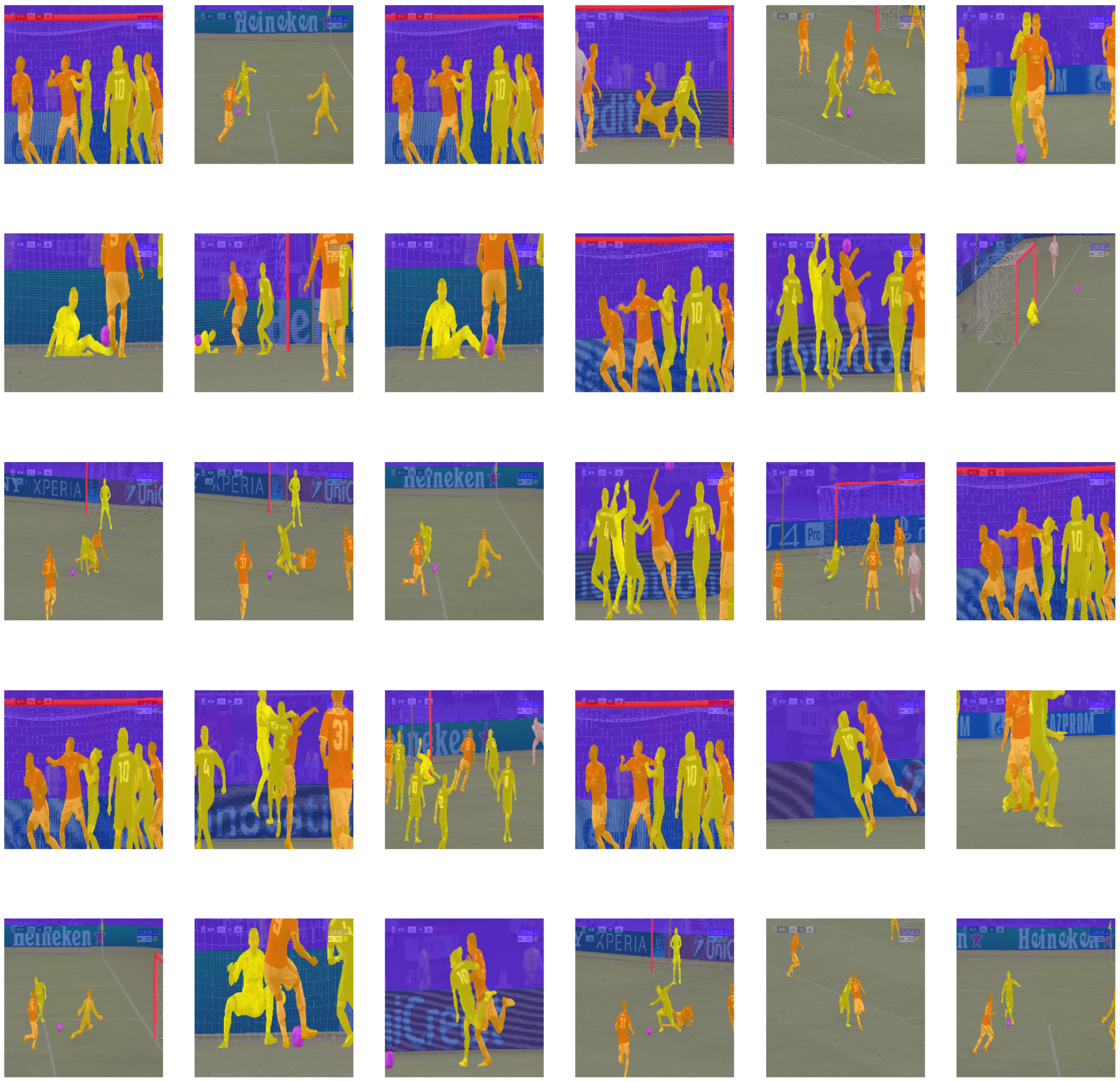
[4]:
```
images, label_maps = load_data(all_images, all_paths)
```

```
Loading: 0it [00:00, ?it/s]2023-03-03 01:51:01.554135: E
tensorflow/compiler/xla/stream_executor/cuda/cuda_driver.cc:267] failed call to
cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
2023-03-03 01:51:01.554288: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_diagnostics.cc:156] kernel
driver does not appear to be running on this host (picox):
/proc/driver/nvidia/version does not exist
2023-03-03 01:51:01.582139: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
Loading: 100it [00:06, 15.64it/s]
```

[5]:
```
def show_map(image, label_map, alpha_1=1, alpha_2=0.7):
    plt.imshow(image, alpha=alpha_1)
    plt.imshow(label_map, alpha=alpha_2)
    plt.axis('off')
```

[6]:
```
def show_maps(images, label_maps, GRID=[5,6], SIZE=(25,25)):

    # Plot Configuration
    n_rows, n_cols = GRID
    n_images = n_rows * n_cols
    plt.figure(figsize=SIZE)

    # Iterate through the Data
    i=1
    for image, label_map in zip(images, label_maps):
        # Plot Image
        plt.subplot(n_rows, n_cols, i)
        show_map(image, label_map)

        i+=1
        if i>n_images:
            break

    # Final Plot
    plt.show()
```

[7]:
```
show_maps(images, label_maps)
```



```
[8]: class Encoder(Layer):

         def __init__(self, filters, rate=0.2, pooling=True, **kwargs):
             super(Encoder, self).__init__(**kwargs)

             self.filters = filters
             self.rate = rate
             self.pooling = pooling

             self.bn = BatchNormalization()
             self.c1 = Conv2D(filters, kernel_size=3, strides=1, padding='same',␣
      ↪activation='relu', kernel_initializer='he_normal')
             self.drop = Dropout(rate)
```

```
        self.c2 = Conv2D(filters, kernel_size=3, strides=1, padding='same',␣
↪activation='relu', kernel_initializer='he_normal')
        self.pool = MaxPool2D()

    def call(self, X):
        x = self.c2(self.drop(self.c1(self.bn(X))))
        if self.pooling:
            y = self.pool(x)
            return x, y
        return x

    def get_config(self):
        base_config = super().get_config()
        return {**base_config, "filters":self.filters, "rate":self.
↪rate,"pooling":self.pooling}
```

[9]:
```
class Decoder(Layer):

    def __init__(self, filters, rate, **kwargs):
        super(Decoder, self).__init__(**kwargs)

        self.filters = filters
        self.rate = rate

        self.cT = Conv2DTranspose(filters, kernel_size=3, strides=2,␣
↪padding='same', activation='relu', kernel_initializer='he_normal')
        self.bn = BatchNormalization()
        self.skip = Concatenate()
        self.net = Encoder(filters, rate, pooling=False)

    def call(self, X):
        x, skip_x = X
        y = self.cT(self.bn(x))
        y = self.net(self.skip([y, skip_x]))
        return y

    def get_config(self):
        base_config = super().get_config()
        return {**base_config, "filters":self.filters, "rate":self.rate}
```

[10]:
```
# Input Layer
InputL = Input(shape=(256,256,3), name="InputImage")

# Encoder Block
c1, p1 = Encoder(filters=64,  rate=0.1, name="Encoder1")(InputL)
c2, p2 = Encoder(filters=128, rate=0.1, name="Encoder2")(p1)
c3, p3 = Encoder(filters=256, rate=0.2, name="Encoder3")(p2)
```

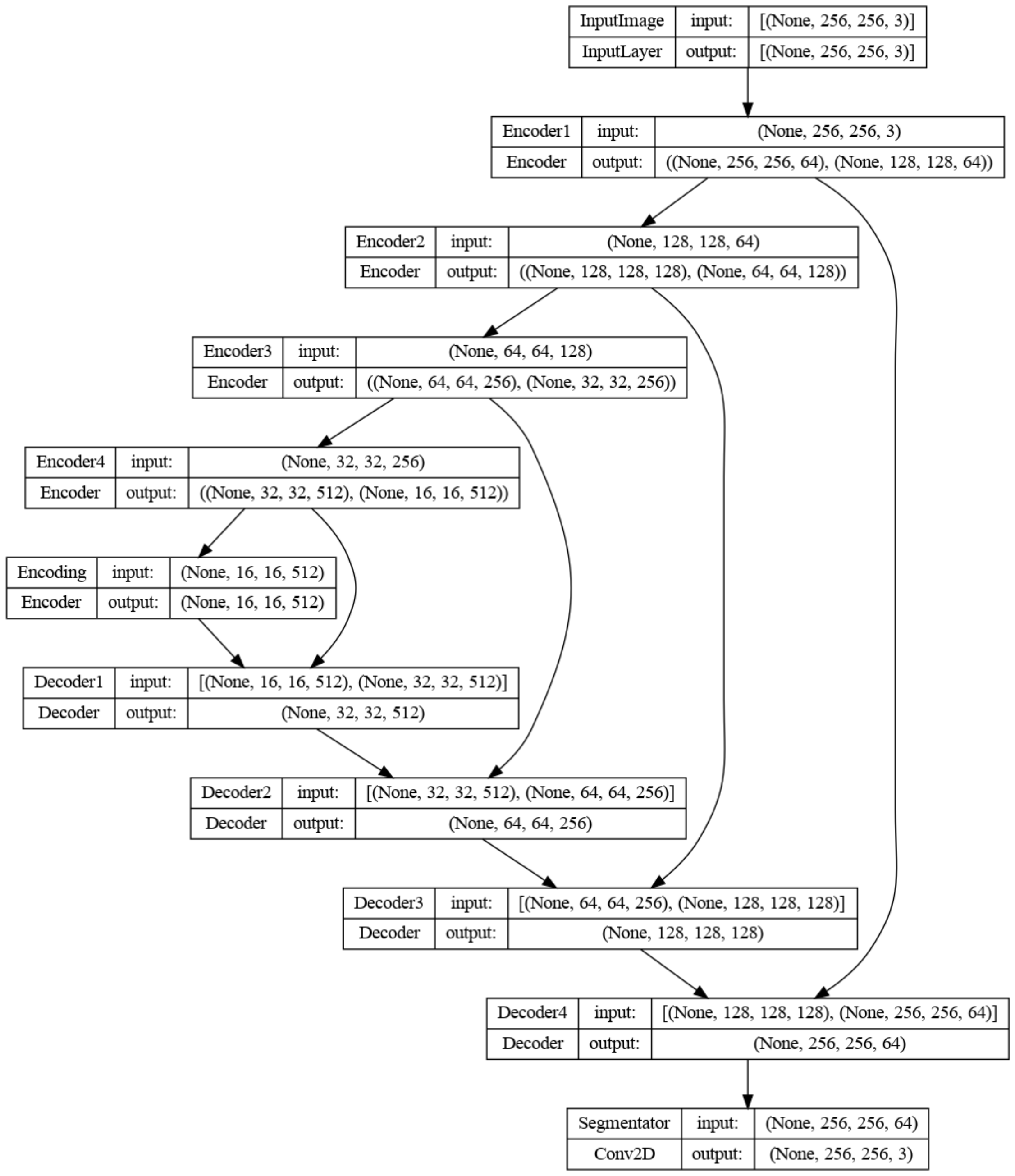
```
c4, p4 = Encoder(filters=512, rate=0.2, name="Encoder4")(p3)

# Encoding Layer
encodings = Encoder(filters=512, rate=0.3, pooling=False, name="Encoding")(p4)

# Decoder Block
d = Decoder(512, 0.2, name='Decoder1')([encodings, c4])
d = Decoder(256, 0.2, name='Decoder2')([d, c3])
d = Decoder(128, 0.1, name='Decoder3')([d, c2])
d = Decoder(64, 0.1, name='Decoder4')([d, c1])

# Output
conv_out = Conv2D(3, kernel_size=3, padding='same', activation='sigmoid',␣
 ↪name="Segmentator")(d)
```

[11]:
```
# Model
model = Model(InputL, conv_out, name="UNet")
model.summary()

# Compile Model
model.compile(loss='binary_crossentropy', optimizer='adam')
```

```
Model: "UNet"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 InputImage (InputLayer)        [(None, 256, 256, 3  0           []
                                )]

 Encoder1 (Encoder)             ((None, 256, 256, 6  38732
['InputImage[0][0]']

                                4),
                                 (None, 128, 128, 6
                                4))

 Encoder2 (Encoder)             ((None, 128, 128, 1  221696
['Encoder1[0][1]']

                                28),
                                 (None, 64, 64, 128
                                ))

 Encoder3 (Encoder)             ((None, 64, 64, 256  885760
['Encoder2[0][1]']

                                ),
                                 (None, 32, 32, 256
                                 ))

 Encoder4 (Encoder)              ((None, 32, 32, 512  3540992
['Encoder3[0][1]']
                                 ),
                                  (None, 16, 16, 512
                                 ))

 Encoding (Encoder)              (None, 16, 16, 512)  4721664
['Encoder4[0][1]']

 Decoder1 (Decoder)              (None, 32, 32, 512)  9444864
['Encoding[0][0]',
'Encoder4[0][0]']

 Decoder2 (Decoder)              (None, 64, 64, 256)  2953984
['Decoder1[0][0]',
'Encoder3[0][0]']

 Decoder3 (Decoder)              (None, 128, 128, 12  739712
['Decoder2[0][0]',
                                 8)

'Encoder2[0][0]']

 Decoder4 (Decoder)              (None, 256, 256, 64  185536
['Decoder3[0][0]',
                                 )

'Encoder1[0][0]']

 Segmentator (Conv2D)            (None, 256, 256, 3)  1731
['Decoder4[0][0]']

================================================================================
==================
Total params: 22,734,671
Trainable params: 22,726,089
Non-trainable params: 8,582
________________________________________________________________________________
__________________
```

[12]:
```python
plot_model(model, "UNet.png", show_shapes=True)
```

[12]:


InputImage | input: | [(None, 256, 256, 3)]
InputLayer | output: | [(None, 256, 256, 3)]
Encoder1 | input: | (None, 256, 256, 3)
Encoder | output: | ((None, 256, 256, 64), (None, 128, 128, 64))
Encoder2 | input: | (None, 128, 128, 64)
Encoder | output: | ((None, 128, 128, 128), (None, 64, 64, 128))
Encoder3 | input: | (None, 64, 64, 128)
Encoder | output: | ((None, 64, 64, 256), (None, 32, 32, 256))
Encoder4 | input: | (None, 32, 32, 256)
Encoder | output: | ((None, 32, 32, 512), (None, 16, 16, 512))
Encoding | input: | (None, 16, 16, 512)
Encoder | output: | (None, 16, 16, 512)
Decoder1 | input: | [(None, 16, 16, 512), (None, 32, 32, 512)]
Decoder | output: | (None, 32, 32, 512)
Decoder2 | input: | [(None, 32, 32, 512), (None, 64, 64, 256)]
Decoder | output: | (None, 64, 64, 256)
Decoder3 | input: | [(None, 64, 64, 256), (None, 128, 128, 128)]
Decoder | output: | (None, 128, 128, 128)
Decoder4 | input: | [(None, 128, 128, 128), (None, 256, 256, 64)]
Decoder | output: | (None, 256, 256, 64)
Segmentator | input: | (None, 256, 256, 64)
Conv2D | output: | (None, 256, 256, 3)

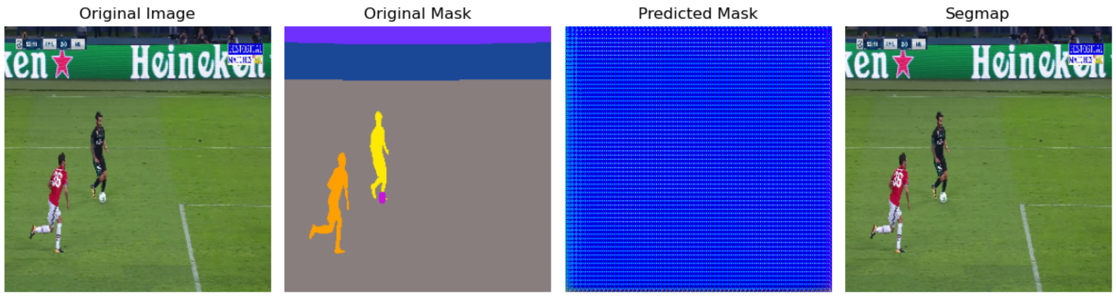
[13]:
```
BATCH_SIZE = 16
SPE = len(images)//BATCH_SIZE

def show_image(image, title=None):
    plt.imshow(image)
    plt.title(title)
    plt.axis('off')
```

[17]:
```
import numpy as np
from PIL import Image
import cv2

class ShowProgress(Callback):
    def on_epoch_end(self, epoch, logs=None):
        id = np.random.randint(len(images))
        image = images[id]
        mask = label_maps[id]
        pred_mask = self.model(tf.expand_dims(image,axis=0))[0]

        plt.figure(figsize=(12,10))
        plt.subplot(1,4,1)
        show_image(image, title="Original Image")

        plt.subplot(1,4,2)
        show_image(mask, title="Original Mask")

        plt.subplot(1,4,3)
        show_image(pred_mask, title="Predicted Mask")


        plt.subplot(1,4,4)
        plt.imshow( np.asarray(image) )
        masked_img = np.ma.masked_where(image == 0, image)
        show_image(masked_img, title="Segmap")

        plt.tight_layout()
        plt.show()
```

[18]:
```
cbs = [ModelCheckpoint("UNetFootPlaySegment.h5",␣
 ↪save_best_only=True),ShowProgress()]
```

[19]:
```
nepochs=1
model.fit( images, label_maps,validation_split=0.1,epochs=nepochs,␣
 ↪batch_size=BATCH_SIZE,steps_per_epoch=SPE, callbacks=cbs)
```

```
6/6 [==============================] - ETA: 0s - loss: 0.6512
```



```
6/6 [==============================] - 327s 55s/step - loss: 0.6512 - val_loss:
22.2010
```

[19]:

```
<keras.callbacks.History at 0x7f89cf59fa30>
```

[ ]:

[ ]: