part-2

Model Builtding : Brain MRI Segmentation

1. Dependencies

```
!pip install keras-unet-collection
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting keras-unet-collection
  Downloading keras_unet_collection-0.1.13-py3-none-any.whl (67 kB)
     |████████████████████████████████| 67 kB 6.3 MB/s
Installing collected packages: keras-unet-collection
Successfully installed keras-unet-collection-0.1.13
```

```
!pip install -U segmentation-models==1.0.1
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting segmentation-models==1.0.1
  Downloading segmentation_models-1.0.1-py3-none-any.whl (33 kB)
Collecting image-classifiers==1.0.0
  Downloading image_classifiers-1.0.0-py3-none-any.whl (19 kB)
Collecting efficientnet==1.0.0
  Downloading efficientnet-1.0.0-py3-none-any.whl (17 kB)
Collecting keras-applications<=1.0.8,>=1.0.7
  Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
     |████████████████████████████████| 50 kB 8.1 MB/s
Requirement already satisfied: scikit-image in /usr/local/lib/python3.7/dist-packages (from efficientnet==1.0.0->segmentation-m
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from keras-applications<=1.0.8,>=1.0.7->segmenta
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras-applications<=1.0.8,>=1.0.7->
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (from h5py->keras-applications<=1.0.8,
```

```
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,>=4.3.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->effi
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->efficien
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image->efficientnet==
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->efficientnet==1.0.0-
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image->efficientnet==1.
Requirement already satisfied: scipy>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image->efficientnet==1.0.0->
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->efficientnet==1.0.0
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-i
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplot
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->sci
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib!
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.1->matplotlib!=3.0.0
Installing collected packages: keras-applications, image-classifiers, efficientnet, segmentation-models
Successfully installed efficientnet-1.0.0 image-classifiers-1.0.0 keras-applications-1.0.8 segmentation-models-1.0.1
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from os import path
import cv2
import os
import re
import random
import pdb
import seaborn as sns

import tensorflow as tf
import keras

from keras.utils.layer_utils import get_source_inputs
import segmentation_models as sm
sm.set_framework('tf.keras')
tf.keras.backend.set_image_data_format('channels_last')
from segmentation_models import Unet

# from tensorflow.keras import Input
```

```
from tensorflow.keras.layers import Input, Activation, BatchNormalization, Dropout, Lambda, Conv2D, Conv2DTranspose, MaxPooling2D, co
from tensorflow.keras.optimizers import Adam
from keras.models import Model, load_model, save_model

from sklearn.model_selection import train_test_split

from segmentation_models.metrics import iou_score
from segmentation_models.losses import DiceLoss
from tensorflow.keras.utils import plot_model
os.environ['TF_FORCE_GPU_ALLOW_GROWTH'] = 'true'

from keras_unet_collection import models
```

```
Segmentation Models: using `keras` framework.
```

```
# https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation
!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
```

```
--2022-10-19 02:52:03--  https://storage.googleapis.com/kaggle-data-sets/181273/407317/bundle/archive.zip?X-Goog-Algorithm=GOOG
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.194.128, 142.251.10.128, 142.251.12.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.194.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 748584920 (714M) [application/zip]
Saving to: ‘archive.zip’

archive.zip         100%[===================>] 713.91M  40.5MB/s    in 18s

2022-10-19 02:52:22 (39.9 MB/s) - ‘archive.zip’ saved [748584920/748584920]
```

```
!unzip "/content/archive.zip"
```

Streaming output truncated to the last 5000 lines.

```
  inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7294_19890104/TCGA_DU_7294_19890104_9_mask.tif
  inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_1.tif
  inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_10.tif
```

```
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_10_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_11.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_11_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_12.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_12_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_13.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_13_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_14.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_14_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_15.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_15_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_16.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_16_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_17.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_17_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_18.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_18_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_19.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_19_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_1_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_2.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_20.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_20_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_21.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_21_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_22.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_22_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_23.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_23_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_24.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_24_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_25.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_25_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_26.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_26_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_27.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_27_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_28.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_28_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_29.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_29_mask.tif
```

```
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_2_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_3.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_30.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_30_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_31.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_31_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_32.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_32_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_3_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_4.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_4_mask.tif
inflating: lgg-mri-segmentation/kaggle_3m/TCGA_DU_7298_19910324/TCGA_DU_7298_19910324_5.tif
```

2. data loading

```
# main_dir = "/content/kaggle_3m/"
# for i in os.listdir(main_dir)[:-1]:
#     sub_dir = main_dir+i+"/"
#     if path.isdir(i+"/"+sub_dir):
#         for j in os.listdir(sub_dir):
#             print("\'{}\'".format(sub_dir+j))
```

```
def return_file_name(root_dir):
    img_url = []
    mask_img_url = []
    df = pd.DataFrame([])
    for i in os.listdir(root_dir):
        sub_dir = root_dir+i+"/"
        # pdb.set_trace()
        if path.isdir(sub_dir):
            for j in os.listdir(sub_dir):
    #             img_dir.append(str(sub_dir+j))
```

```
                if "mask" in j :
                    mask_img_url.append(str(sub_dir+j))

    # pdb.set_trace()
    img_url = [re.sub("_mask","", i) for i in mask_img_url]

    df["image"] = img_url
    df["mask"] = mask_img_url

    return  df

main_dir = "/content/kaggle_3m/"
data = return_file_name(main_dir)




"""
    checking all file path
"""
b = True
for i in range(data.shape[0]):
    if not path.isfile(data["image"][i]):
        b = False
    elif not path.isfile(data["mask"][i]):
        b = False

print(b)
```

```
True
```

```
data.head()
```

	image	mask
0	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...
1	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...
2	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...
3	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...	/content/kaggle_3m/TCGA_DU_8164_19970111/TCGA_...

```
data.shape
```

```
(3929, 2)
```

4. Data Preprocessing

while doing EDA we observe there are some images which doesn't contain any information(black image) so we will remove those images

```
def garbage_img_prepross(df):
    """
        finding image which doesn't have much infomation
        here we choose 30 as pixel value threshhold  all img which maximum pixel value
        is less that 30 considered to be garbage image
    """
    thres = 30
    temp_img = []

    for i in df["image"]:
        val = np.max(cv2.imread(i))
        if val <thres:
            temp_img.append(i)

    temp_img = np.array(temp_img)
    df = df[~df["image"].isin(temp_img)]
```

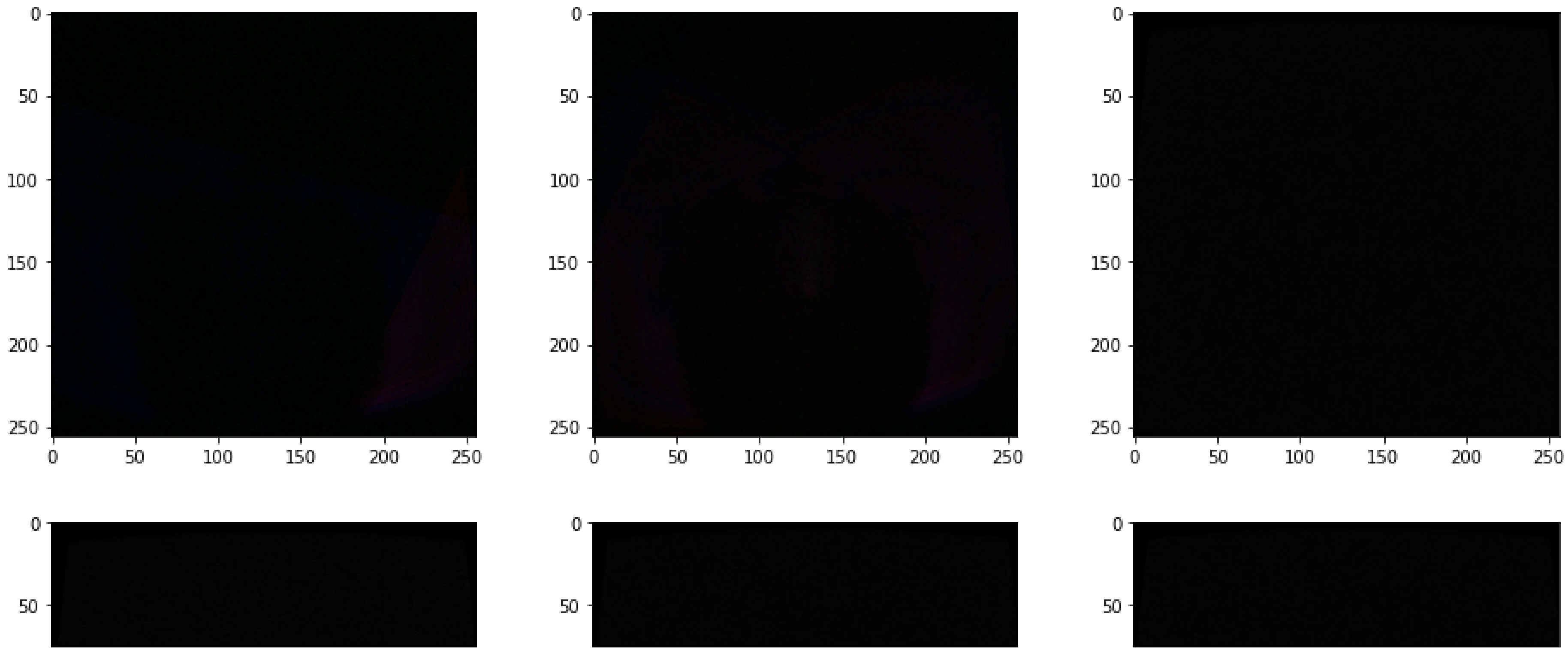
```
    return df, temp_img

data, temp_img = garbage_img_prepross(data)


print("we got total {} image which doesn't contains information, so in order to prepross, we removed it from data".format(len(temp_im
```

```
we got total 89 image which doesn't contains information, so in order to prepross, we removed it from data
```

```
data.shape
```

```
(3840, 2)
```

```
# visulization garbase images which having less information (max picxel val < 30)

r = 2
c = 3
fig,axis = plt.subplots(r, c, figsize=(16,10))
p = 0
for j in range(r):
    for  i in range(c):
        axis[j,i].imshow(cv2.imread(temp_img[82-p]))
        p +=1
```



these image in our train data doesn't have any information so as next move we will remove it from our train data so that our model learn better

NOW, going ahead, we resizes of all images(will done in data pipeline), remove garbage images, now our data is ready for next task - (data pipeline, augmentation , modeling)

```
# data.to_pickle("preprocessed_data.pkl")
# data = pd.read_pickle("/content/preprocessed_data.pkl")
```

Data Pipeline

```
!nvidia-smi
```

```
Wed Oct 19 02:52:44 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
!nvidia-smi
```

```
Wed Oct 19 02:52:45 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
```

```
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```python
from sklearn.model_selection import train_test_split

X_train, X_test = train_test_split(data, test_size=0.20, random_state=42)
X_train = X_train.reset_index(drop=True)
X_test = X_test.reset_index(drop=True)


# 1. creating image generator - through url
from tensorflow.keras.preprocessing.image import ImageDataGenerator


datagen=ImageDataGenerator(rescale=1./255.,
                           rotation_range=0.2, zoom_range=0.1, horizontal_flip=True,
                                width_shift_range=0.05,
                                height_shift_range=0.05,
                                shear_range=0.05, fill_mode='nearest')

mask_datagen=ImageDataGenerator(rescale=1./255.,
                           rotation_range=0.2, zoom_range=0.1, horizontal_flip=True,
                                width_shift_range=0.05,
                                height_shift_range=0.05,
                                shear_range=0.05, fill_mode='nearest')

val_datagen=ImageDataGenerator(rescale=1./255. )

val_mask_datagen=ImageDataGenerator(rescale=1./255. )
```

```python
BATCH_SIZE = 16
train_generator=datagen.flow_from_dataframe(dataframe=X_train, x_col='image',
                                            color_mode = 'rgb', class_mode=None,
                                            target_size=(256,256),batch_size=BATCH_SIZE,
                                            seed=42, shuffle=True)

train_mask_generator = mask_datagen.flow_from_dataframe(dataframe=X_train,
                                                        x_col='mask',
                                                        batch_size=BATCH_SIZE,
                                                        class_mode=None,
                                                        target_size=(256, 256),
                                                        seed=42,
                                                        color_mode='grayscale')

val_image_generator = val_datagen.flow_from_dataframe(dataframe=X_test, x_col='image',
                                                batch_size=BATCH_SIZE, seed=42,
                                                shuffle=True,  color_mode='rgb',
                                                class_mode=None,target_size=(256,256))

val_mask_generator = val_mask_datagen.flow_from_dataframe(dataframe=X_test, x_col='mask',
                                                batch_size=BATCH_SIZE, seed=42,
                                                shuffle=True,  color_mode='grayscale',
                                                class_mode=None,target_size=(256,256))



def data_iterator(image_generator, mask_generator):
    while True:
        X, Y = next(image_generator), next(mask_generator)
        yield X, Y

def data_generator(train_image_generator, train_mask_generator, val_image_generator, val_mask_generator):
    return data_iterator(train_image_generator, train_mask_generator), data_iterator(val_image_generator, val_mask_generator)

train_data_loader, val_data_loader = data_generator(train_generator, train_mask_generator, val_image_generator, val_mask_generator)
```

```
Found 3072 validated image filenames.
Found 3072 validated image filenames.
Found 768 validated image filenames.
Found 768 validated image filenames.
```

callbacks

```
%load_ext tensorboard
```

```
# define callbacks for learning rate scheduling and best checkpoints saving
import datetime
from tensorflow.keras.callbacks import ModelCheckpoint

def create_callback_lists(name = ""):
    filepath='best_model_with_{}.hdf5'.format(name)
    checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_iou_score', verbose=1, save_best_only=True, mode='max')

    learning_rt = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', min_lr=0.0001,patience=1)
    # !rm -rf ./logs/
    log_dir="logs/fit/" + datetime.datetime.now().strftime("%Y_%m_%d-%H_%M")

    early_stop_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=6)

    tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True)

    return [early_stop_callback, checkpoint,tensorboard_callback, learning_rt]

# callback_list = create_callback_lists()
```

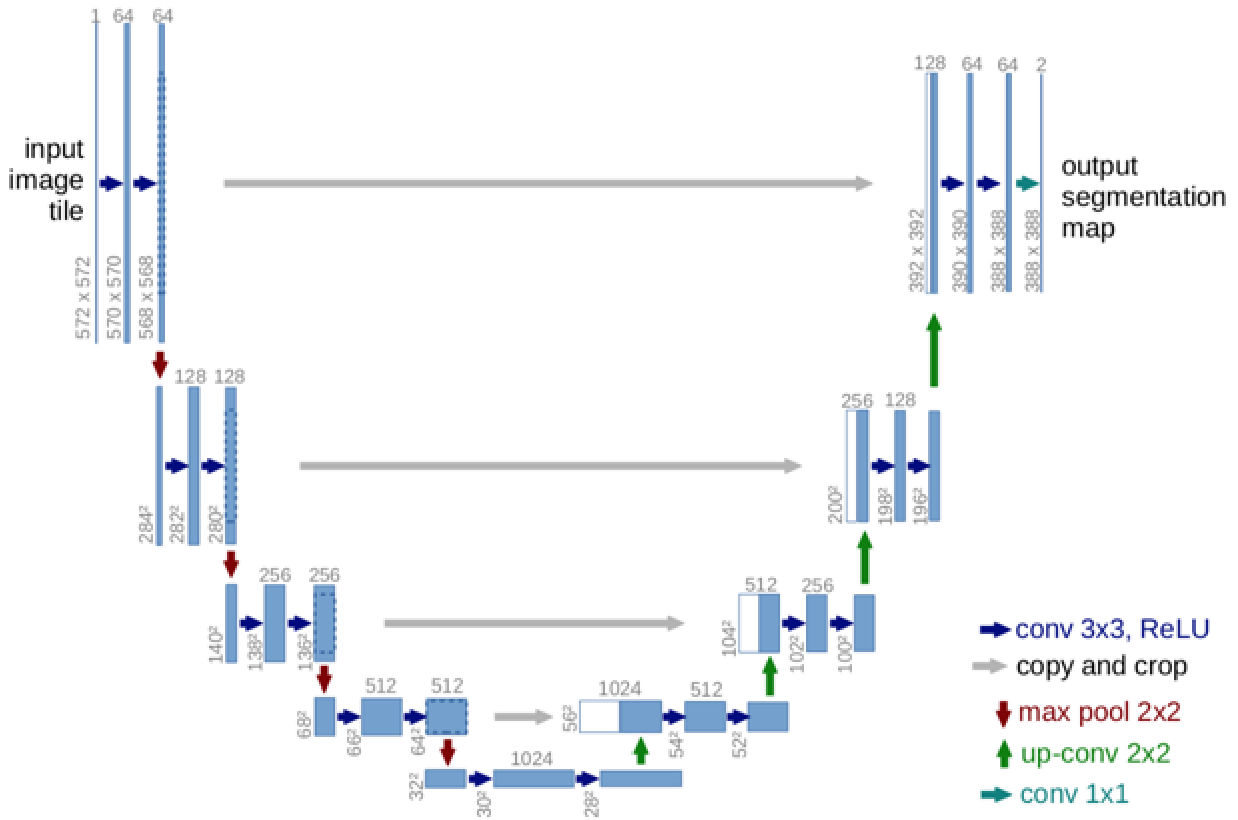
5. Modeling

UNET model

UNET architecture:

it looks like a U shape. it has a contract path (or Encoder) on the left and an expansive path(or decoder) in right. the goal of the contracting path is to find the information of objects in the image and the expansive path is doing pixel-wise prediction based on localization information, getting through the skip connection (from the encoder). here in the encoder part we are reducing the size of images and increasing the size of the filter and in the decoder part, we are doubling the size of images (with help of localization information) and reducing the size of the filter.

for more you can check out its research paper: [check this]



```python
# refer - https://github.com/morteza89/Brain-Tumor-Segmentation/blob/0dfde85bec40d8a0379777b581ea78ceaabff32c/MRI-Brain_Segmentation.

dropout_rate=0.1


def conv2d_block( inputs, filters, kernel_size, batchnorm=False):
        """
        This function creates a convolutional block consisting of two convolutional layers
        and an optional batch normalization layer.
        """
        # first layer
        x = Conv2D(filters, kernel_size=(kernel_size, kernel_size), kernel_initializer='he_normal', padding='same')(inputs)
        x = Activation('relu')(x)
        if batchnorm:
            x = BatchNormalization()(x)
        # second layer
        x = Conv2D(filters, kernel_size=(kernel_size, kernel_size), kernel_initializer='he_normal', padding='same')(x)
        x = Activation('relu')(x)
        if batchnorm:
            x = BatchNormalization()(x)
        return x

def build_UNET(ImgHieght = None, ImgWidth = None, Channels = None, batch_norm = False):

    inputs = Input((ImgHieght, ImgWidth, Channels))
    # first layer
    x = conv2d_block(inputs, 64, 3, batchnorm=batch_norm)
    # encoder side of the UNET
    enc1 = conv2d_block(x, 64, 3, batchnorm=batch_norm)
    pol1 = MaxPooling2D((2, 2))(enc1)
    drp1 = Dropout(dropout_rate)(pol1)
    # second layer
    enc2 = conv2d_block(drp1, 128, 3, batchnorm=batch_norm)
    pol2 = MaxPooling2D((2, 2))(enc2)
    drp2 = Dropout(dropout_rate)(pol2)
    # third layer
    enc3 = conv2d_block(drp2, 256, 3, batchnorm=batch_norm)
```

```python
    pol3 = MaxPooling2D((2, 2))(enc3)
    drp3 = Dropout(dropout_rate)(pol3)
    # fourth layer
    enc4 = conv2d_block(drp3, 512, 3, batchnorm=batch_norm)
    pol4 = MaxPooling2D((2, 2))(enc4)
    drp4 = Dropout(dropout_rate)(pol4)
    # fifth layer or the bottleneck
    enc5 = conv2d_block(drp4, 1024, 3, batchnorm=batch_norm)
    # decoder side of the UNET
    # pdb.set_trace()
    # 1
    dec1 = Conv2DTranspose(512, 3, strides=2, padding='same', name = "dec1_transpose")(enc5)
    dec2 = conv2d_block(concatenate([dec1, enc4]), 512, 3, batchnorm=batch_norm)
    dec2 = Dropout(dropout_rate)(dec2)
    # 2
    dec2 = Conv2DTranspose(256, 3, strides=2, padding='same',name = "dec2_transpose")(dec2)
    dec3 = conv2d_block(concatenate([dec2, enc3]), 256, 3, batchnorm=batch_norm)
    dec3 = Dropout(dropout_rate)(dec3)
    # 3
    dec3 = Conv2DTranspose(128, 3, strides=2, padding='same', name = "dec3_transpose")(dec3)
    dec4 = conv2d_block(concatenate([dec3, enc2]), 128, 3, batchnorm=batch_norm)
    dec4 = Dropout(dropout_rate)(dec4)
    # 4
    dec4 = Conv2DTranspose(64, 3, strides=2, padding='same', name = "dec4_transpose")(dec4)
    dec5 = conv2d_block(concatenate([dec4, enc1]), 32, 3, batchnorm=batch_norm)
    dec5 = Dropout(dropout_rate)(dec5)
    # final layer
    outputs = Conv2D(1, (1, 1), activation='sigmoid')(dec5)
    model = Model(inputs=[inputs], outputs=[outputs])
    return model
```

```python
model = build_UNET(ImgHieght = 256, ImgWidth = 256, Channels = 3, batch_norm = True)
```

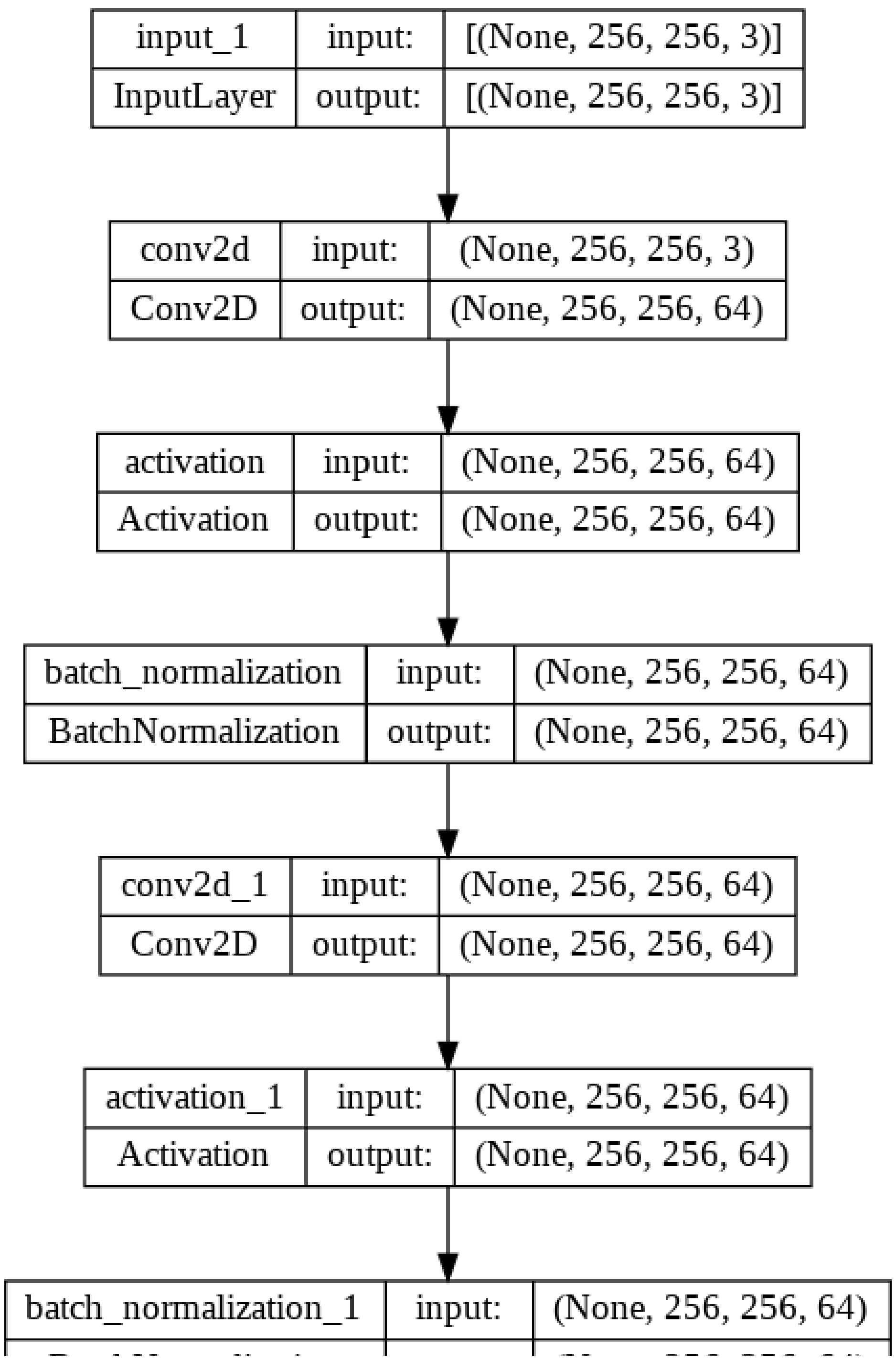
```python
model.summary()
```

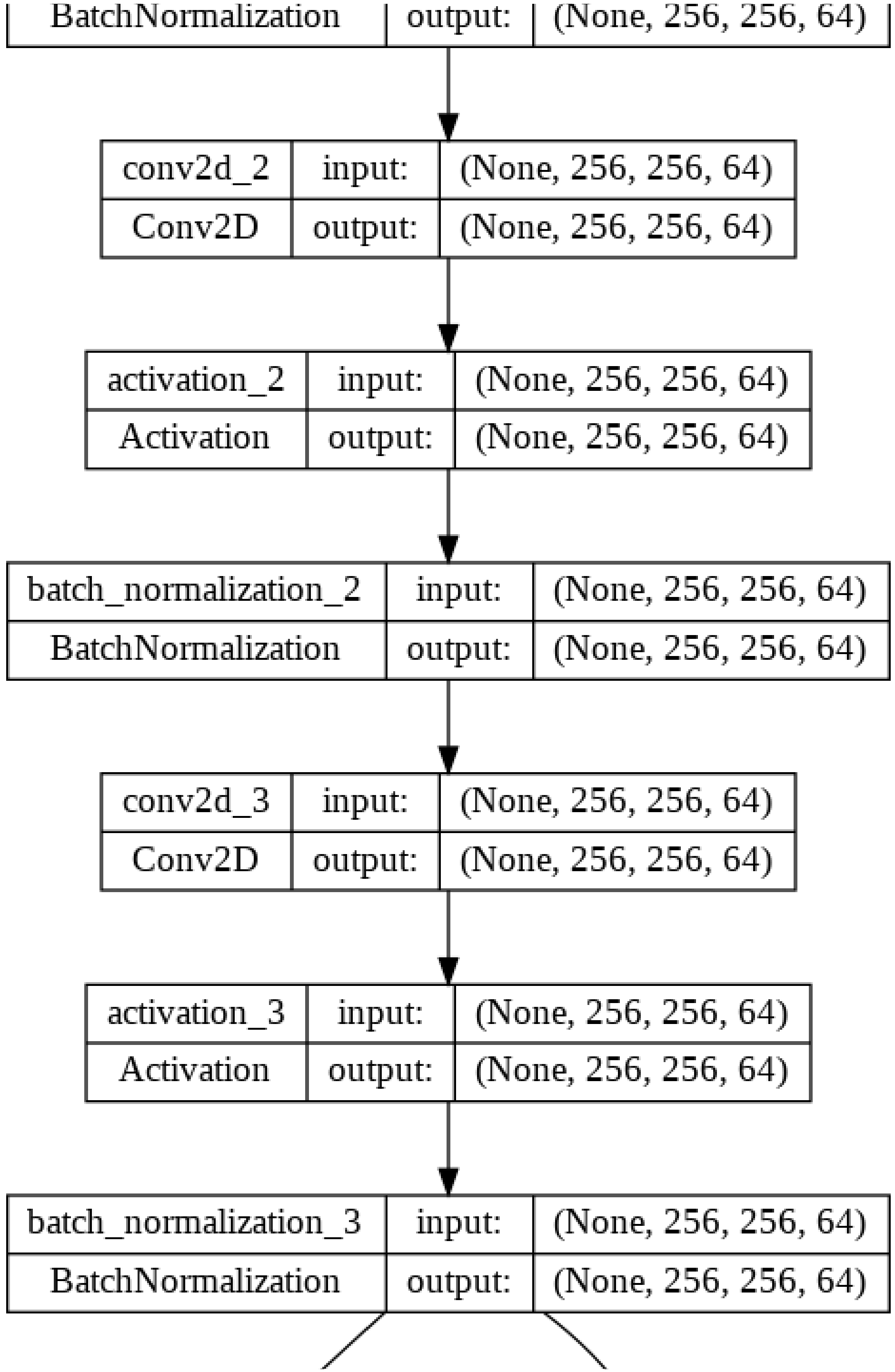
```
Model: "model"
```

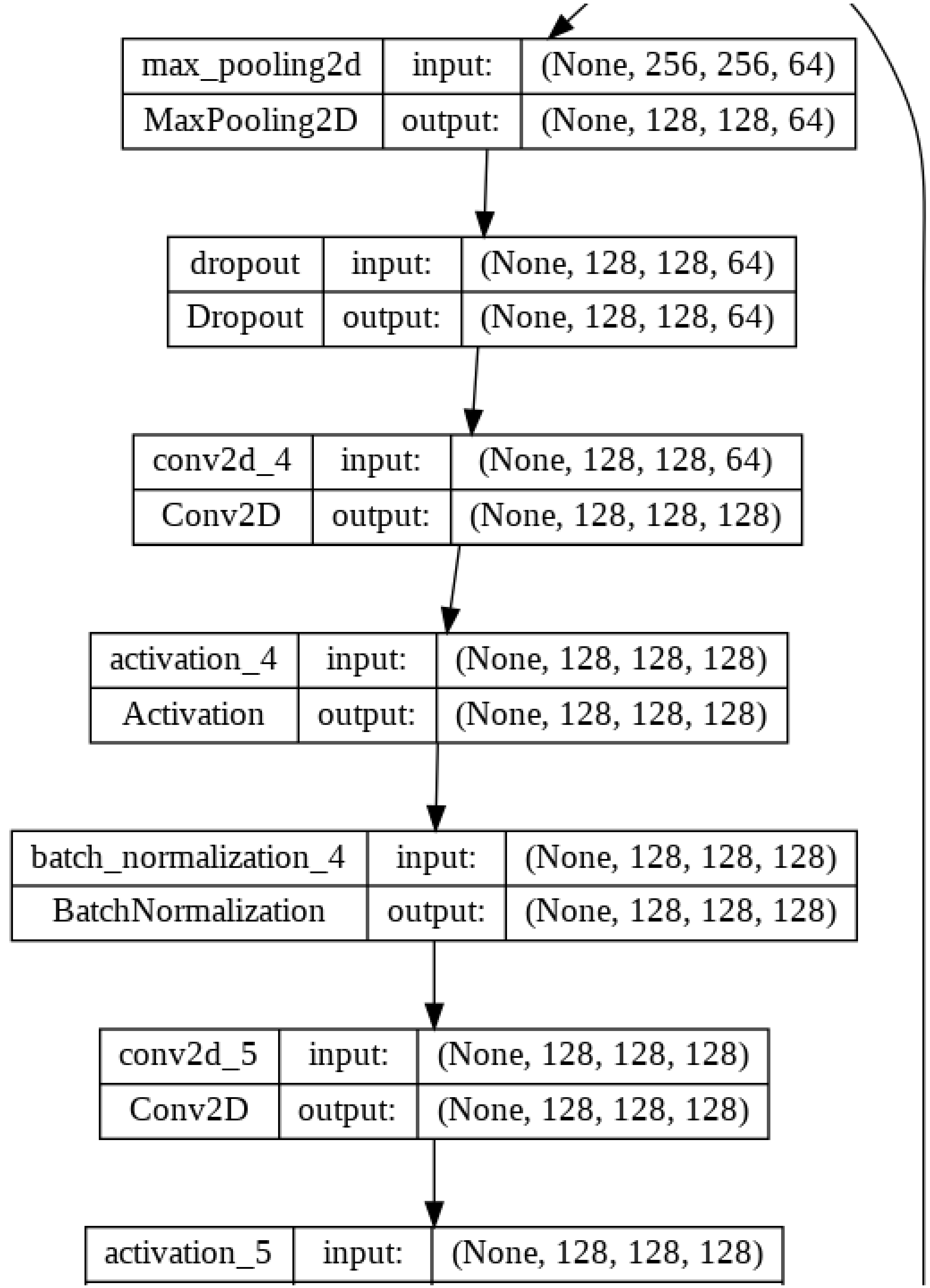
```
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 conv2d (Conv2D)                (None, 256, 256, 64  1792        ['input_1[0][0]']
                                )

 activation (Activation)        (None, 256, 256, 64  0           ['conv2d[0][0]']
                                )

 batch_normalization (BatchNorm  (None, 256, 256, 64  256        ['activation[0][0]']
 alization)                     )

 conv2d_1 (Conv2D)              (None, 256, 256, 64  36928       ['batch_normalization[0][0]']
                                )

 activation_1 (Activation)      (None, 256, 256, 64  0           ['conv2d_1[0][0]']
                                )

 batch_normalization_1 (BatchNo  (None, 256, 256, 64  256        ['activation_1[0][0]']
 rmalization)                   )

 conv2d_2 (Conv2D)              (None, 256, 256, 64  36928       ['batch_normalization_1[0][0]']
                                )

 activation_2 (Activation)      (None, 256, 256, 64  0           ['conv2d_2[0][0]']
                                )

 batch_normalization_2 (BatchNo  (None, 256, 256, 64  256        ['activation_2[0][0]']
 rmalization)                   )

 conv2d_3 (Conv2D)              (None, 256, 256, 64  36928       ['batch_normalization_2[0][0]']
                                )

 activation_3 (Activation)      (None, 256, 256, 64  0           ['conv2d_3[0][0]']
                                )

 batch_normalization_3 (BatchNo  (None, 256, 256, 64  256        ['activation_3[0][0]']
 rmalization)                   )
```

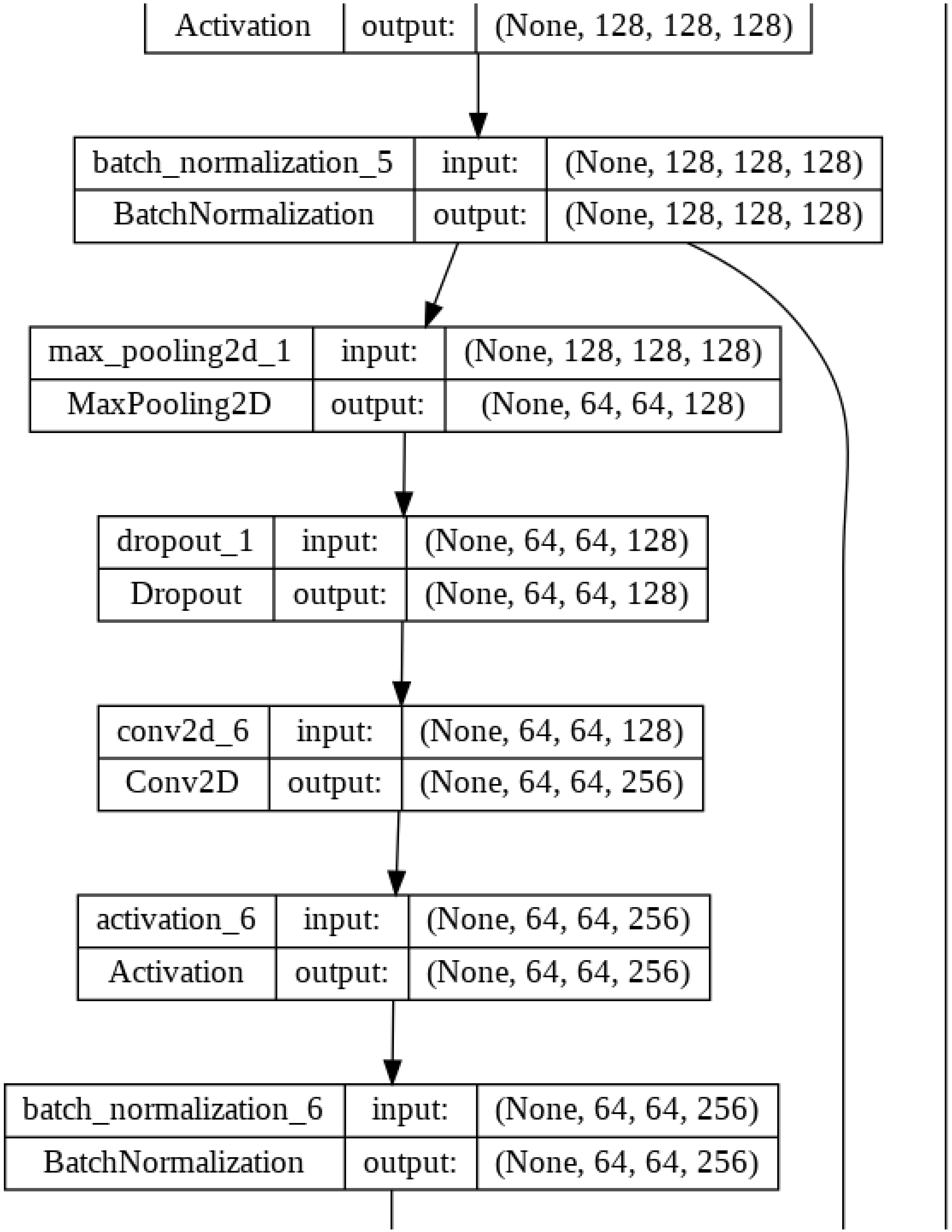
```
 max_pooling2d (MaxPooling2D)   (None, 128, 128, 64  0           ['batch_normalization_3[0][0]']
                                )

 dropout (Dropout)              (None, 128, 128, 64  0           ['max_pooling2d[0][0]']
                                )

 conv2d_4 (Conv2D)              (None, 128, 128, 12  73856       ['dropout[0][0]']
                                8)

 activation_4 (Activation)      (None, 128, 128, 12  0           ['conv2d_4[0][0]']
                                8)

 batch_normalization_4 (BatchNo  (None, 128, 128, 12  512        ['activation_4[0][0]']
 rmalization)                   8)
```

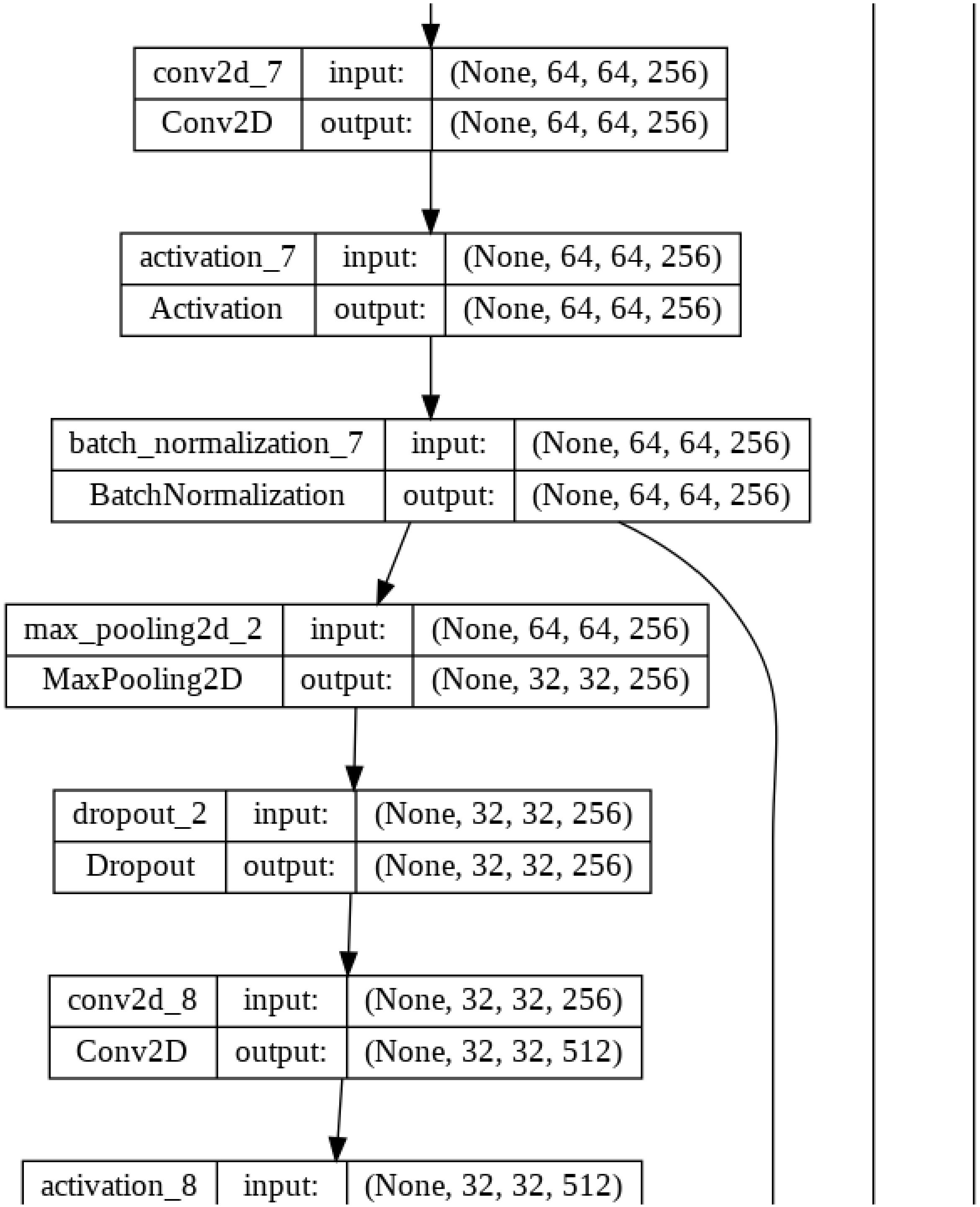
```
plot_model(model, show_shapes=True)
```

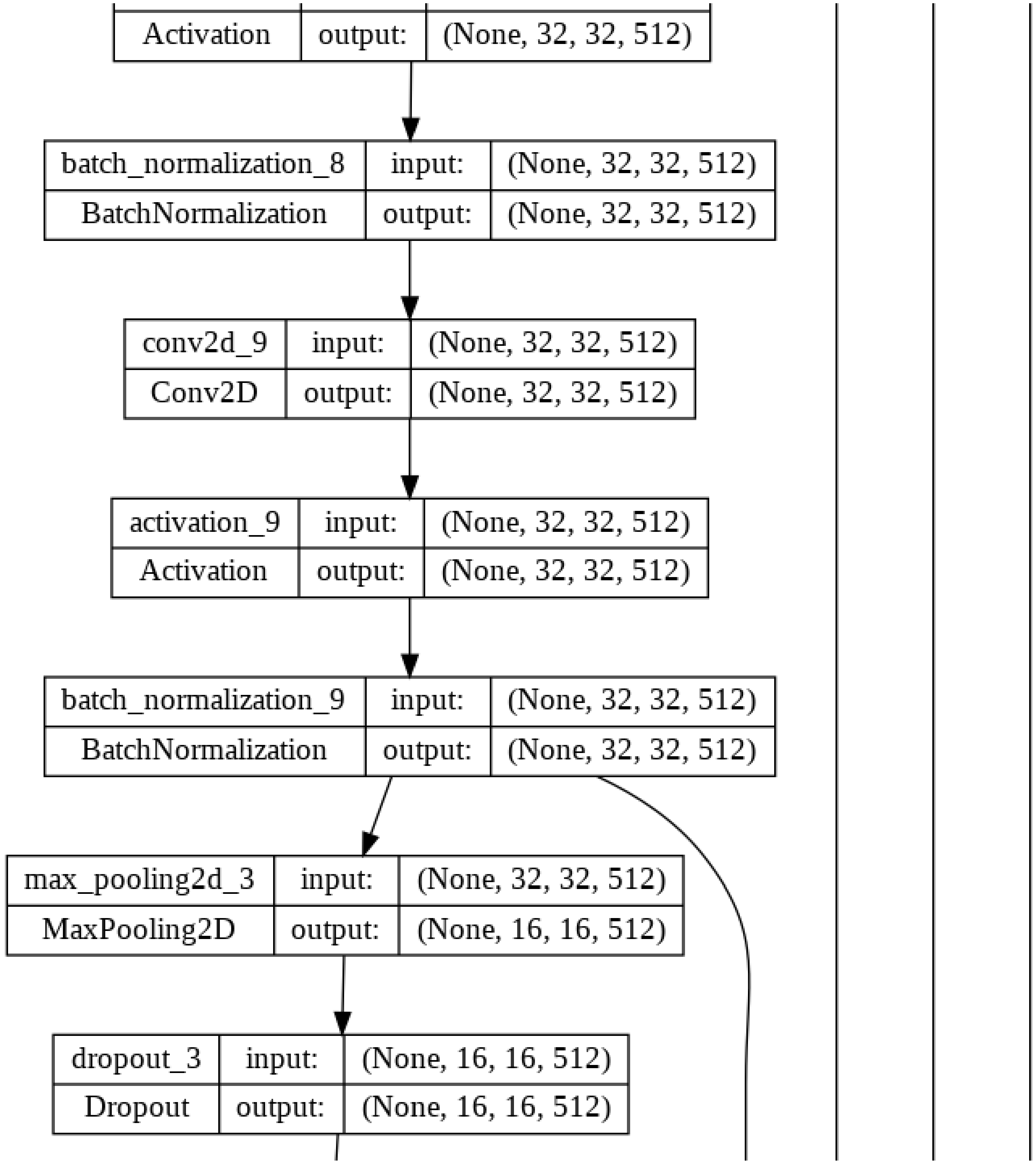

input_1
input:
[(None, 256, 256, 3)]
InputLayer
output:
[(None, 256, 256, 3)]
conv2d
input:
(None, 256, 256, 3)
Conv2D
output:
(None, 256, 256, 64)
activation
input:
(None, 256, 256, 64)
Activation
output:
(None, 256, 256, 64)
batch_normalization
input:
(None, 256, 256, 64)
BatchNormalization
output:
(None, 256, 256, 64)
conv2d_1
input:
(None, 256, 256, 64)
Conv2D
output:
(None, 256, 256, 64)
activation_1
input:
(None, 256, 256, 64)
Activation
output:
(None, 256, 256, 64)
batch_normalization_1
input:
(None, 256, 256, 64)

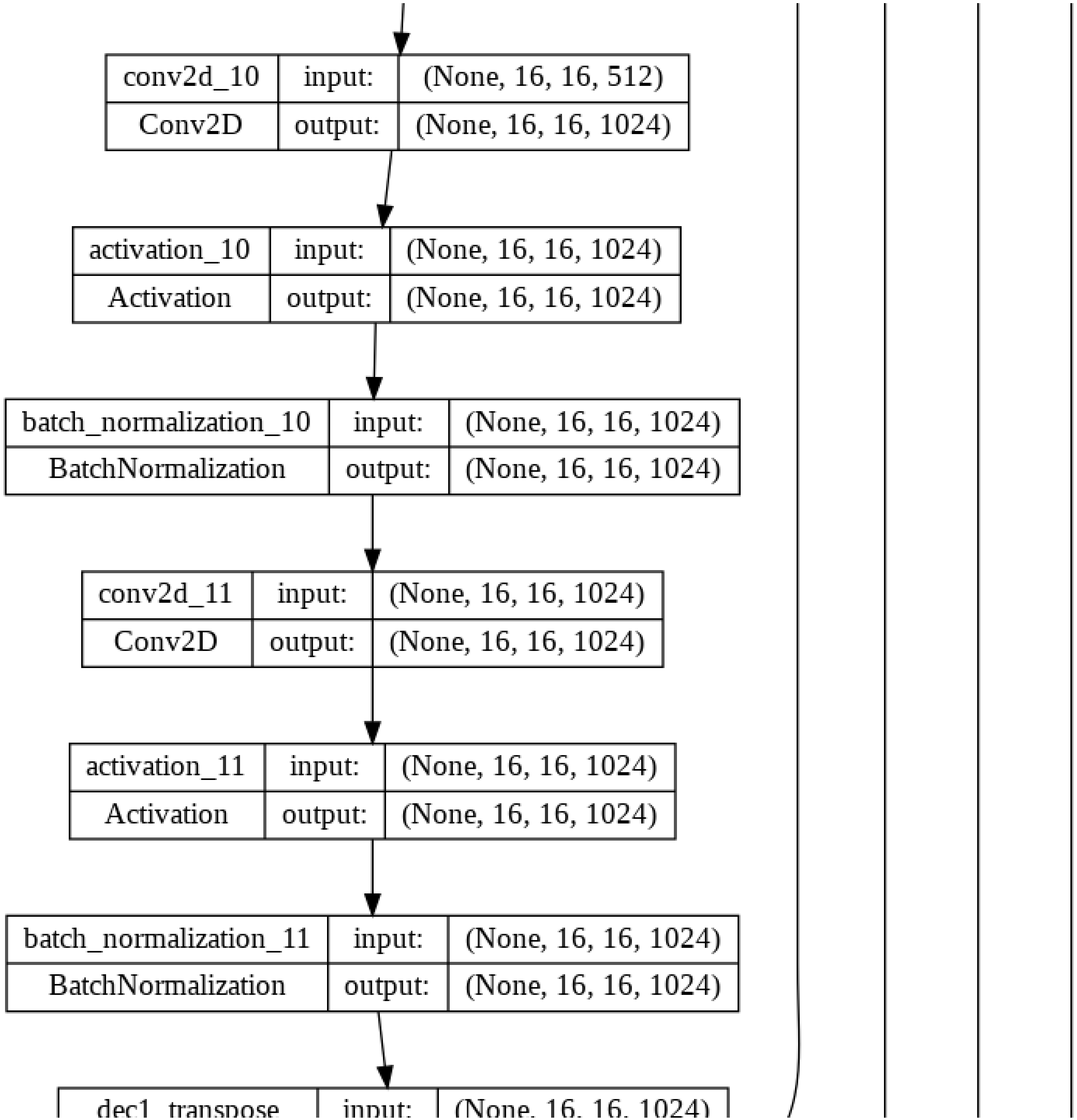

BatchNormalization
output:
(None, 256, 256, 64)
conv2d_2
input:
(None, 256, 256, 64)
Conv2D
output:
(None, 256, 256, 64)
activation_2
input:
(None, 256, 256, 64)
Activation
output:
(None, 256, 256, 64)
batch_normalization_2
input:
(None, 256, 256, 64)
BatchNormalization
output:
(None, 256, 256, 64)
conv2d_3
input:
(None, 256, 256, 64)
Conv2D
output:
(None, 256, 256, 64)
activation_3
input:
(None, 256, 256, 64)
Activation
output:
(None, 256, 256, 64)
batch_normalization_3
input:
(None, 256, 256, 64)
BatchNormalization
output:
(None, 256, 256, 64)

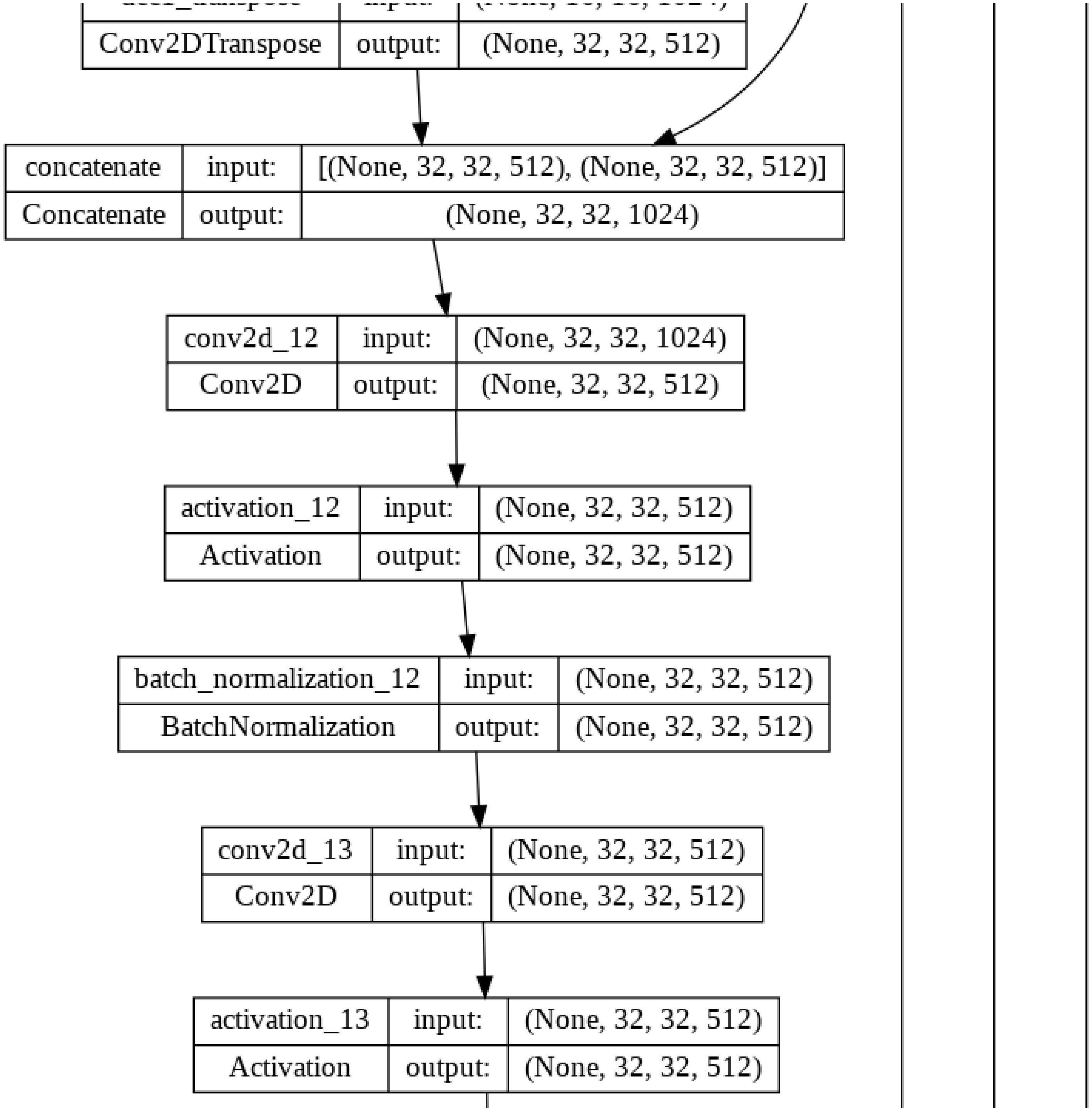

max_pooling2d
MaxPooling2D
input:
output:
(None, 256, 256, 64)
(None, 128, 128, 64)
dropout
Dropout
input:
output:
(None, 128, 128, 64)
(None, 128, 128, 64)
conv2d_4
Conv2D
input:
output:
(None, 128, 128, 64)
(None, 128, 128, 128)
activation_4
Activation
input:
output:
(None, 128, 128, 128)
(None, 128, 128, 128)
batch_normalization_4
BatchNormalization
input:
output:
(None, 128, 128, 128)
(None, 128, 128, 128)
conv2d_5
Conv2D
input:
output:
(None, 128, 128, 128)
(None, 128, 128, 128)
activation_5
input:
(None, 128, 128, 128)

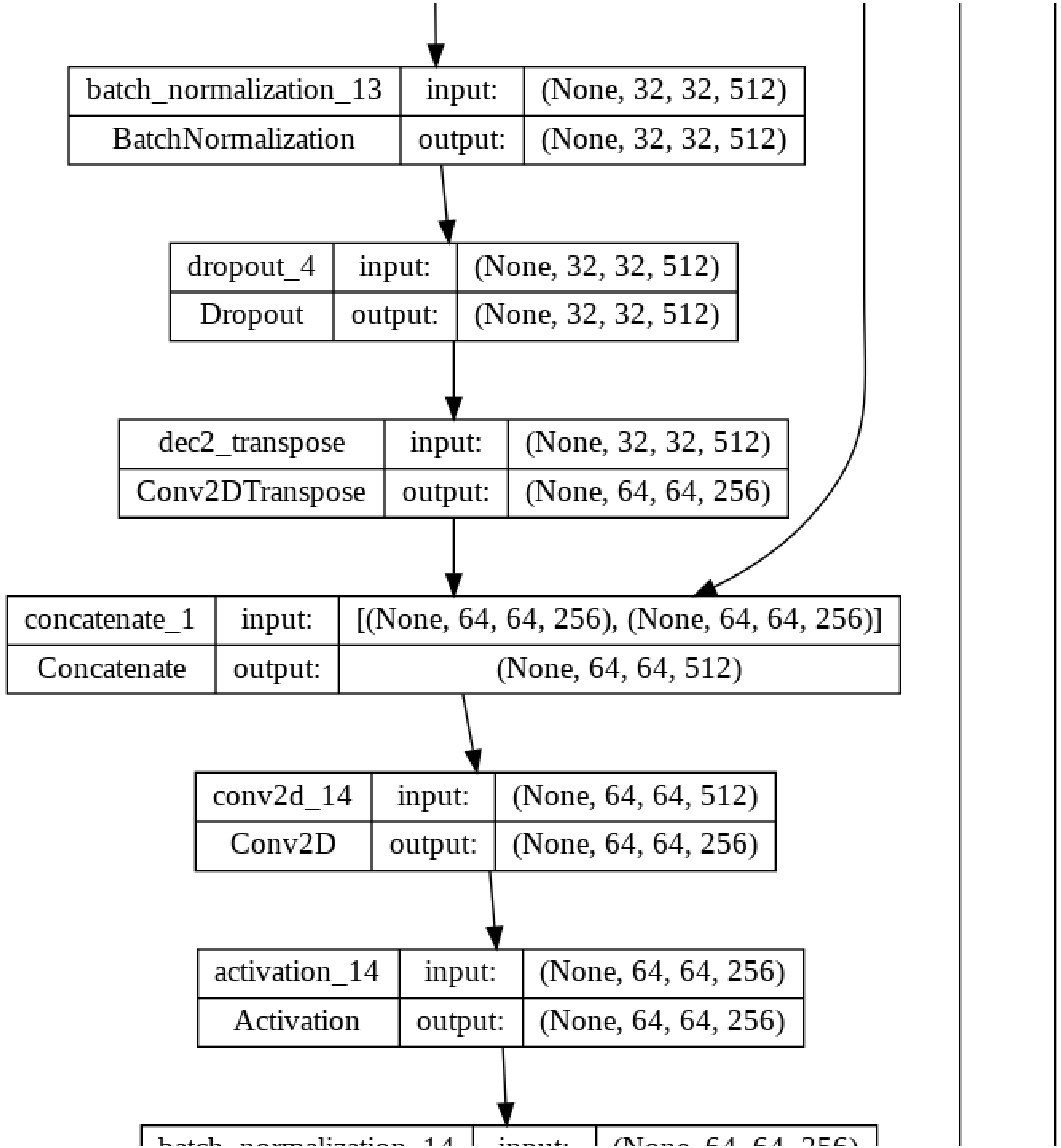

Activation | output: | (None, 128, 128, 128)
batch_normalization_5 | input: | (None, 128, 128, 128)
BatchNormalization | output: | (None, 128, 128, 128)
max_pooling2d_1 | input: | (None, 128, 128, 128)
MaxPooling2D | output: | (None, 64, 64, 128)
dropout_1 | input: | (None, 64, 64, 128)
Dropout | output: | (None, 64, 64, 128)
conv2d_6 | input: | (None, 64, 64, 128)
Conv2D | output: | (None, 64, 64, 256)
activation_6 | input: | (None, 64, 64, 256)
Activation | output: | (None, 64, 64, 256)
batch_normalization_6 | input: | (None, 64, 64, 256)
BatchNormalization | output: | (None, 64, 64, 256)

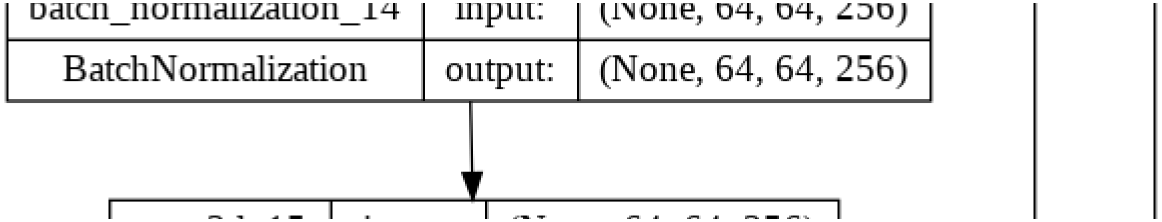

conv2d_7
Conv2D
input:
output:
(None, 64, 64, 256)
(None, 64, 64, 256)
activation_7
Activation
input:
output:
(None, 64, 64, 256)
(None, 64, 64, 256)
batch_normalization_7
BatchNormalization
input:
output:
(None, 64, 64, 256)
(None, 64, 64, 256)
max_pooling2d_2
MaxPooling2D
input:
output:
(None, 64, 64, 256)
(None, 32, 32, 256)
dropout_2
Dropout
input:
output:
(None, 32, 32, 256)
(None, 32, 32, 256)
conv2d_8
Conv2D
input:
output:
(None, 32, 32, 256)
(None, 32, 32, 512)
activation_8
input:
(None, 32, 32, 512)


Activation | output: | (None, 32, 32, 512)
batch_normalization_8 | input: | (None, 32, 32, 512)
BatchNormalization | output: | (None, 32, 32, 512)
conv2d_9 | input: | (None, 32, 32, 512)
Conv2D | output: | (None, 32, 32, 512)
activation_9 | input: | (None, 32, 32, 512)
Activation | output: | (None, 32, 32, 512)
batch_normalization_9 | input: | (None, 32, 32, 512)
BatchNormalization | output: | (None, 32, 32, 512)
max_pooling2d_3 | input: | (None, 32, 32, 512)
MaxPooling2D | output: | (None, 16, 16, 512)
dropout_3 | input: | (None, 16, 16, 512)
Dropout | output: | (None, 16, 16, 512)


conv2d_10
input:
(None, 16, 16, 512)
Conv2D
output:
(None, 16, 16, 1024)
activation_10
input:
(None, 16, 16, 1024)
Activation
output:
(None, 16, 16, 1024)
batch_normalization_10
input:
(None, 16, 16, 1024)
BatchNormalization
output:
(None, 16, 16, 1024)
conv2d_11
input:
(None, 16, 16, 1024)
Conv2D
output:
(None, 16, 16, 1024)
activation_11
input:
(None, 16, 16, 1024)
Activation
output:
(None, 16, 16, 1024)
batch_normalization_11
input:
(None, 16, 16, 1024)
BatchNormalization
output:
(None, 16, 16, 1024)
dec1_transpose
input:
(None, 16, 16, 1024)


Conv2DTranspose
output:
(None, 32, 32, 512)
concatenate
input:
[(None, 32, 32, 512), (None, 32, 32, 512)]
Concatenate
output:
(None, 32, 32, 1024)
conv2d_12
input:
(None, 32, 32, 1024)
Conv2D
output:
(None, 32, 32, 512)
activation_12
input:
(None, 32, 32, 512)
Activation
output:
(None, 32, 32, 512)
batch_normalization_12
input:
(None, 32, 32, 512)
BatchNormalization
output:
(None, 32, 32, 512)
conv2d_13
input:
(None, 32, 32, 512)
Conv2D
output:
(None, 32, 32, 512)
activation_13
input:
(None, 32, 32, 512)
Activation
output:
(None, 32, 32, 512)


batch_normalization_13
BatchNormalization
input:
(None, 32, 32, 512)
output:
(None, 32, 32, 512)
dropout_4
Dropout
input:
(None, 32, 32, 512)
output:
(None, 32, 32, 512)
dec2_transpose
Conv2DTranspose
input:
(None, 32, 32, 512)
output:
(None, 64, 64, 256)
concatenate_1
Concatenate
input:
[(None, 64, 64, 256), (None, 64, 64, 256)]
output:
(None, 64, 64, 512)
conv2d_14
Conv2D
input:
(None, 64, 64, 512)
output:
(None, 64, 64, 256)
activation_14
Activation
input:
(None, 64, 64, 256)
output:
(None, 64, 64, 256)




```
train_generator.n//8, val_image_generator.n//8
```

```
(384, 96)
```

compile and training

```
optim = tf.keras.optimizers.Adam()
loss =  DiceLoss()#sm.losses.bce_jaccard_loss

model.compile(optim, loss, metrics=[iou_score])

train_data_loader, val_data_loader = data_generator(train_generator, train_mask_generator, val_image_generator, val_mask_generator)
callback_list = create_callback_lists("baseline_unet")

history = model.fit(train_data_loader, steps_per_epoch=384, epochs=20,
                     use_multiprocessing = True,initial_epoch = 0,
                    callbacks = callback_list,validation_data = val_data_loader,
                    validation_steps = 96,)#
```

```
Epoch 1/20
384/384 [==============================] - ETA: 0s - loss: 0.7686 - iou_score: 0.1502
Epoch 1: val_iou_score improved from -inf to 0.01708, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 282s 685ms/step - loss: 0.7686 - iou_score: 0.1502 - val_loss: 0.9669 - val_iou_sc
Epoch 2/20
384/384 [==============================] - ETA: 0s - loss: 0.5024 - iou_score: 0.3698
Epoch 2: val_iou_score improved from 0.01708 to 0.30498, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 262s 681ms/step - loss: 0.5024 - iou_score: 0.3698 - val_loss: 0.5720 - val_iou_sc
Epoch 3/20
384/384 [==============================] - ETA: 0s - loss: 0.4909 - iou_score: 0.3847
```

```
Epoch 3: val_iou_score improved from 0.30498 to 0.32926, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 280s 729ms/step - loss: 0.4909 - iou_score: 0.3847 - val_loss: 0.5510 - val_iou_sc
Epoch 4/20
384/384 [==============================] - ETA: 0s - loss: 0.4724 - iou_score: 0.3976
Epoch 4: val_iou_score improved from 0.32926 to 0.37784, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 259s 674ms/step - loss: 0.4724 - iou_score: 0.3976 - val_loss: 0.5059 - val_iou_sc
Epoch 5/20
384/384 [==============================] - ETA: 0s - loss: 0.4324 - iou_score: 0.4373
Epoch 5: val_iou_score did not improve from 0.37784
384/384 [==============================] - 257s 671ms/step - loss: 0.4324 - iou_score: 0.4373 - val_loss: 0.5214 - val_iou_sc
Epoch 6/20
384/384 [==============================] - ETA: 0s - loss: 0.4303 - iou_score: 0.4424
Epoch 6: val_iou_score improved from 0.37784 to 0.40256, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 259s 674ms/step - loss: 0.4303 - iou_score: 0.4424 - val_loss: 0.4875 - val_iou_sc
Epoch 7/20
384/384 [==============================] - ETA: 0s - loss: 0.3999 - iou_score: 0.4692
Epoch 7: val_iou_score improved from 0.40256 to 0.43933, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 258s 673ms/step - loss: 0.3999 - iou_score: 0.4692 - val_loss: 0.4470 - val_iou_sc
Epoch 8/20
384/384 [==============================] - ETA: 0s - loss: 0.3942 - iou_score: 0.4744
Epoch 8: val_iou_score did not improve from 0.43933
384/384 [==============================] - 257s 670ms/step - loss: 0.3942 - iou_score: 0.4744 - val_loss: 0.6579 - val_iou_sc
Epoch 9/20
384/384 [==============================] - ETA: 0s - loss: 0.3799 - iou_score: 0.4898
Epoch 9: val_iou_score did not improve from 0.43933
384/384 [==============================] - 257s 670ms/step - loss: 0.3799 - iou_score: 0.4898 - val_loss: 0.5601 - val_iou_sc
Epoch 10/20
384/384 [==============================] - ETA: 0s - loss: 0.3805 - iou_score: 0.4897
Epoch 10: val_iou_score did not improve from 0.43933
384/384 [==============================] - 260s 678ms/step - loss: 0.3805 - iou_score: 0.4897 - val_loss: 0.5042 - val_iou_sc
Epoch 11/20
384/384 [==============================] - ETA: 0s - loss: 0.3661 - iou_score: 0.4989
Epoch 11: val_iou_score improved from 0.43933 to 0.45144, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 278s 725ms/step - loss: 0.3661 - iou_score: 0.4989 - val_loss: 0.4287 - val_iou_sc
Epoch 12/20
384/384 [==============================] - ETA: 0s - loss: 0.3609 - iou_score: 0.5075
Epoch 12: val_iou_score did not improve from 0.45144
384/384 [==============================] - 298s 776ms/step - loss: 0.3609 - iou_score: 0.5075 - val_loss: 0.5599 - val_iou_sc
Epoch 13/20
384/384 [==============================] - ETA: 0s - loss: 0.3730 - iou_score: 0.4974
Epoch 13: val_iou_score did not improve from 0.45144
```

```
384/384 [==============================] - 265s 690ms/step - loss: 0.3730 - iou_score: 0.4974 - val_loss: 0.5538 - val_iou_sc
Epoch 14/20
384/384 [==============================] - ETA: 0s - loss: 0.3556 - iou_score: 0.5151
Epoch 14: val_iou_score did not improve from 0.45144
384/384 [==============================] - 280s 729ms/step - loss: 0.3556 - iou_score: 0.5151 - val_loss: 0.6912 - val_iou_sc
Epoch 15/20
```

| dec4_transpose | input: | (None, 128, 128, 128) | 7

```
# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

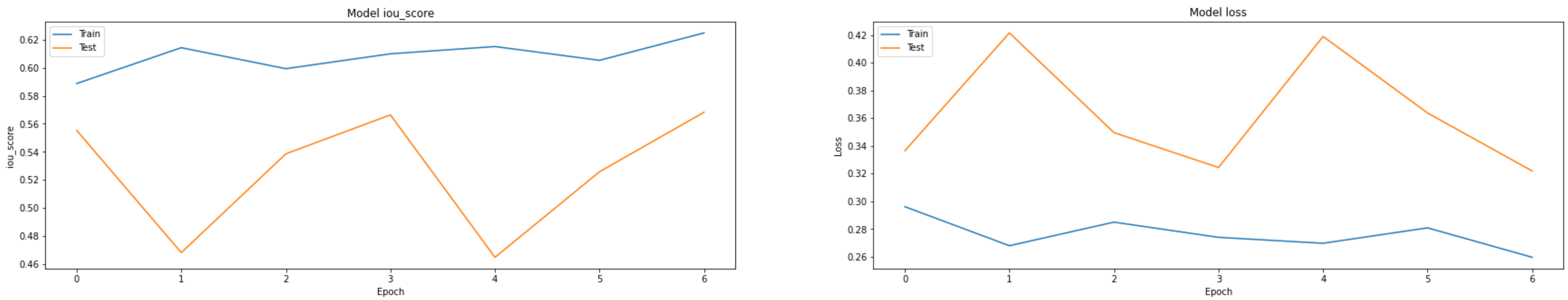
```
history = model.fit(train_data_loader, steps_per_epoch=384, epochs=40,
                     use_multiprocessing = True,initial_epoch = 25,
                    callbacks = callback_list,validation_data = val_data_loader,
                    validation_steps = 96,)#
```

```
Epoch 26/40
384/384 [==============================] - ETA: 0s - loss: 0.3123 - iou_score: 0.5681
Epoch 26: val_iou_score improved from 0.50562 to 0.52545, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 263s 684ms/step - loss: 0.3123 - iou_score: 0.5681 - val_loss: 0.3623 - val_iou_scor
Epoch 27/40
384/384 [==============================] - ETA: 0s - loss: 0.2964 - iou_score: 0.5814
Epoch 27: val_iou_score did not improve from 0.52545
384/384 [==============================] - 265s 691ms/step - loss: 0.2964 - iou_score: 0.5814 - val_loss: 0.3601 - val_iou_scor
Epoch 28/40
384/384 [==============================] - ETA: 0s - loss: 0.3019 - iou_score: 0.5789
Epoch 28: val_iou_score did not improve from 0.52545
384/384 [==============================] - 258s 672ms/step - loss: 0.3019 - iou_score: 0.5789 - val_loss: 0.4748 - val_iou_scor
Epoch 29/40
384/384 [==============================] - ETA: 0s - loss: 0.2988 - iou_score: 0.5785
Epoch 29: val_iou_score did not improve from 0.52545
384/384 [==============================] - 261s 678ms/step - loss: 0.2988 - iou_score: 0.5785 - val_loss: 0.5849 - val_iou_scor
Epoch 30/40
384/384 [==============================] - ETA: 0s - loss: 0.2907 - iou_score: 0.5895
Epoch 30: val_iou_score did not improve from 0.52545
384/384 [==============================] - 265s 689ms/step - loss: 0.2907 - iou_score: 0.5895 - val_loss: 0.4092 - val_iou_scor
Epoch 31/40
384/384 [==============================] - ETA: 0s - loss: 0.3047 - iou_score: 0.5780
Epoch 31: val_iou_score did not improve from 0.52545
384/384 [==============================] - 267s 695ms/step - loss: 0.3047 - iou_score: 0.5780 - val_loss: 0.4211 - val_iou_scor
Epoch 32/40
384/384 [==============================] - ETA: 0s - loss: 0.2745 - iou_score: 0.6072
Epoch 32: val_iou_score did not improve from 0.52545
384/384 [==============================] - 259s 674ms/step - loss: 0.2745 - iou_score: 0.6072 - val_loss: 0.3869 - val_iou_scor
Epoch 33/40
384/384 [==============================] - ETA: 0s - loss: 0.2739 - iou_score: 0.6048
Epoch 33: val_iou_score did not improve from 0.52545
384/384 [==============================] - 259s 675ms/step - loss: 0.2739 - iou_score: 0.6048 - val_loss: 0.4761 - val_iou_scor
```

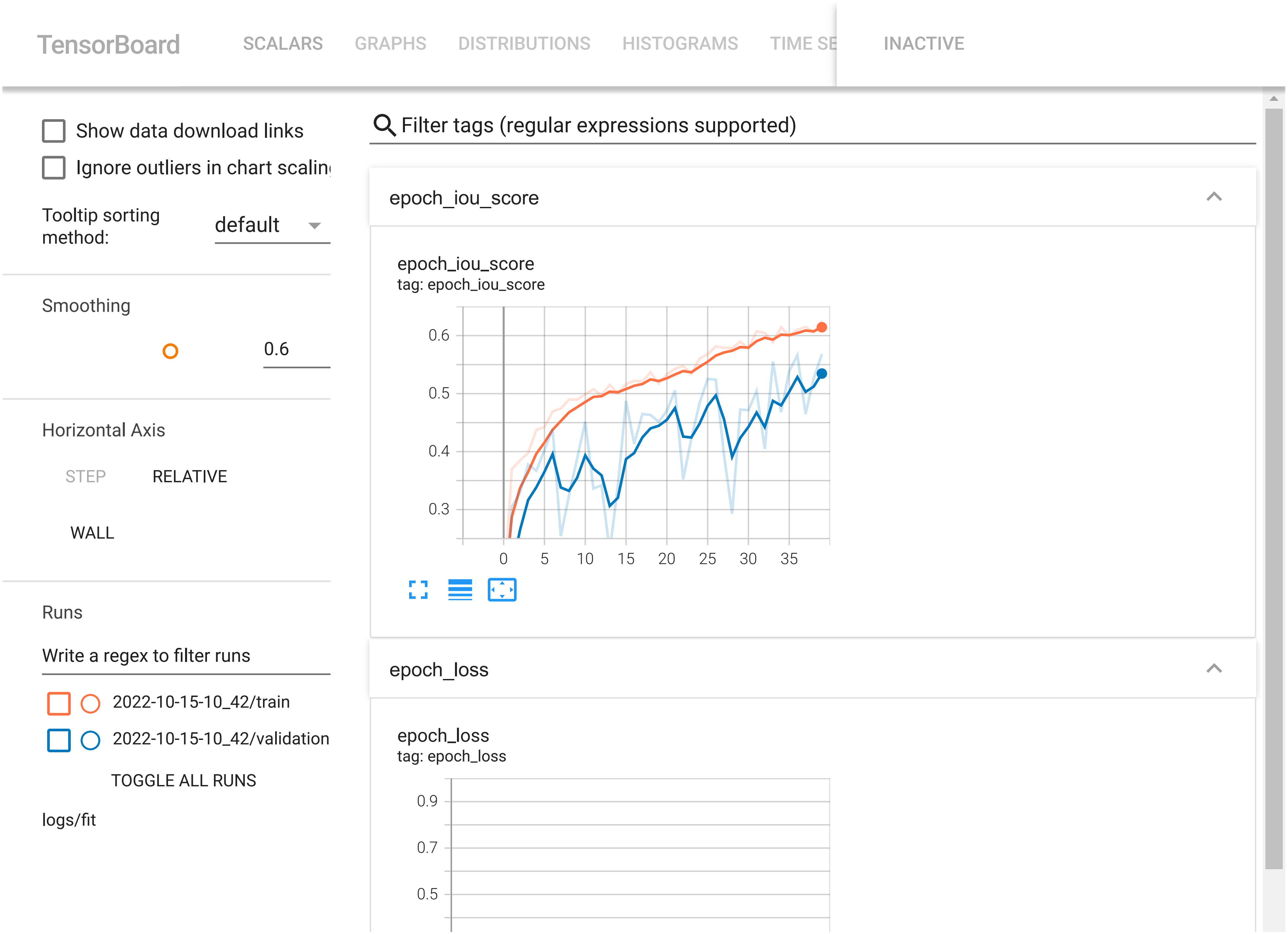
```
optim = tf.keras.optimizers.Adam(lr = .0006)
loss =  DiceLoss()#sm.losses.bce_jaccard_loss

model.compile(optim, loss, metrics=[iou_score])

history = model.fit(train_data_loader, steps_per_epoch=384, epochs=40,
                     use_multiprocessing = True,initial_epoch = 33,
                    callbacks = callback_list,validation_data = val_data_loader,
                    validation_steps = 96,)#
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizers/optimizer_v2/adam.py:110: UserWarning: The `lr` argument is deprecated,
  super(Adam, self).__init__(name, **kwargs)
Epoch 34/40
384/384 [==============================] - ETA: 0s - loss: 0.2961 - iou_score: 0.5888
Epoch 34: val_iou_score improved from 0.52545 to 0.55528, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 265s 680ms/step - loss: 0.2961 - iou_score: 0.5888 - val_loss: 0.3365 - val_iou_scor
Epoch 35/40
384/384 [==============================] - ETA: 0s - loss: 0.2680 - iou_score: 0.6144
Epoch 35: val_iou_score did not improve from 0.55528
384/384 [==============================] - 259s 673ms/step - loss: 0.2680 - iou_score: 0.6144 - val_loss: 0.4215 - val_iou_scor
Epoch 36/40
384/384 [==============================] - ETA: 0s - loss: 0.2850 - iou_score: 0.5994
Epoch 36: val_iou_score did not improve from 0.55528
384/384 [==============================] - 259s 673ms/step - loss: 0.2850 - iou_score: 0.5994 - val_loss: 0.3496 - val_iou_scor
Epoch 37/40
384/384 [==============================] - ETA: 0s - loss: 0.2740 - iou_score: 0.6100
Epoch 37: val_iou_score improved from 0.55528 to 0.56641, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 264s 687ms/step - loss: 0.2740 - iou_score: 0.6100 - val_loss: 0.3244 - val_iou_scor
Epoch 38/40
384/384 [==============================] - ETA: 0s - loss: 0.2697 - iou_score: 0.6152
Epoch 38: val_iou_score did not improve from 0.56641
384/384 [==============================] - 262s 683ms/step - loss: 0.2697 - iou_score: 0.6152 - val_loss: 0.4190 - val_iou_scor
Epoch 39/40
384/384 [==============================] - ETA: 0s - loss: 0.2808 - iou_score: 0.6054
Epoch 39: val_iou_score did not improve from 0.56641
384/384 [==============================] - 280s 731ms/step - loss: 0.2808 - iou_score: 0.6054 - val_loss: 0.3637 - val_iou_scor
Epoch 40/40
384/384 [==============================] - ETA: 0s - loss: 0.2596 - iou_score: 0.6249
Epoch 40: val_iou_score improved from 0.56641 to 0.56830, saving model to best_model_with_baseline_unet.hdf5
384/384 [==============================] - 262s 683ms/step - loss: 0.2596 - iou_score: 0.6249 - val_loss: 0.3219 - val_iou_scor
```

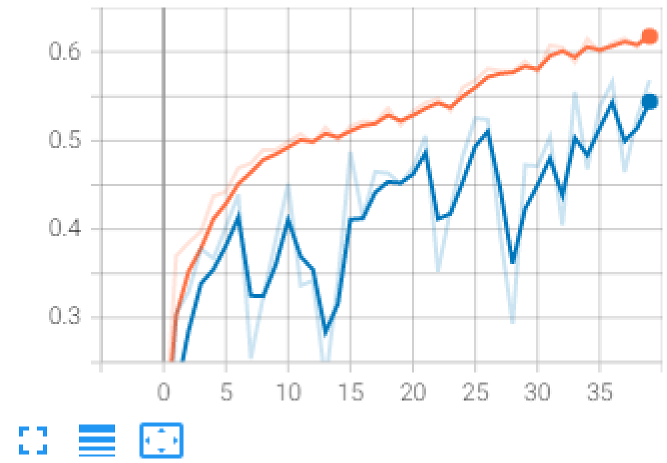
```
# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



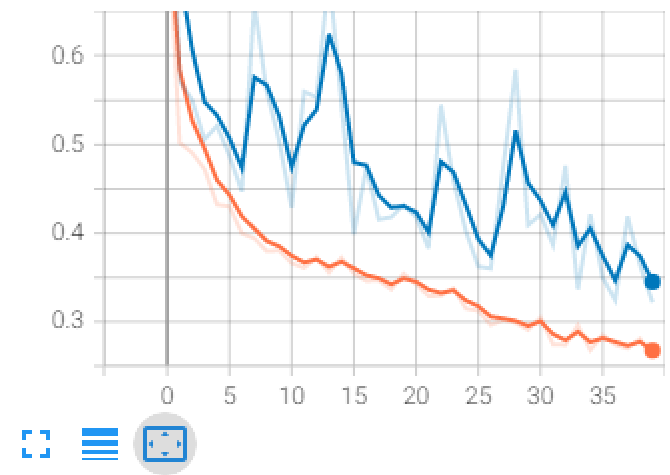

```
# loading tensorboard
%tensorboard --logdir logs/fit
```


TensorBoard
SCALARS
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
TIME SE
INACTIVE
Show data download links
Ignore outliers in chart scalin
Tooltip sorting method:
default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
2022-10-15-10_42/train
2022-10-15-10_42/validation
TOGGLE ALL RUNS
logs/fit
Filter tags (regular expressions supported)
epoch_iou_score
epoch_iou_score
tag: epoch_iou_score
0.6
0.5
0.4
0.3
0
5
10
15
20
25
30
35
epoch_loss
epoch_loss
tag: epoch_loss
0.9
0.7
0.5

epoch_iou_score
tag: epoch_iou_score



epoch_loss
tag: epoch_loss



Observation

- we have got **0.56 IOU score with val_loss = 0.32 on test data and 0.62 IOU score with 0.25 loss on train data.**
- we have **run** model for **40 epochs** and its take total **4hr time**

Attantion unet model

```
model = models.att_unet_2d((256, 256, 3), filter_num=[64, 128, 256, 512, 1024], n_labels=1,
                           stack_num_down=2, stack_num_up=2, activation='ReLU',
                           atten_activation='ReLU', attention='add', output_activation='Sigmoid',
                           batch_norm=False, pool=False, unpool=False,
                           backbone='VGG16', weights='imagenet',
                           freeze_backbone=True, freeze_batch_norm=True,
                           name='attunet')
model.compile(optim, loss, metrics=[iou_score])
```

```
optim = tf.keras.optimizers.Adam()
```

```
loss =  DiceLoss()#sm.losses.bce_jaccard_loss

model.compile(optim, loss, metrics=[iou_score])

train_data_loader, val_data_loader = data_generator(train_generator, train_mask_generator, val_image_generator, val_mask_generator)
```

```
%%time
callback_list = create_callback_lists("att_unet")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=1,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
192/192 [==============================] - ETA: 0s - loss: 0.5530 - iou_score: 0.3178
Epoch 1: val_iou_score improved from -inf to 0.05296, saving model to best_model_with_att_unet.hdf5
192/192 [==============================] - 187s 840ms/step - loss: 0.5530 - iou_score: 0.3178 - val_loss: 0.9006 - val_iou_scor
CPU times: user 2min 13s, sys: 9 s, total: 2min 22s
Wall time: 3min 7s
```

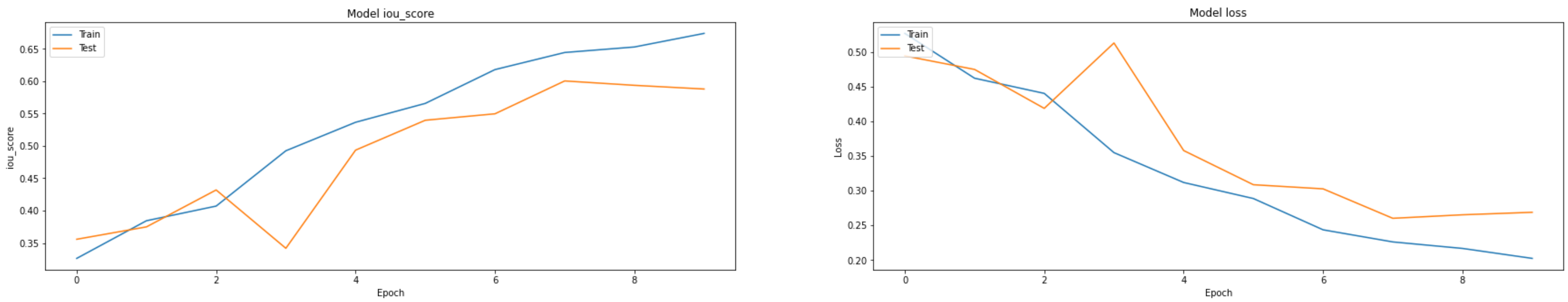
```
%%time
callback_list = create_callback_lists("att_unet_2d")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=10,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
Epoch 1/10
192/192 [==============================] - ETA: 0s - loss: 0.5269 - iou_score: 0.3262
Epoch 1: val_iou_score improved from -inf to 0.35576, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 168s 852ms/step - loss: 0.5269 - iou_score: 0.3262 - val_loss: 0.4945 - val_iou_scor
Epoch 2/10
192/192 [==============================] - ETA: 0s - loss: 0.4623 - iou_score: 0.3844
Epoch 2: val_iou_score improved from 0.35576 to 0.37490, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 161s 841ms/step - loss: 0.4623 - iou_score: 0.3844 - val_loss: 0.4749 - val_iou_scor
Epoch 3/10
```

```
192/192 [==============================] - ETA: 0s - loss: 0.4404 - iou_score: 0.4070
Epoch 3: val_iou_score improved from 0.37490 to 0.43196, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 160s 834ms/step - loss: 0.4404 - iou_score: 0.4070 - val_loss: 0.4188 - val_iou_scor
Epoch 4/10
192/192 [==============================] - ETA: 0s - loss: 0.3550 - iou_score: 0.4925
Epoch 4: val_iou_score did not improve from 0.43196
192/192 [==============================] - 157s 820ms/step - loss: 0.3550 - iou_score: 0.4925 - val_loss: 0.5131 - val_iou_scor
Epoch 5/10
192/192 [==============================] - ETA: 0s - loss: 0.3120 - iou_score: 0.5365
Epoch 5: val_iou_score improved from 0.43196 to 0.49348, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 158s 821ms/step - loss: 0.3120 - iou_score: 0.5365 - val_loss: 0.3580 - val_iou_scor
Epoch 6/10
192/192 [==============================] - ETA: 0s - loss: 0.2888 - iou_score: 0.5657
Epoch 6: val_iou_score improved from 0.49348 to 0.53965, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 820ms/step - loss: 0.2888 - iou_score: 0.5657 - val_loss: 0.3087 - val_iou_scor
Epoch 7/10
192/192 [==============================] - ETA: 0s - loss: 0.2437 - iou_score: 0.6179
Epoch 7: val_iou_score improved from 0.53965 to 0.54963, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 820ms/step - loss: 0.2437 - iou_score: 0.6179 - val_loss: 0.3028 - val_iou_scor
Epoch 8/10
192/192 [==============================] - ETA: 0s - loss: 0.2263 - iou_score: 0.6444
Epoch 8: val_iou_score improved from 0.54963 to 0.60040, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 158s 824ms/step - loss: 0.2263 - iou_score: 0.6444 - val_loss: 0.2603 - val_iou_scor
Epoch 9/10
192/192 [==============================] - ETA: 0s - loss: 0.2169 - iou_score: 0.6529
Epoch 9: val_iou_score did not improve from 0.60040
192/192 [==============================] - 157s 820ms/step - loss: 0.2169 - iou_score: 0.6529 - val_loss: 0.2654 - val_iou_scor
Epoch 10/10
192/192 [==============================] - ETA: 0s - loss: 0.2025 - iou_score: 0.6739
Epoch 10: val_iou_score did not improve from 0.60040
192/192 [==============================] - 157s 816ms/step - loss: 0.2025 - iou_score: 0.6739 - val_loss: 0.2690 - val_iou_scor
CPU times: user 18min 54s, sys: 1min 19s, total: 20min 14s
Wall time: 26min 32s
```

```
# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
```

```
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



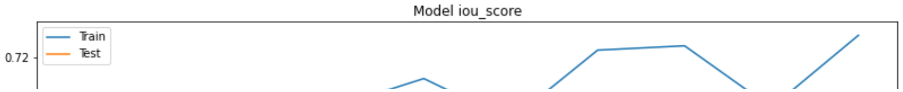

```
model.save_weights("/content/drive/MyDrive/dl_model_save/att_unet_60.hdf5")
```

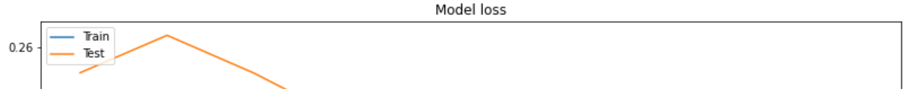
```
%%time
# callback_list = create_callback_lists("att_unet_2d")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=20,initial_epoch =10,
                    validation_data = val_data_loader,  validation_steps = 48,
```

```
              use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,


Epoch 11/20
192/192 [==============================] - ETA: 0s - loss: 0.2004 - iou_score: 0.6738
Epoch 11: val_iou_score improved from 0.60040 to 0.60863, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 159s 825ms/step - loss: 0.2004 - iou_score: 0.6738 - val_loss: 0.2505 - val_iou_scor
Epoch 12/20
192/192 [==============================] - ETA: 0s - loss: 0.1972 - iou_score: 0.6797
Epoch 12: val_iou_score did not improve from 0.60863
192/192 [==============================] - 157s 820ms/step - loss: 0.1972 - iou_score: 0.6797 - val_loss: 0.2645 - val_iou_scor
Epoch 13/20
192/192 [==============================] - ETA: 0s - loss: 0.1868 - iou_score: 0.6928
Epoch 13: val_iou_score improved from 0.60863 to 0.61234, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 158s 823ms/step - loss: 0.1868 - iou_score: 0.6928 - val_loss: 0.2503 - val_iou_scor
Epoch 14/20
192/192 [==============================] - ETA: 0s - loss: 0.1865 - iou_score: 0.6955
Epoch 14: val_iou_score improved from 0.61234 to 0.63252, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 820ms/step - loss: 0.1865 - iou_score: 0.6955 - val_loss: 0.2336 - val_iou_scor
Epoch 15/20
192/192 [==============================] - ETA: 0s - loss: 0.1755 - iou_score: 0.7093
Epoch 15: val_iou_score did not improve from 0.63252
192/192 [==============================] - 157s 818ms/step - loss: 0.1755 - iou_score: 0.7093 - val_loss: 0.2363 - val_iou_scor
Epoch 16/20
192/192 [==============================] - ETA: 0s - loss: 0.1925 - iou_score: 0.6885
Epoch 16: val_iou_score improved from 0.63252 to 0.63802, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 817ms/step - loss: 0.1925 - iou_score: 0.6885 - val_loss: 0.2275 - val_iou_scor
Epoch 17/20
192/192 [==============================] - ETA: 0s - loss: 0.1652 - iou_score: 0.7235
Epoch 17: val_iou_score improved from 0.63802 to 0.64462, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 820ms/step - loss: 0.1652 - iou_score: 0.7235 - val_loss: 0.2234 - val_iou_scor
Epoch 18/20
192/192 [==============================] - ETA: 0s - loss: 0.1659 - iou_score: 0.7257
Epoch 18: val_iou_score improved from 0.64462 to 0.64922, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 819ms/step - loss: 0.1659 - iou_score: 0.7257 - val_loss: 0.2227 - val_iou_scor
Epoch 19/20
192/192 [==============================] - ETA: 0s - loss: 0.1886 - iou_score: 0.6949
Epoch 19: val_iou_score improved from 0.64922 to 0.65389, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 817ms/step - loss: 0.1886 - iou_score: 0.6949 - val_loss: 0.2160 - val_iou_scor
Epoch 20/20
192/192 [==============================] - ETA: 0s - loss: 0.1602 - iou_score: 0.7309
```

```
Epoch 20: val_iou_score improved from 0.65389 to 0.67651, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 161s 840ms/step - loss: 0.1602 - iou_score: 0.7309 - val_loss: 0.1985 - val_iou_scor
CPU times: user 18min 44s, sys: 1min 25s, total: 20min 10s
Wall time: 26min 17s
```

```
model.load_weights("/content/best_model_with_att_unet_2d.hdf5")
model.save_weights("/content/drive/MyDrive/dl_model_save/att_unet_67.hdf5")


# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```






```
%%time
# callback_list = create_callback_lists("att_unet_2d")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=35,initial_epoch =20,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
Epoch 21/35
192/192 [==============================] - ETA: 0s - loss: 0.1563 - iou_score: 0.7352
Epoch 21: val_iou_score improved from 0.67651 to 0.67906, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 817ms/step - loss: 0.1563 - iou_score: 0.7352 - val_loss: 0.1967 - val_iou_scor
Epoch 22/35
192/192 [==============================] - ETA: 0s - loss: 0.1495 - iou_score: 0.7446
Epoch 22: val_iou_score did not improve from 0.67906
192/192 [==============================] - 156s 812ms/step - loss: 0.1495 - iou_score: 0.7446 - val_loss: 0.2016 - val_iou_scor
Epoch 23/35
192/192 [==============================] - ETA: 0s - loss: 0.1512 - iou_score: 0.7427
Epoch 23: val_iou_score improved from 0.67906 to 0.68845, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 156s 815ms/step - loss: 0.1512 - iou_score: 0.7427 - val_loss: 0.1893 - val_iou_scor
Epoch 24/35
192/192 [==============================] - ETA: 0s - loss: 0.1805 - iou_score: 0.7052
Epoch 24: val_iou_score did not improve from 0.68845
192/192 [==============================] - 156s 813ms/step - loss: 0.1805 - iou_score: 0.7052 - val_loss: 0.2044 - val_iou_scor
Epoch 25/35
192/192 [==============================] - ETA: 0s - loss: 0.1566 - iou_score: 0.7346
Epoch 25: val_iou_score did not improve from 0.68845
192/192 [==============================] - 156s 812ms/step - loss: 0.1566 - iou_score: 0.7346 - val_loss: 0.2011 - val_iou_scor
Epoch 26/35
192/192 [==============================] - ETA: 0s - loss: 0.2026 - iou_score: 0.6790
Epoch 26: val_iou_score did not improve from 0.68845
192/192 [==============================] - 156s 813ms/step - loss: 0.2026 - iou_score: 0.6790 - val_loss: 0.2120 - val_iou_scor
Epoch 27/35
192/192 [==============================] - ETA: 0s - loss: 0.1542 - iou_score: 0.7385
Epoch 27: val_iou_score did not improve from 0.68845
192/192 [==============================] - 156s 812ms/step - loss: 0.1542 - iou_score: 0.7385 - val_loss: 0.1910 - val_iou_scor
Epoch 28/35
```

```
192/192 [==============================] - ETA: 0s - loss: 0.1528 - iou_score: 0.7427
Epoch 28: val_iou_score did not improve from 0.68845
192/192 [==============================] - 155s 810ms/step - loss: 0.1528 - iou_score: 0.7427 - val_loss: 0.1921 - val_iou_scor
Epoch 29/35
192/192 [==============================] - ETA: 0s - loss: 0.1606 - iou_score: 0.7304
Epoch 29: val_iou_score did not improve from 0.68845
192/192 [==============================] - 156s 811ms/step - loss: 0.1606 - iou_score: 0.7304 - val_loss: 0.2077 - val_iou_scor
CPU times: user 16min 46s, sys: 1min 21s, total: 18min 8s
Wall time: 23min 24s
```

```
# model.save_weights("/content/drive/MyDrive/dl_model_save/att_unet_67.hdf5")
```

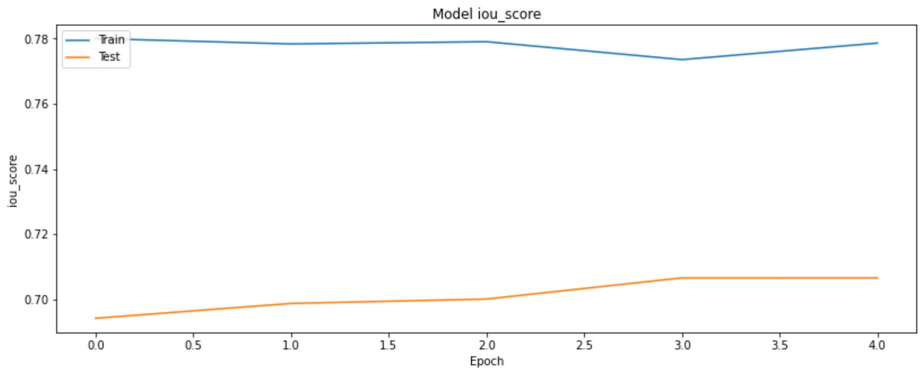
```
optim = tf.keras.optimizers.Adam(learning_rate = .0001)
loss =  DiceLoss()#sm.losses.bce_jaccard_loss
callback_list = create_callback_lists("att_unet_2d")

model.compile(optim, loss, metrics=[iou_score])
model.load_weights("/content/best_model_with_att_unet_2d.hdf5")
```

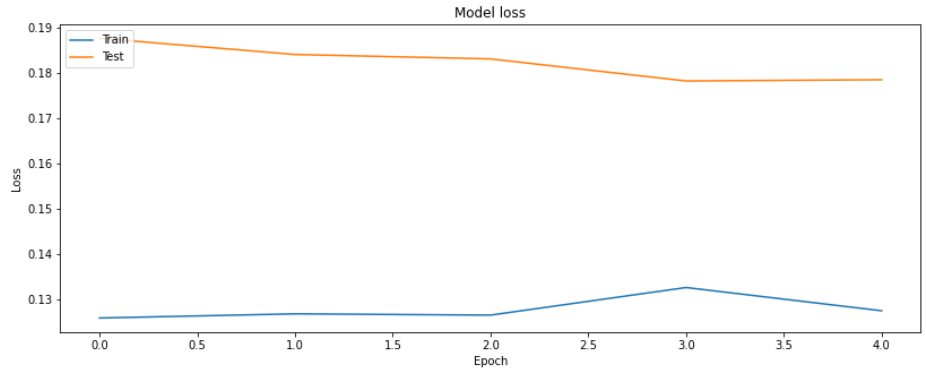
```
%%time
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=40,initial_epoch =35,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
Epoch 36/40
192/192 [==============================] - ETA: 0s - loss: 0.1259 - iou_score: 0.7800
Epoch 36: val_iou_score improved from -inf to 0.69416, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 160s 822ms/step - loss: 0.1259 - iou_score: 0.7800 - val_loss: 0.1876 - val_iou_scor
Epoch 37/40
192/192 [==============================] - ETA: 0s - loss: 0.1268 - iou_score: 0.7784
Epoch 37: val_iou_score improved from 0.69416 to 0.69870, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 157s 820ms/step - loss: 0.1268 - iou_score: 0.7784 - val_loss: 0.1840 - val_iou_scor
Epoch 38/40
192/192 [==============================] - ETA: 0s - loss: 0.1265 - iou_score: 0.7791
```

```
Epoch 38: val_iou_score improved from 0.69870 to 0.70005, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 158s 825ms/step - loss: 0.1265 - iou_score: 0.7791 - val_loss: 0.1830 - val_iou_scor
Epoch 39/40
192/192 [==============================] - ETA: 0s - loss: 0.1326 - iou_score: 0.7736
Epoch 39: val_iou_score improved from 0.70005 to 0.70655, saving model to best_model_with_att_unet_2d.hdf5
192/192 [==============================] - 160s 833ms/step - loss: 0.1326 - iou_score: 0.7736 - val_loss: 0.1781 - val_iou_scor
Epoch 40/40
192/192 [==============================] - ETA: 0s - loss: 0.1275 - iou_score: 0.7786
Epoch 40: val_iou_score did not improve from 0.70655
192/192 [==============================] - 158s 820ms/step - loss: 0.1275 - iou_score: 0.7786 - val_loss: 0.1784 - val_iou_scor
CPU times: user 9min 28s, sys: 1min 8s, total: 10min 37s
Wall time: 13min 14s
```

```
# model.save_weights("/content/drive/MyDrive/dl_model_save/att_unet_70.hdf5")


# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```






```
# optim = tf.keras.optimizers.Adam(learning_rate = .0001)
# loss =  DiceLoss()#sm.losses.bce_jaccard_loss
# callback_list = create_callback_lists("att_unet_2d")

# model.compile(optim, loss, metrics=[iou_score])
# model.load_weights("/content/best_model_with_att_unet_2d.hdf5")
```

```
# %%time
# history = model.fit(train_data_loader, steps_per_epoch=192, epochs=45,initial_epoch =40,
#                       validation_data = val_data_loader,  validation_steps = 48,
#                      use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
# loading tensorboard
# %tensorboard --logdir logs/fit
```

```
%%time
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=40,initial_epoch =35,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

Vnet model

refer : https://github.com/yingkaisha/keras-unet-collection/blob/main/keras_unet_collection/_model_vnet_2d.py

```
model = models.vnet_2d((256, 256, 3), filter_num=[64, 128, 256, 512], n_labels=1,
                            activation='ReLU',  res_num_ini=1, res_num_max=3,
                             output_activation='Sigmoid',
            batch_norm=False, pool=False, unpool=False, name='vnet')
```

```
optim = tf.keras.optimizers.Adam(learning_rate = .0001)
loss =  sm.losses.bce_jaccard_loss #DiceLoss()#sm.losses.bce_jaccard_loss
```

```
model.compile(optim, loss, metrics=[iou_score])

# train_data_loader, val_data_loader = data_generator(train_generator, train_mask_generator, val_image_generator, val_mask_generator)
```

```
%%time
callback_list = create_callback_lists("vnet")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=1,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
192/192 [==============================] - ETA: 0s - loss: 0.9914 - iou_score: 0.0046
Epoch 1: val_iou_score improved from -inf to 0.00000, saving model to best_model_with_vnet.hdf5
192/192 [==============================] - 166s 849ms/step - loss: 0.9914 - iou_score: 0.0046 - val_loss: 1.0000 - val_iou_scor
CPU times: user 2min 3s, sys: 8.92 s, total: 2min 12s
Wall time: 2min 46s
```

```
%%time
callback_list = create_callback_lists("vnet")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=1,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
192/192 [==============================] - ETA: 0s - loss: 0.9476 - iou_score: 0.1478
Epoch 1: val_iou_score improved from -inf to 0.27393, saving model to best_model_with_vnet.hdf5
192/192 [==============================] - 175s 899ms/step - loss: 0.9476 - iou_score: 0.1478 - val_loss: 0.8066 - val_iou_scor
CPU times: user 2min 16s, sys: 19.8 s, total: 2min 36s
Wall time: 2min 55s
```

```
%%time
callback_list = create_callback_lists("vnet")
```

```
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=1,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
192/192 [==============================] - ETA: 0s - loss: 0.7145 - iou_score: 0.3412
Epoch 1: val_iou_score improved from -inf to 0.30380, saving model to best_model_with_vnet.hdf5
192/192 [==============================] - 174s 906ms/step - loss: 0.7145 - iou_score: 0.3412 - val_loss: 0.7664 - val_iou_scor
CPU times: user 2min 15s, sys: 29.5 s, total: 2min 44s
Wall time: 3min 22s
```

```
%%time
callback_list = create_callback_lists("vnet")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=3,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
%%time
callback_list = create_callback_lists("vnet")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=2,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
Epoch 1/2
192/192 [==============================] - ETA: 0s - loss: 1.0000 - iou_score: 1.2830e-09
Epoch 1: val_iou_score improved from -inf to 0.02083, saving model to best_model_with_vnet.hdf5
192/192 [==============================] - 169s 870ms/step - loss: 1.0000 - iou_score: 1.2830e-09 - val_loss: 0.9792 - val_iou_
Epoch 2/2
192/192 [==============================] - ETA: 0s - loss: 1.0000 - iou_score: 1.4416e-09
Epoch 2: val_iou_score did not improve from 0.02083
192/192 [==============================] - 168s 873ms/step - loss: 1.0000 - iou_score: 1.4416e-09 - val_loss: 0.9792 - val_iou_
CPU times: user 4min 11s, sys: 32.2 s, total: 4min 44s
Wall time: 5min 36s
```

```
%%time
callback_list = create_callback_lists("vnet")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=5,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
Epoch 1/5
192/192 [==============================] - ETA: 0s - loss: 0.9976 - iou_score: 0.0012
Epoch 1: val_iou_score improved from -inf to 0.02083, saving model to best_model_with_vnet.hdf5
192/192 [==============================] - 178s 835ms/step - loss: 0.9976 - iou_score: 0.0012 - val_loss: 0.9792 - val_iou_scor
Epoch 2/5
192/192 [==============================] - ETA: 0s - loss: 1.0000 - iou_score: 1.4334e-09
Epoch 2: val_iou_score did not improve from 0.02083
192/192 [==============================] - 165s 861ms/step - loss: 1.0000 - iou_score: 1.4334e-09 - val_loss: 0.9792 - val_iou_
Epoch 3/5
192/192 [==============================] - ETA: 0s - loss: 1.0000 - iou_score: 1.2999e-09
Epoch 3: val_iou_score did not improve from 0.02083
192/192 [==============================] - 167s 869ms/step - loss: 1.0000 - iou_score: 1.2999e-09 - val_loss: 0.9792 - val_iou_
Epoch 4/5
192/192 [==============================] - ETA: 0s - loss: 0.9948 - iou_score: 0.0052
Epoch 4: val_iou_score did not improve from 0.02083
192/192 [==============================] - 168s 874ms/step - loss: 0.9948 - iou_score: 0.0052 - val_loss: 0.9792 - val_iou_scor
Epoch 5/5
192/192 [==============================] - ETA: 0s - loss: 1.0000 - iou_score: 1.1715e-09
Epoch 5: val_iou_score did not improve from 0.02083
192/192 [==============================] - 168s 875ms/step - loss: 1.0000 - iou_score: 1.1715e-09 - val_loss: 0.9792 - val_iou_
CPU times: user 10min 29s, sys: 1min 3s, total: 11min 33s
Wall time: 14min 6s
```

senet

```
# loading unet model with backbone - senet

model = sm.Unet('senet154', encoder_weights="imagenet", classes=1,
```

```
                activation='sigmoid',encoder_freeze=True, input_shape=(256, 256,3))
```

```
callback_list = create_callback_lists(name = "senet154_unet")
optim = tf.keras.optimizers.Adam()
loss =  DiceLoss()#sm.losses.bce_jaccard_loss

model.compile(optim, loss, metrics=[iou_score])
```

```
%%time
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=1,
                    validation_data = val_data_loader,  validation_steps = 48,
                   use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
192/192 [==============================] - ETA: 0s - loss: 0.4664 - iou_score: 0.4150
Epoch 1: val_iou_score improved from -inf to 0.07701, saving model to best_model_with_.hdf5
192/192 [==============================] - 417s 2s/step - loss: 0.4664 - iou_score: 0.4150 - val_loss: 0.8585 - val_iou_score:
CPU times: user 6min 24s, sys: 1min 49s, total: 8min 13s
Wall time: 7min
```

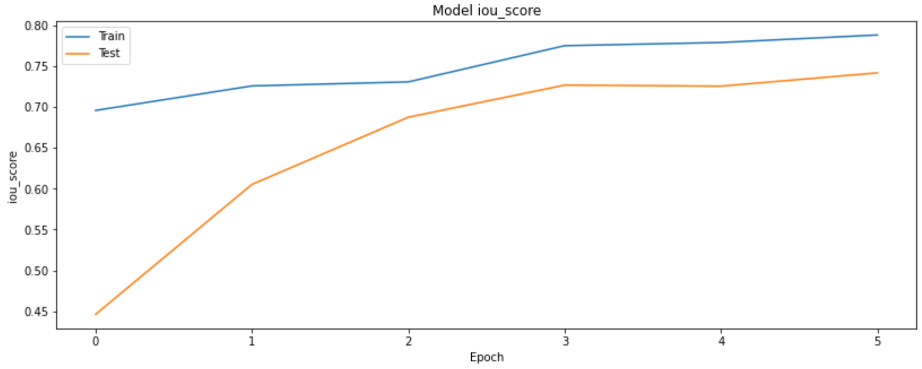
```
%%time
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=6,
                    validation_data = val_data_loader,  validation_steps = 48,
                    initial_epoch = 0,use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

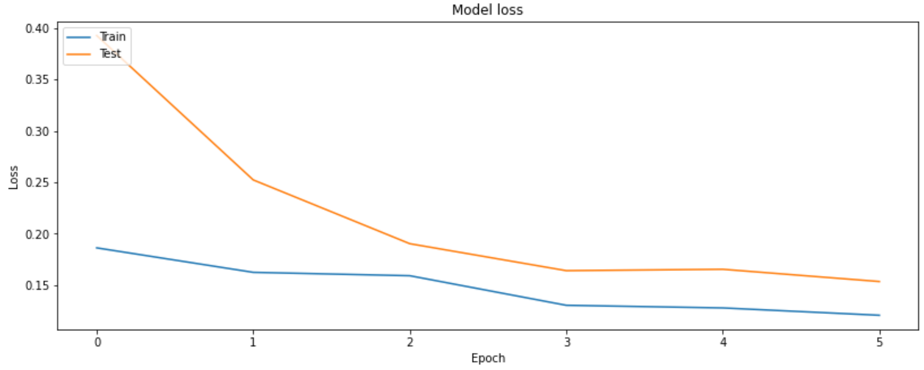
```
Epoch 1/6
192/192 [==============================] - ETA: 0s - loss: 0.1862 - iou_score: 0.6960
Epoch 1: val_iou_score improved from 0.07701 to 0.44615, saving model to best_model_with_.hdf5
192/192 [==============================] - 389s 2s/step - loss: 0.1862 - iou_score: 0.6960 - val_loss: 0.3927 - val_iou_score:
Epoch 2/6
192/192 [==============================] - ETA: 0s - loss: 0.1624 - iou_score: 0.7260
Epoch 2: val_iou_score improved from 0.44615 to 0.60533, saving model to best_model_with_.hdf5
192/192 [==============================] - 389s 2s/step - loss: 0.1624 - iou_score: 0.7260 - val_loss: 0.2523 - val_iou_score:
Epoch 3/6
```

```
192/192 [==============================] - ETA: 0s - loss: 0.1591 - iou_score: 0.7309
Epoch 3: val_iou_score improved from 0.60533 to 0.68767, saving model to best_model_with_.hdf5
192/192 [==============================] - 384s 2s/step - loss: 0.1591 - iou_score: 0.7309 - val_loss: 0.1902 - val_iou_score:
Epoch 4/6
192/192 [==============================] - ETA: 0s - loss: 0.1303 - iou_score: 0.7753
Epoch 4: val_iou_score improved from 0.68767 to 0.72700, saving model to best_model_with_.hdf5
192/192 [==============================] - 383s 2s/step - loss: 0.1303 - iou_score: 0.7753 - val_loss: 0.1640 - val_iou_score:
Epoch 5/6
192/192 [==============================] - ETA: 0s - loss: 0.1278 - iou_score: 0.7792
Epoch 5: val_iou_score did not improve from 0.72700
192/192 [==============================] - 375s 2s/step - loss: 0.1278 - iou_score: 0.7792 - val_loss: 0.1653 - val_iou_score:
Epoch 6/6
192/192 [==============================] - ETA: 0s - loss: 0.1206 - iou_score: 0.7884
Epoch 6: val_iou_score improved from 0.72700 to 0.74200, saving model to best_model_with_.hdf5
192/192 [==============================] - 385s 2s/step - loss: 0.1206 - iou_score: 0.7884 - val_loss: 0.1534 - val_iou_score:
CPU times: user 34min 29s, sys: 14min 1s, total: 48min 31s
Wall time: 38min 29s
```

```
# model.load_weights("/content/best_model_with_.hdf5")
# model.save_weights("/content/drive/MyDrive/DL_models_hdf/unet_senet_model.hdf5")
# model.load_weights("/content/drive/MyDrive/imp document/unet_res_model.hdf5")


# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
```

```
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```





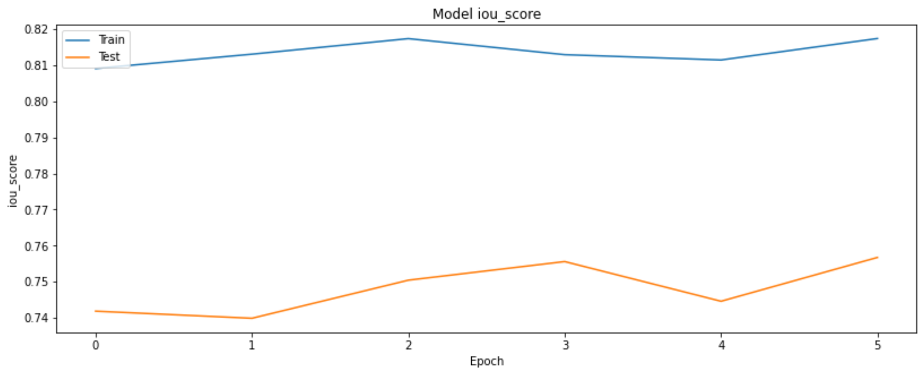

```
callback_list = create_callback_lists(name = "senet154_unet")
```

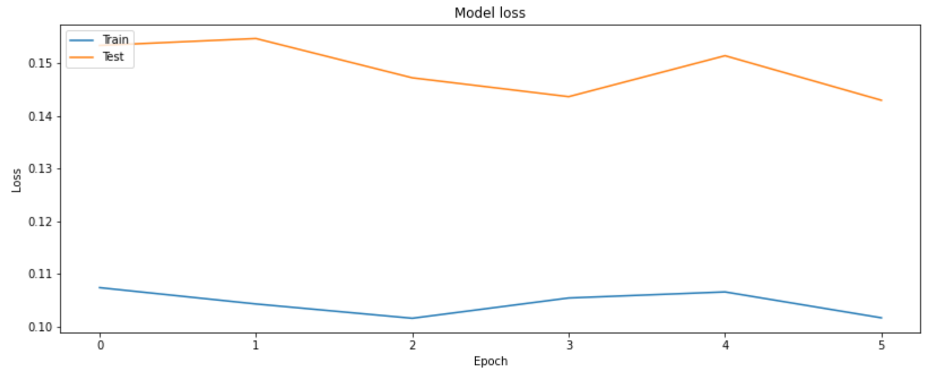
```
%%time
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=12,
                    validation_data = val_data_loader,  validation_steps = 48,
                    initial_epoch = 6,use_multiprocessing = True, callbacks = callback_list )#callbacks = callback_list,
```

```
Epoch 7/12
192/192 [==============================] - ETA: 0s - loss: 0.1074 - iou_score: 0.8091
Epoch 7: val_iou_score improved from -inf to 0.74189, saving model to best_model_with_senet154_unet.hdf5
192/192 [==============================] - 407s 2s/step - loss: 0.1074 - iou_score: 0.8091 - val_loss: 0.1533 - val_iou_score:
Epoch 8/12
192/192 [==============================] - ETA: 0s - loss: 0.1043 - iou_score: 0.8130
Epoch 8: val_iou_score did not improve from 0.74189
192/192 [==============================] - 363s 2s/step - loss: 0.1043 - iou_score: 0.8130 - val_loss: 0.1546 - val_iou_score:
Epoch 9/12
192/192 [==============================] - ETA: 0s - loss: 0.1016 - iou_score: 0.8173
```

```
Epoch 9: val_iou_score improved from 0.74189 to 0.75046, saving model to best_model_with_senet154_unet.hdf5
192/192 [==============================] - 368s 2s/step - loss: 0.1016 - iou_score: 0.8173 - val_loss: 0.1472 - val_iou_score:
Epoch 10/12
192/192 [==============================] - ETA: 0s - loss: 0.1054 - iou_score: 0.8129
Epoch 10: val_iou_score improved from 0.75046 to 0.75562, saving model to best_model_with_senet154_unet.hdf5
192/192 [==============================] - 362s 2s/step - loss: 0.1054 - iou_score: 0.8129 - val_loss: 0.1436 - val_iou_score:
Epoch 11/12
192/192 [==============================] - ETA: 0s - loss: 0.1066 - iou_score: 0.8114
Epoch 11: val_iou_score did not improve from 0.75562
192/192 [==============================] - 346s 2s/step - loss: 0.1066 - iou_score: 0.8114 - val_loss: 0.1514 - val_iou_score:
Epoch 12/12
192/192 [==============================] - ETA: 0s - loss: 0.1017 - iou_score: 0.8174
Epoch 12: val_iou_score improved from 0.75562 to 0.75676, saving model to best_model_with_senet154_unet.hdf5
192/192 [==============================] - 357s 2s/step - loss: 0.1017 - iou_score: 0.8174 - val_loss: 0.1429 - val_iou_score:
CPU times: user 29min 31s, sys: 16min 42s, total: 46min 13s
Wall time: 37min 22s
```

```python
# Plot training & validation iou_score values
plt.figure(figsize=(30, 5))
plt.subplot(121)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(122)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```






```
# callback_list = create_callback_lists(name = "senet154_unet")
optim = tf.keras.optimizers.Adam(learning_rate = .006)
loss =  DiceLoss()#sm.losses.bce_jaccard_loss

model.compile(optim, loss, metrics=[iou_score])
```

```
%%time
model.load_weights("/content/drive/MyDrive/DL_models_hdf/unet_senet_model_76_IOU.hdf5")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=18,
                    validation_data = val_data_loader,  validation_steps = 48,
                    initial_epoch = 16,use_multiprocessing = True)#callbacks = callback_list,
```

```
Epoch 17/18
192/192 [==============================] - 374s 2s/step - loss: 0.0860 - iou_score: 0.8429 - val_loss: 0.1356 - val_iou_score:
Epoch 18/18
192/192 [==============================] - 345s 2s/step - loss: 0.0860 - iou_score: 0.8428 - val_loss: 0.1327 - val_iou_score:
CPU times: user 10min 20s, sys: 3min 49s, total: 14min 9s
Wall time: 15min 26s
```

```
%%time
model.load_weights("/content/drive/MyDrive/DL_models_hdf/unet_senet_model_77_IOU.hdf5")
history = model.fit(train_data_loader, steps_per_epoch=192, epochs=20,
                    validation_data = val_data_loader,  validation_steps = 48,
                    initial_epoch = 18,use_multiprocessing = True)#callbacks = callback_list,
```

```
Epoch 19/20
192/192 [==============================] - 344s 2s/step - loss: 0.0803 - iou_score: 0.8523 - val_loss: 0.1321 - val_iou_score:
Epoch 20/20
192/192 [==============================] - 347s 2s/step - loss: 0.0801 - iou_score: 0.8526 - val_loss: 0.1311 - val_iou_score:
CPU times: user 9min 44s, sys: 3min 42s, total: 13min 27s
Wall time: 11min 31s
```

```
model.save_weights("/content/drive/MyDrive/DL_models_hdf/unet_senet_model_77_IOU.hdf5")
```

```
# callback_list = create_callback_lists(name = "senet154_unet")
# optim = tf.keras.optimizers.Adam(learning_rate = 0.006)
# loss =  DiceLoss()#sm.losses.bce_jaccard_loss

# model.compile(optim, loss, metrics=[iou_score])
```

```
# %%time
# model.load_weights("/content/drive/MyDrive/DL_models_hdf/unet_senet_model_77_IOU.hdf5")
# history = model.fit(train_data_loader, steps_per_epoch=192, epochs=22,
#                     validation_data = val_data_loader,  validation_steps = 48,
#                     initial_epoch = 20,use_multiprocessing = True)#callbacks = callback_list,
```

Inference

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
def load_and_evaluate(weights_path, data_loader, ):
```

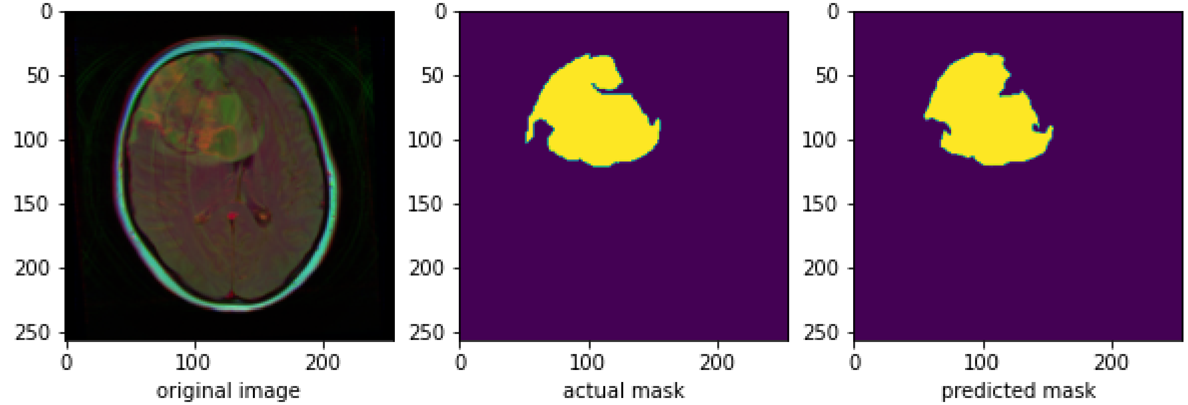
```
  '''
        here we take path of model.hdf5 file
        then load and compile it and then
        evalute on test image
  '''
  new_model = sm.Unet('efficientnetb4', encoder_weights="imagenet", classes=1,
                activation='sigmoid',encoder_freeze=True, input_shape=(256, 256,3))

  new_model.load_weights("{}".format(weights_path) )
  optim = tf.keras.optimizers.Adam()
  loss =  DiceLoss()#sm.losses.bce_jaccard_loss

  new_model.compile(optim, loss, metrics=[iou_score])
  result = new_model.evaluate(data_loader, steps=92)
  print("test loss, test IOU score:", result)
  return new_model

model_hdf_file = "/content/drive/MyDrive/dl_model_save/efficientnetb4_unet_81.h5"

new_model = load_and_evaluate(weights_path= model_hdf_file, data_loader = val_data_loader )
```

```
Downloading data from https://github.com/Callidior/keras-applications/releases/download/efficientnet/efficientnet-b4_weights_tf
71892840/71892840 [==============================] - 9s 0us/step
92/92 [==============================] - 27s 134ms/step - loss: 0.0886 - iou_score: 0.8489
test loss, test IOU score: [0.08858896046876907, 0.8489263653755188]
```

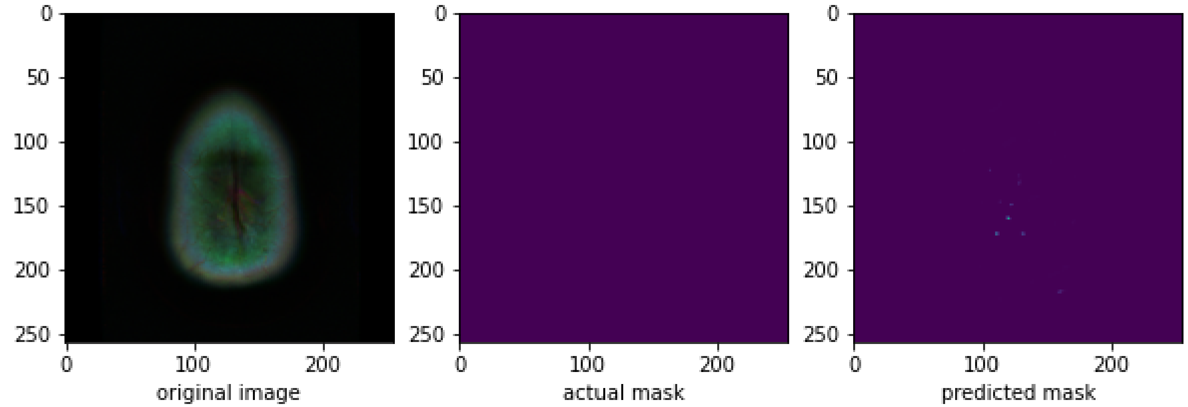
```
#predicted segmentation map
# https://kiansoon.medium.com/semantic-segmentation-is-the-task-of-partitioning-an-image-into-multiple-segments-based-on-the-356a5582


def predict_mask_image(X_test, new_model, num = 10, ):
  # plotting the image

  for i in range(num):
    # original image
    idx = int(np.random.randint(0, X_test.shape[0],1 ))
```

```
    # idx = 2172
    # print(idx)
    image = cv2.imread(X_test['image'].iloc[idx], cv2.IMREAD_UNCHANGED)
    image = image/255.
    image = cv2.resize(image, (256,256))


    predicted  = new_model.predict(image[np.newaxis,:,:,:])# np.newaxis increases dimention

    #original segmentation map
    image_mask = cv2.imread(X_test['mask'].iloc[idx], cv2.IMREAD_UNCHANGED)
    image_mask = cv2.resize(image_mask, (256,256))
    result = cv2.normalize(image, dst=None, alpha=0, beta=255,
                           norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8U)
    cv2.imwrite("mask_img2.jpg", result)
    plt.figure(figsize=(10,6))
    plt.subplot(131)
    plt.imshow(image)
    plt.xlabel("original image")
    plt.subplot(132)
    plt.imshow(image_mask)
    plt.xlabel("actual mask")
    plt.subplot(133)
    plt.imshow(predicted[0,:,:,0])
    plt.xlabel("predicted mask")
    plt.show()
  return predicted
predicted = predict_mask_image(X_train, new_model, num = 5)
```

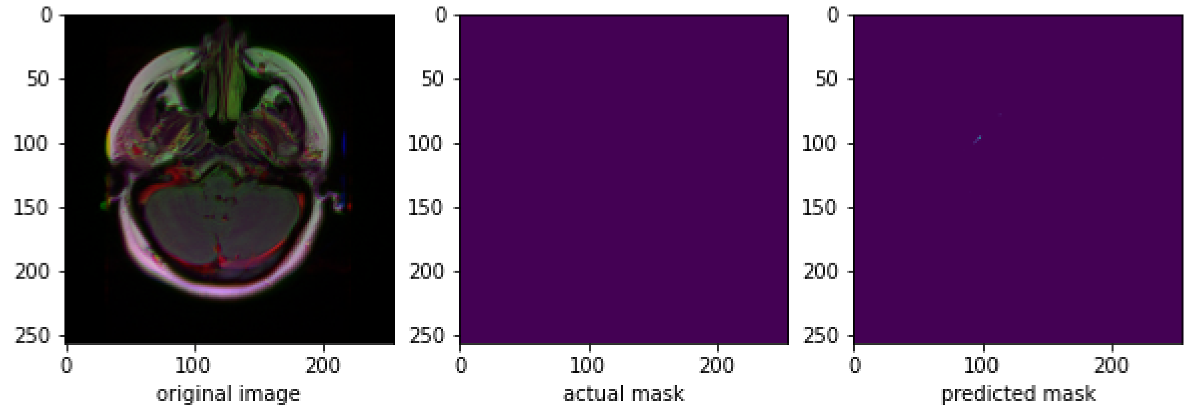
1/1 [==============================] - 0s 33ms/step



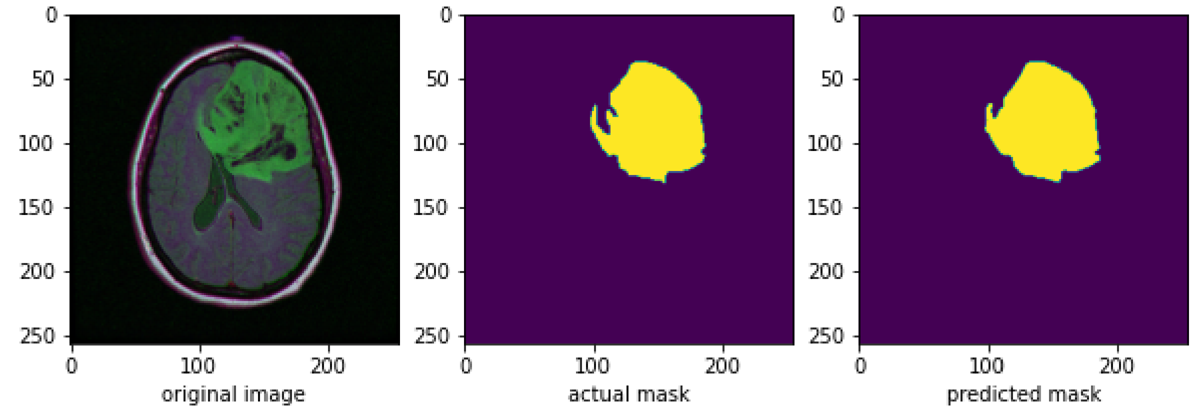
1/1 [==============================] - 0s 32ms/step



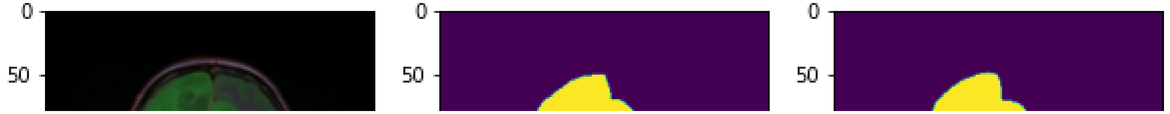
1/1 [==============================] - 0s 32ms/step



1/1 [==============================] - 0s 31ms/step




```
1/1 [==============================] - 0s 32ms/step
```



```
def predict_masks_with_url(new_model, url = ""):

    image = cv2.imread(url, cv2.IMREAD_UNCHANGED)
    image = image/255.
    image = cv2.resize(image, (256,256))
    predicted  = new_model.predict(image[np.newaxis,:,:,:])
    # predicted = tf.argmax(predicted, axis=-1)
    # predicted = tf.expand_dims(predicted, axis=-1)

    # #original segmentation map
    # image_mask = cv2.imread(X_test['mask'].iloc[6], cv2.IMREAD_UNCHANGED)
    # image_mask = cv2.resize(image_mask, (256,256))

    plt.figure(figsize=(10,6))
    plt.subplot(131)
    plt.imshow(image)
    plt.xlabel("original image")
    plt.subplot(132)
    plt.imshow(predicted[0,:,:,0])
    plt.xlabel("predicted mask")
```

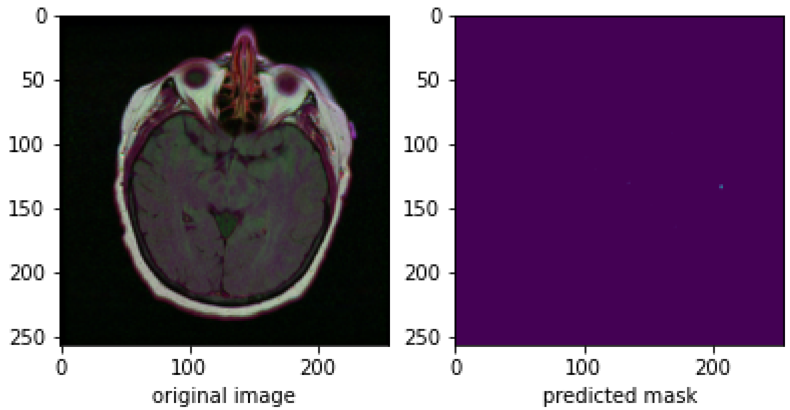
```
    plt.show()
```

```
X_train.iloc[10].values
```

```
array(['/content/kaggle_3m/TCGA_DU_7010_19860307/TCGA_DU_7010_19860307_23.tif',
       '/content/kaggle_3m/TCGA_DU_7010_19860307/TCGA_DU_7010_19860307_23_mask.tif'],
      dtype=object)
```

```
path = "/content/kaggle_3m/TCGA_DU_7010_19860307/TCGA_DU_7010_19860307_23.tif"
predict_masks_with_url(new_model, url = path, )
```

```
1/1 [==============================] - 0s 34ms/step
```



performance table

```
from prettytable import PrettyTable
x = PrettyTable()
x.field_names = ["Model","Backbon","loss", "val loss", "IOU score","val_IOU score", "epoch"]
x.add_rows(
    [
```

```
        ["UNET", "resnet_34", .1951, .2221, .70, .67,20],
        ["UNET", "inceptionv3", .1657, .2066, .74, .69, 20],
        ["UNET", "effiecientnetb1", .1654, .1827, .74, .72, 20],
        ["UNET", "efficientnetb4", .1678, .1760, .74, .73, 20],
        ["UNET", "senet154",.1466, .1624, .76, .75, 20],
        ["VNET", "NA", 1,1, .10, 00, 10 ],
        ["UNET", "NA", .2596, .3219, .62, .56, 40],
        ["UNET", "senet154",.1466, .1624, .85, .77, "20+"],
        ["UNet","attantion", .13, .1700, .77, .71, "40+"],
        ["UNET", "efficientb4" , .076, .088, .88, .84, "50+"]
    ]
)
print(x)
```

```
+-------+-----------------+--------+----------+-----------+---------------+-------+
| Model |     Backbon     |  loss  | val loss | IOU score | val_IOU score | epoch |
+-------+-----------------+--------+----------+-----------+---------------+-------+
|  UNET |    resnet_34    | 0.1951 |  0.2221  |    0.7    |      0.67     |   20  |
|  UNET |   inceptionv3   | 0.1657 |  0.2066  |    0.74   |      0.69     |   20  |
|  UNET | effiecientnetb1 | 0.1654 |  0.1827  |    0.74   |      0.72     |   20  |
|  UNET |  efficientnetb4 | 0.1678 |  0.176   |    0.74   |      0.73     |   20  |
|  UNET |     senet154    | 0.1466 |  0.1624  |    0.76   |      0.75     |   20  |
|  VNET |        NA       |   1    |    1     |    0.1    |       0       |   10  |
|  UNET |        NA       | 0.2596 |  0.3219  |    0.62   |      0.56     |   40  |
|  UNET |     senet154    | 0.1466 |  0.1624  |    0.85   |      0.77     |  20+  |
|  UNet |    attantion    |  0.13  |   0.17   |    0.77   |      0.71     |  40+  |
|  UNET |   efficientb4   | 0.076  |  0.088   |    0.88   |      0.84     |  50+  |
+-------+-----------------+--------+----------+-----------+---------------+-------+
```

conclusion

so we have used 8 models in which 7 models included UNet and its variation (like res34_unet) and 1 model is VNet. all model's performance is close and there is no clear winner except VNet(performance is very low compared to UNet models) .

First we trained all models for 20 epoch then check which model is performing better.

In all 8 models, UNet with senet154 and UNet with efficientnetb4 are performing better than other models so we train it for 50+ epochs (unless performance stop increasing) and see UNet with senet154 is giving a good result but it's taking too much time to train and **efficientnetb4 model** which is giving **.84 IOU score** which is highest over all models and also it's faster to train.

Colab paid products - Cancel contracts here