In [73]:
```
import os
import numpy as np
import cv2
from glob import glob

import tensorflow as tf
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Concatenate
from tensorflow.keras.layers import Conv2DTranspose
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import CSVLogger
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import TensorBoard
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import Recall
from tensorflow.keras.metrics import Precision
from tensorflow.keras import backend as K

from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
```

In [74]:
```
!pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\snand\anaconda3\envs\dl\lib\site-packages (4.8.1.78)
Requirement already satisfied: numpy>=1.19.3 in c:\users\snand\anaconda3\envs\dl\lib\site-packages (from opencv-python)
(1.22.4)
```

In [75]:
```
def conv_block(inputs, num_filters):
    x = Conv2D(num_filters, 3, padding="same")(inputs)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    x = Conv2D(num_filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    return x
```

In [76]:
```
def encoder_block(inputs, num_filters):
    x = conv_block(inputs, num_filters)
    p = MaxPool2D((2, 2))(x)
    return x, p
```

In [77]:
```
def decoder_block(inputs, skip_features, num_filters):
    x = Conv2DTranspose(num_filters, (2, 2), strides=2, padding="same")(inputs)
    x = Concatenate()([x, skip_features])
    x = conv_block(x, num_filters)
    return x
```

In [78]:
```
def build_unet(input_shape):
    inputs = Input(input_shape)
    s1, p1 = encoder_block(inputs, 64)
    s2, p2 = encoder_block(p1, 128)
    s3, p3 = encoder_block(p2, 256)
    s4, p4 = encoder_block(p3, 512)
    b1 = conv_block(p4, 1024)
    d1 = decoder_block(b1, s4, 512)
    d2 = decoder_block(d1, s3, 256)
    d3 = decoder_block(d2, s2, 128)
    d4 = decoder_block(d3, s1, 64)
    outputs = Conv2D(1, 1, padding="same", activation="sigmoid")(d4)
    model = Model(inputs, outputs)
    return model
```

In [79]:
```
H = 256
W = 256

def create_dir(path):
    if not os.path.exists(path):
        os.makedirs(path)
```

In [80]:
```
def shuffling(x, y):
    x, y = shuffle(x, y, random_state=42)
    return x, y
```

In [81]:
```
def load_data(dataset_path, split=0.2):
    images = sorted(glob(os.path.join(dataset_path, "ISIC2018_Task1-2_Training_Input/ISIC2018_Task1-2_Training_Input",
    masks = sorted(glob(os.path.join(dataset_path, "ISIC2018_Task1_Training_GroundTruth/ISIC2018_Task1_Training_GroundT

    test_size = int(len(images) * split)
```

```
    train_x, valid_x = train_test_split(images, test_size=test_size, random_state=42)
    train_y, valid_y = train_test_split(masks, test_size=test_size, random_state=42)

    train_x, test_x = train_test_split(train_x, test_size=test_size, random_state=42)
    train_y, test_y = train_test_split(train_y, test_size=test_size, random_state=42)

    return (train_x, train_y), (valid_x, valid_y), (test_x, test_y)
```

In [82]:
```
def read_image(path):
    path = path.decode()
    x = cv2.imread(path, cv2.IMREAD_COLOR)  ## (H, W, 3)
    x = cv2.resize(x, (W, H))
    x = x/255.0
    x = x.astype(np.float32)
    return x
```

In [83]:
```
def read_mask(path):
    path = path.decode()
    x = cv2.imread(path, cv2.IMREAD_GRAYSCALE)  ## (H, W)
    x = cv2.resize(x, (W, H))
    x = x/255.0
    x = x.astype(np.float32)                    ## (256, 256)
    x = np.expand_dims(x, axis=-1)              ## (256, 256, 1)
    return x
```

In [84]:
```
def tf_parse(x, y):
    def _parse(x, y):
        x = read_image(x)
        y = read_mask(y)
        return x, y

    x, y = tf.numpy_function(_parse, [x, y], [tf.float32, tf.float32])
    x.set_shape([H, W, 3])
    y.set_shape([H, W, 1])
    return x, y
```

In [85]:
```
def tf_dataset(X, Y, batch):
    dataset = tf.data.Dataset.from_tensor_slices((X, Y))
    dataset = dataset.map(tf_parse)
    dataset = dataset.batch(batch)
    dataset = dataset.prefetch(10)
    return dataset
```

In [86]:
```
def iou(y_true, y_pred):
    def f(y_true, y_pred):
        intersection = (y_true * y_pred).sum()
        union = y_true.sum() + y_pred.sum() - intersection
        x = (intersection + 1e-15) / (union + 1e-15)
        x = x.astype(np.float32)
        return x
    return tf.numpy_function(f, [y_true, y_pred], tf.float32)
```

In [87]:
```
smooth = 1e-15
def dice_coef(y_true, y_pred):
    y_true = tf.keras.layers.Flatten()(y_true)
    y_pred = tf.keras.layers.Flatten()(y_pred)
    intersection = tf.reduce_sum(y_true * y_pred)
    return (2. * intersection + smooth) / (tf.reduce_sum(y_true) + tf.reduce_sum(y_pred) + smooth)
```

In [88]:
```
def dice_loss(y_true, y_pred):
    return 1.0 - dice_coef(y_true, y_pred)
```

In [89]:
```
np.random.seed(42)
tf.random.set_seed(42)
```

In [90]:
```
create_dir("files")
```

In [91]:
```
batch_size = 4
lr = 1e-4 ## (0.0001)
num_epoch = 5
model_path = "files/model.h5"
csv_path = "files/data.csv"
```

In [92]:
```
dataset_path = "C:/Games/Data/"
(train_x, train_y), (valid_x, valid_y), (test_x, test_y) = load_data(dataset_path)
```

In [93]:
```
print(f"Train: {len(train_x)} - {len(train_y)}")
print(f"Valid: {len(valid_x)} - {len(valid_y)}")
print(f"Test: {len(test_x)} - {len(test_y)}")
```

```
Train: 1558 - 1558
Valid: 518 - 518
Test: 518 - 518
```

In [ ]:
```
```

In [94]:
```
train_dataset = tf_dataset(train_x, train_y, batch_size)
valid_dataset = tf_dataset(valid_x, valid_y, batch_size)

train_steps = len(train_x)//batch_size
valid_steps = len(valid_x)//batch_size
```

In [95]:
```
if len(train_x) % batch_size != 0:
    train_steps += 1

if len(valid_x) % batch_size != 0:
    valid_steps += 1
```

In [96]:
```
model = build_unet((H, W, 3))
metrics = [dice_coef, iou, Recall(), Precision()]
model.compile(loss="binary_crossentropy", optimizer=Adam(lr), metrics=metrics)
model.summary()
```

```
Model: "model_1"
```

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 256, 256, 3)	0	
conv2d_19 (Conv2D)	(None, 256, 256, 64)	1792	input_2[0][0]
batch_normalization_18 (BatchNo	(None, 256, 256, 64)	256	conv2d_19[0][0]
activation_18 (Activation)	(None, 256, 256, 64)	0	batch_normalization_18[0][0]
conv2d_20 (Conv2D)	(None, 256, 256, 64)	36928	activation_18[0][0]
batch_normalization_19 (BatchNo	(None, 256, 256, 64)	256	conv2d_20[0][0]
activation_19 (Activation)	(None, 256, 256, 64)	0	batch_normalization_19[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 128, 128, 64)	0	activation_19[0][0]
conv2d_21 (Conv2D)	(None, 128, 128, 128	73856	max_pooling2d_4[0][0]
batch_normalization_20 (BatchNo	(None, 128, 128, 128	512	conv2d_21[0][0]
activation_20 (Activation)	(None, 128, 128, 128	0	batch_normalization_20[0][0]
conv2d_22 (Conv2D)	(None, 128, 128, 128	147584	activation_20[0][0]
batch_normalization_21 (BatchNo	(None, 128, 128, 128	512	conv2d_22[0][0]
activation_21 (Activation)	(None, 128, 128, 128	0	batch_normalization_21[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 64, 64, 128)	0	activation_21[0][0]
conv2d_23 (Conv2D)	(None, 64, 64, 256)	295168	max_pooling2d_5[0][0]
batch_normalization_22 (BatchNo	(None, 64, 64, 256)	1024	conv2d_23[0][0]
activation_22 (Activation)	(None, 64, 64, 256)	0	batch_normalization_22[0][0]
conv2d_24 (Conv2D)	(None, 64, 64, 256)	590080	activation_22[0][0]
batch_normalization_23 (BatchNo	(None, 64, 64, 256)	1024	conv2d_24[0][0]
activation_23 (Activation)	(None, 64, 64, 256)	0	batch_normalization_23[0][0]

max_pooling2d_6 (MaxPooling2D)	(None, 32, 32, 256)	0	activation_23[0][0]
conv2d_25 (Conv2D)	(None, 32, 32, 512)	1180160	max_pooling2d_6[0][0]
batch_normalization_24 (BatchNo	(None, 32, 32, 512)	2048	conv2d_25[0][0]
activation_24 (Activation)	(None, 32, 32, 512)	0	batch_normalization_24[0][0]
conv2d_26 (Conv2D)	(None, 32, 32, 512)	2359808	activation_24[0][0]
batch_normalization_25 (BatchNo	(None, 32, 32, 512)	2048	conv2d_26[0][0]
activation_25 (Activation)	(None, 32, 32, 512)	0	batch_normalization_25[0][0]
max_pooling2d_7 (MaxPooling2D)	(None, 16, 16, 512)	0	activation_25[0][0]
conv2d_27 (Conv2D)	(None, 16, 16, 1024)	4719616	max_pooling2d_7[0][0]
batch_normalization_26 (BatchNo	(None, 16, 16, 1024)	4096	conv2d_27[0][0]
activation_26 (Activation)	(None, 16, 16, 1024)	0	batch_normalization_26[0][0]
conv2d_28 (Conv2D)	(None, 16, 16, 1024)	9438208	activation_26[0][0]
batch_normalization_27 (BatchNo	(None, 16, 16, 1024)	4096	conv2d_28[0][0]
activation_27 (Activation)	(None, 16, 16, 1024)	0	batch_normalization_27[0][0]
conv2d_transpose_4 (Conv2DTrans	(None, 32, 32, 512)	2097664	activation_27[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 1024)	0	conv2d_transpose_4[0][0] activation_25[0][0]
conv2d_29 (Conv2D)	(None, 32, 32, 512)	4719104	concatenate_4[0][0]
batch_normalization_28 (BatchNo	(None, 32, 32, 512)	2048	conv2d_29[0][0]
activation_28 (Activation)	(None, 32, 32, 512)	0	batch_normalization_28[0][0]
conv2d_30 (Conv2D)	(None, 32, 32, 512)	2359808	activation_28[0][0]
batch_normalization_29 (BatchNo	(None, 32, 32, 512)	2048	conv2d_30[0][0]
activation_29 (Activation)	(None, 32, 32, 512)	0	batch_normalization_29[0][0]

conv2d_transpose_5 (Conv2DTrans	(None, 64, 64, 256)	524544	activation_29[0][0]
concatenate_5 (Concatenate)	(None, 64, 64, 512)	0	conv2d_transpose_5[0][0] activation_23[0][0]
conv2d_31 (Conv2D)	(None, 64, 64, 256)	1179904	concatenate_5[0][0]
batch_normalization_30 (BatchNo	(None, 64, 64, 256)	1024	conv2d_31[0][0]
activation_30 (Activation)	(None, 64, 64, 256)	0	batch_normalization_30[0][0]
conv2d_32 (Conv2D)	(None, 64, 64, 256)	590080	activation_30[0][0]
batch_normalization_31 (BatchNo	(None, 64, 64, 256)	1024	conv2d_32[0][0]
activation_31 (Activation)	(None, 64, 64, 256)	0	batch_normalization_31[0][0]
conv2d_transpose_6 (Conv2DTrans	(None, 128, 128, 128	131200	activation_31[0][0]
concatenate_6 (Concatenate)	(None, 128, 128, 256	0	conv2d_transpose_6[0][0] activation_21[0][0]
conv2d_33 (Conv2D)	(None, 128, 128, 128	295040	concatenate_6[0][0]
batch_normalization_32 (BatchNo	(None, 128, 128, 128	512	conv2d_33[0][0]
activation_32 (Activation)	(None, 128, 128, 128	0	batch_normalization_32[0][0]
conv2d_34 (Conv2D)	(None, 128, 128, 128	147584	activation_32[0][0]
batch_normalization_33 (BatchNo	(None, 128, 128, 128	512	conv2d_34[0][0]
activation_33 (Activation)	(None, 128, 128, 128	0	batch_normalization_33[0][0]
conv2d_transpose_7 (Conv2DTrans	(None, 256, 256, 64)	32832	activation_33[0][0]
concatenate_7 (Concatenate)	(None, 256, 256, 128	0	conv2d_transpose_7[0][0] activation_19[0][0]
conv2d_35 (Conv2D)	(None, 256, 256, 64)	73792	concatenate_7[0][0]
batch_normalization_34 (BatchNo	(None, 256, 256, 64)	256	conv2d_35[0][0]
activation_34 (Activation)	(None, 256, 256, 64)	0	batch_normalization_34[0][0]

```
conv2d_36 (Conv2D)              (None, 256, 256, 64) 36928       activation_34[0][0]
batch_normalization_35 (BatchNo (None, 256, 256, 64) 256         conv2d_36[0][0]
activation_35 (Activation)      (None, 256, 256, 64) 0           batch_normalization_35[0][0]
conv2d_37 (Conv2D)              (None, 256, 256, 1)  65          activation_35[0][0]
==================================================================================================
Total params: 31,055,297
Trainable params: 31,043,521
Non-trainable params: 11,776
```

In [97]:
```
callbacks = [
    ModelCheckpoint(model_path, verbose=1, save_best_only=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=5, min_lr=1e-7, verbose=1),
    CSVLogger(csv_path),
    TensorBoard(),
    EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=False)
]
```

In [98]:
```
model.fit(
    train_dataset,
    epochs=num_epoch,
    validation_data=valid_dataset,
    steps_per_epoch=train_steps,
    validation_steps=valid_steps,
    callbacks=callbacks
)
```

```
Epoch 1/5
390/390 [==============================] - 1425s 4s/step - loss: 0.3225 - dice_coef: 0.5473 - iou: 0.3873 - recall: 0.6
156 - precision: 0.7564 - val_loss: 0.2798 - val_dice_coef: 0.6319 - val_iou: 0.4730 - val_recall: 0.6460 - val_precisi
on: 0.8301

Epoch 00001: val_loss improved from inf to 0.27977, saving model to files\model.h5
Epoch 2/5
390/390 [==============================] - 1554s 4s/step - loss: 0.2357 - dice_coef: 0.6448 - iou: 0.4857 - recall: 0.7
067 - precision: 0.8497 - val_loss: 0.2193 - val_dice_coef: 0.6912 - val_iou: 0.5409 - val_recall: 0.7246 - val_precisi
on: 0.8818

Epoch 00002: val_loss improved from 0.27977 to 0.21928, saving model to files\model.h5
Epoch 3/5
390/390 [==============================] - 1562s 4s/step - loss: 0.2034 - dice_coef: 0.6829 - iou: 0.5294 - recall: 0.7
483 - precision: 0.8698 - val_loss: 0.2136 - val_dice_coef: 0.6923 - val_iou: 0.5425 - val_recall: 0.7339 - val_precisi
on: 0.8698

Epoch 00003: val_loss improved from 0.21928 to 0.21355, saving model to files\model.h5
Epoch 4/5
390/390 [==============================] - 1539s 4s/step - loss: 0.1833 - dice_coef: 0.7107 - iou: 0.5618 - recall: 0.7
764 - precision: 0.8801 - val_loss: 0.1983 - val_dice_coef: 0.7136 - val_iou: 0.5682 - val_recall: 0.7555 - val_precisi
on: 0.8852

Epoch 00004: val_loss improved from 0.21355 to 0.19830, saving model to files\model.h5
Epoch 5/5
390/390 [==============================] - 1532s 4s/step - loss: 0.1680 - dice_coef: 0.7330 - iou: 0.5886 - recall: 0.7
974 - precision: 0.8888 - val_loss: 0.1921 - val_dice_coef: 0.7202 - val_iou: 0.5760 - val_recall: 0.7624 - val_precisi
on: 0.8841

Epoch 00005: val_loss improved from 0.19830 to 0.19206, saving model to files\model.h5
```

Out[98]: `<tensorflow.python.keras.callbacks.History at 0x1e71cc01490>`

In [ ]: