

Melanoma Diagnosis Using Lesion Segmentation

Unet FCN

Computer vision

Mostafa EL-Zawawy 5311, Omar EL-Hariry 5360, Malak Kassem 5979

Unet

Model's architecture

The network architecture is illustrated in Figure 1. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64- component feature vector to the desired number of classes. In total the network has 23 convolutional layers. To allow a seamless tiling of the output segmentation map (see Figure 2), it is important to select the input tile size such that all 2x2 max-pooling operations are applied to a layer with an even x- and y-size

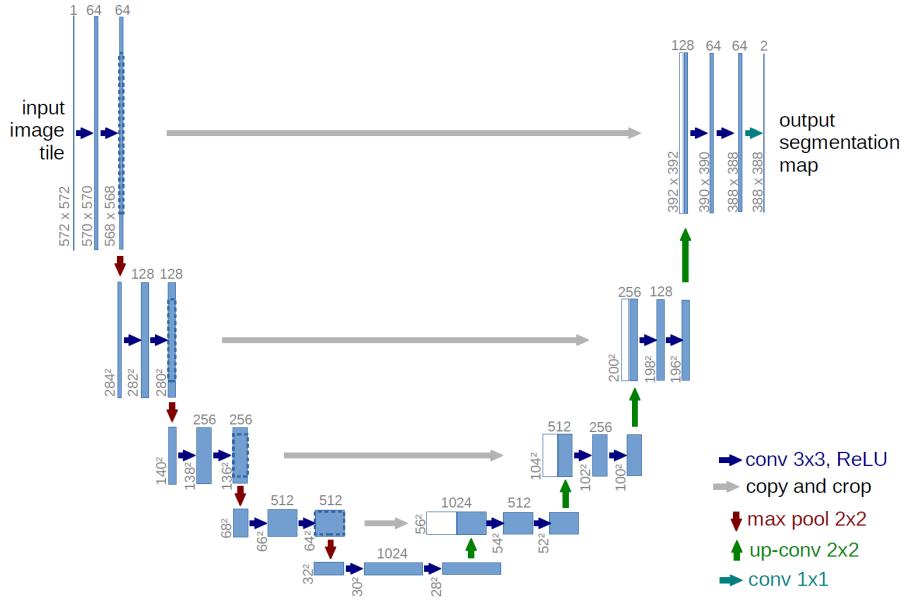


Figure 1: Unet architecture

Training

Learning rate=0.1

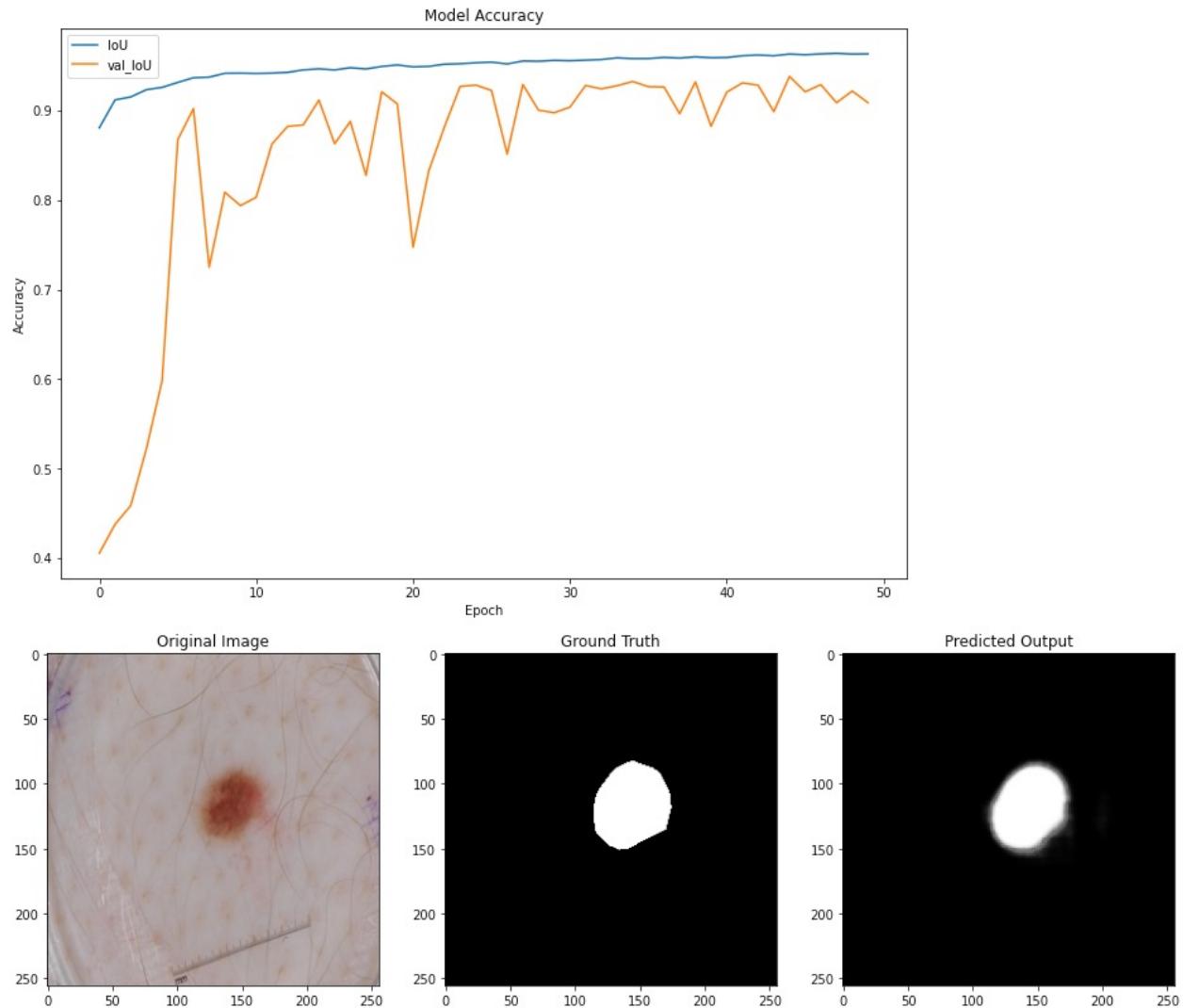


Figure 2: Unet accuracy result after 50 epoch with Lr=0.1

IoU: 90.31%

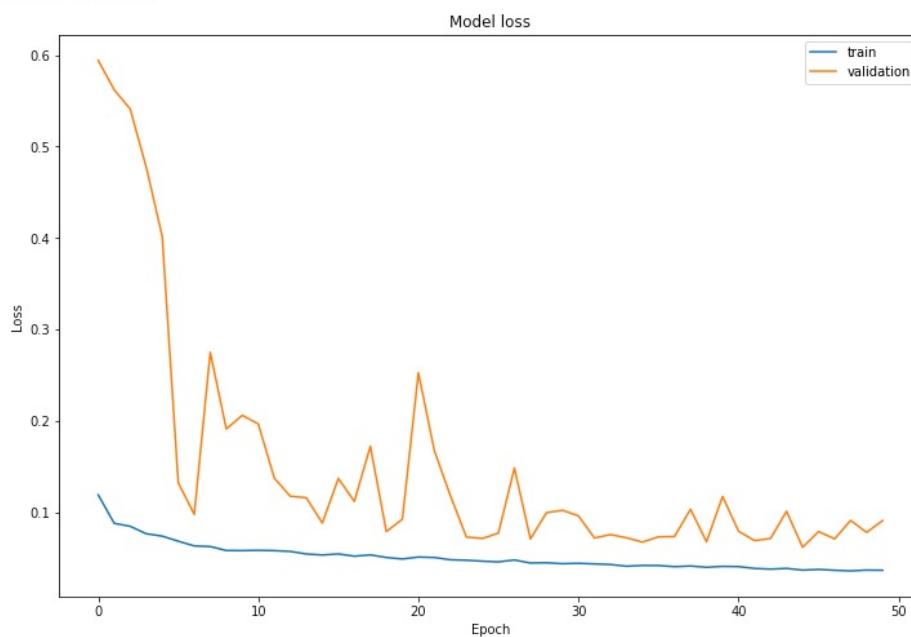


Figure 3: Unet loss result after 50 epoch with Lr=0.1

Learning rate=0.001

IoU accuracy=91.90

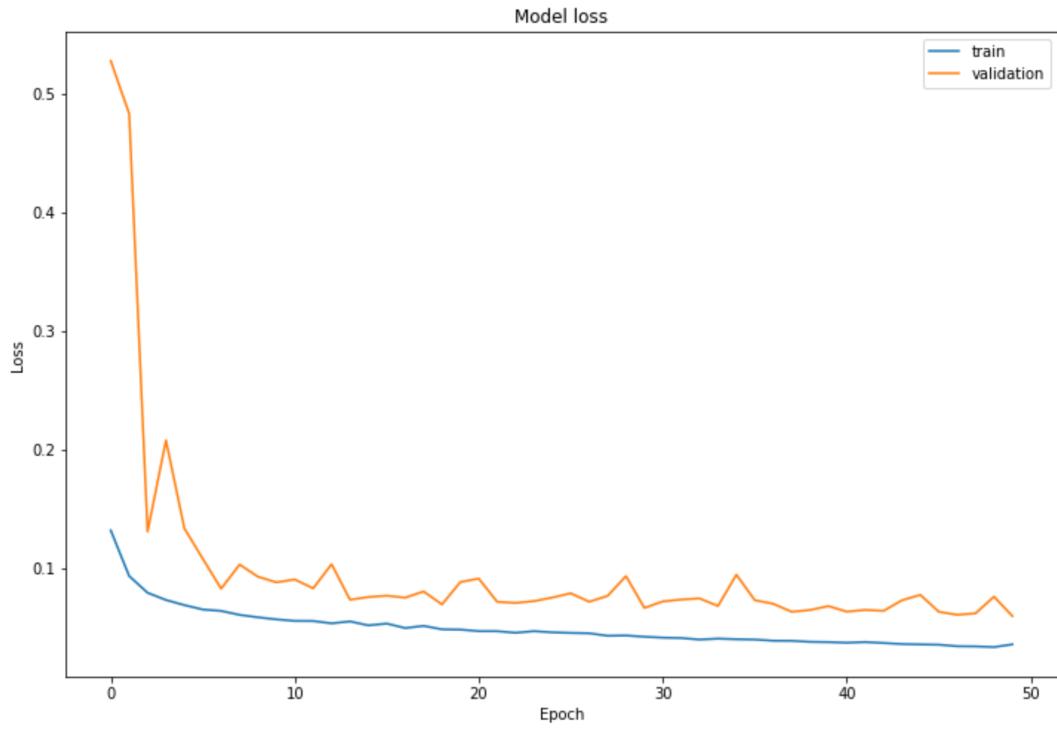


Figure 4: Unet loss result after 50 epoch with Lr=0.001

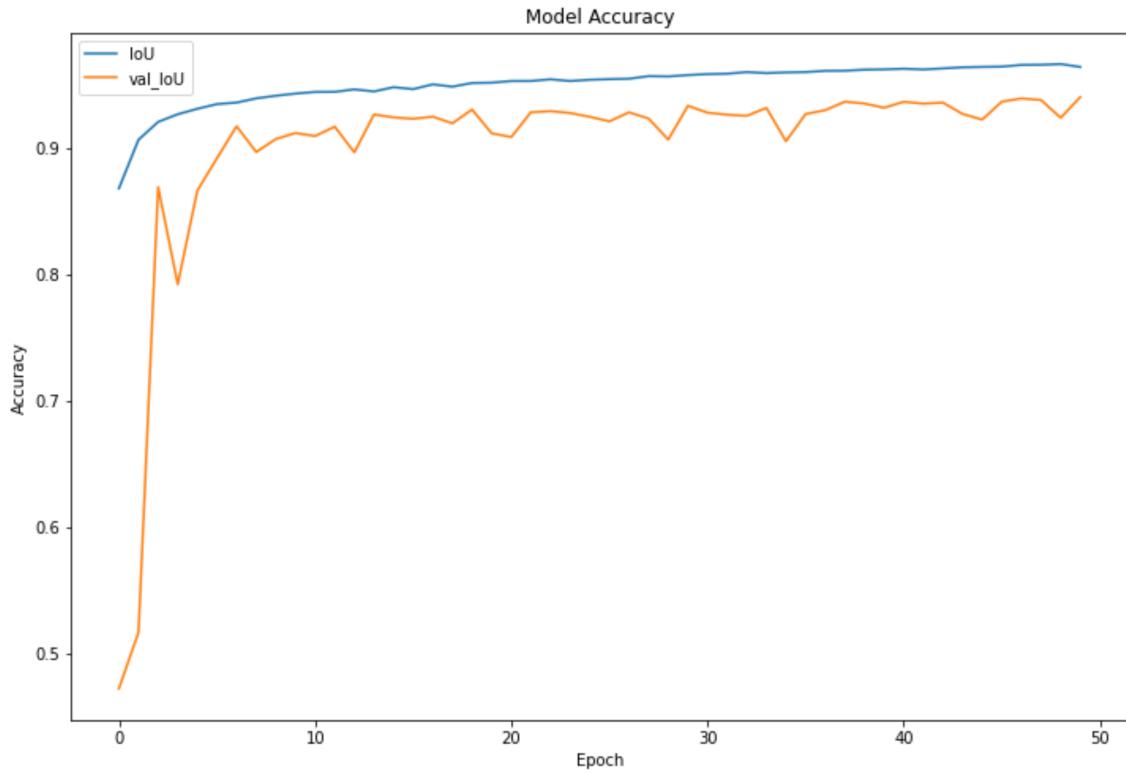


Figure 5: Unet loss result after 50 epoch with Lr=0.001

Learning rate=0.00001

IoU accuracy=90.94

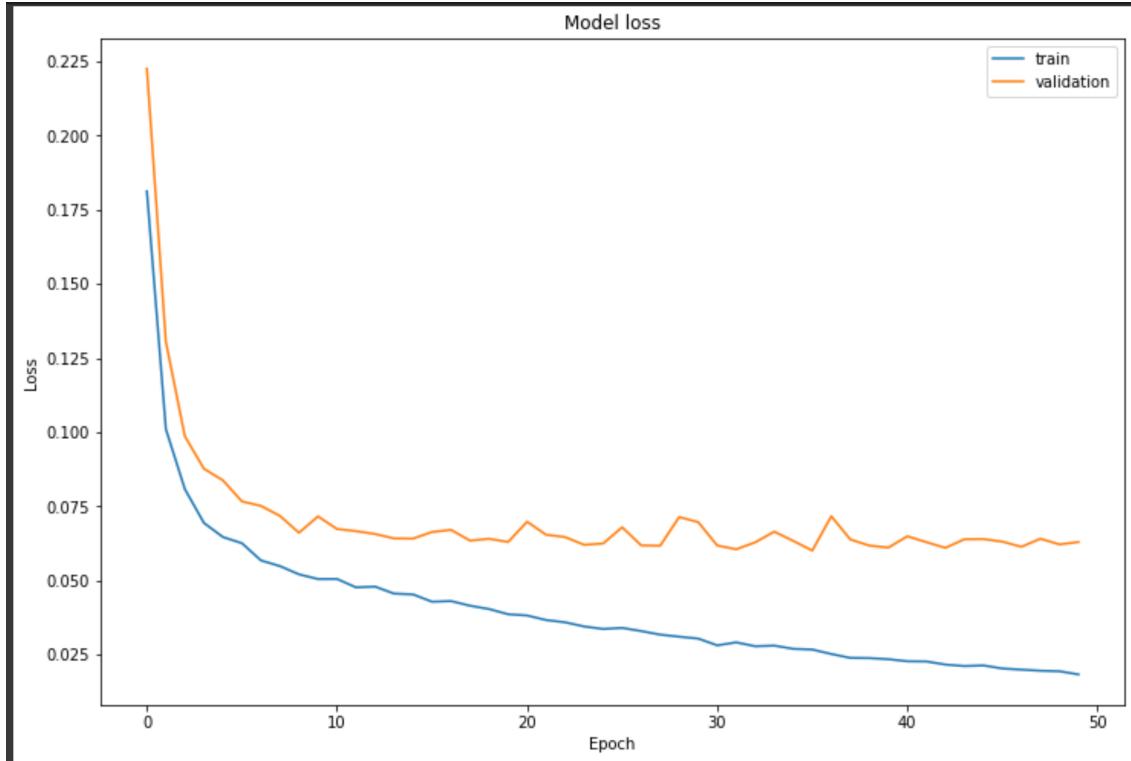


Figure 6: Unet loss result after 50 epoch with Lr=0.00001

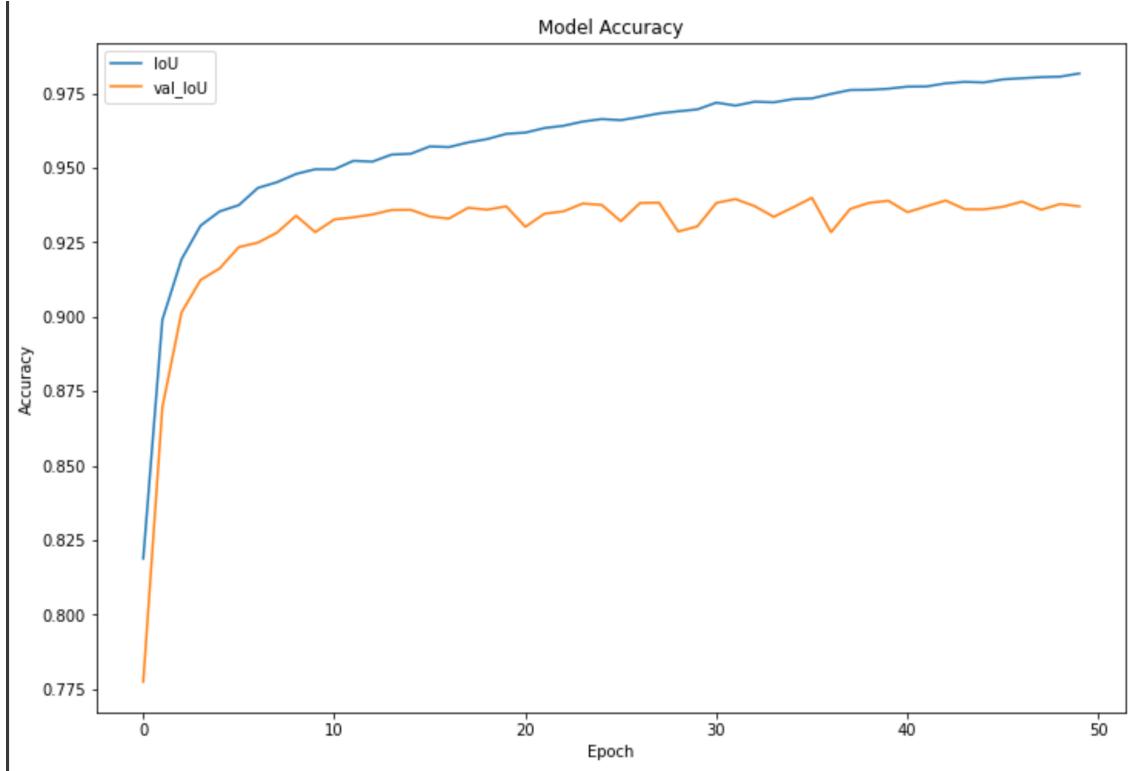


Figure 7: Unet loss result after 50 epoch with Lr=0.00001

Unet cases

Good results

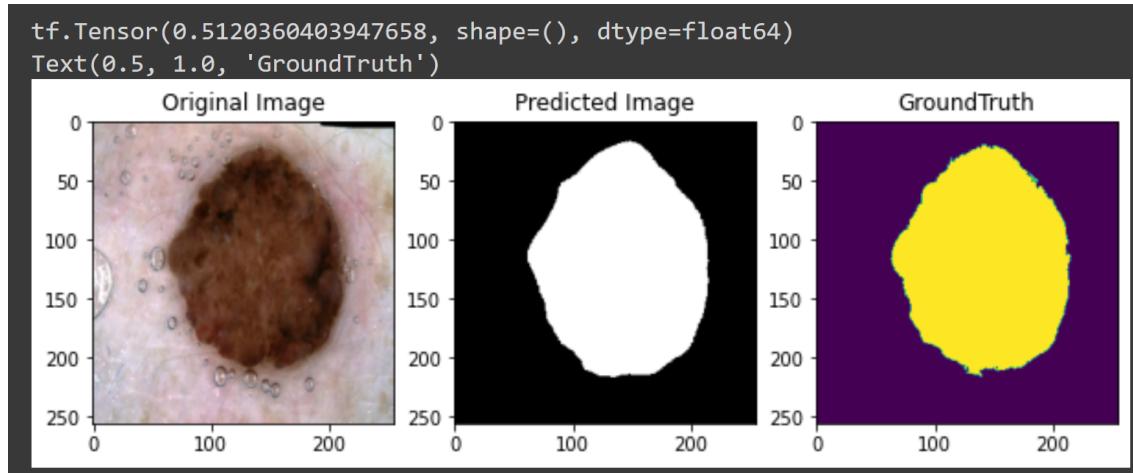


Figure 8: Good result for unet

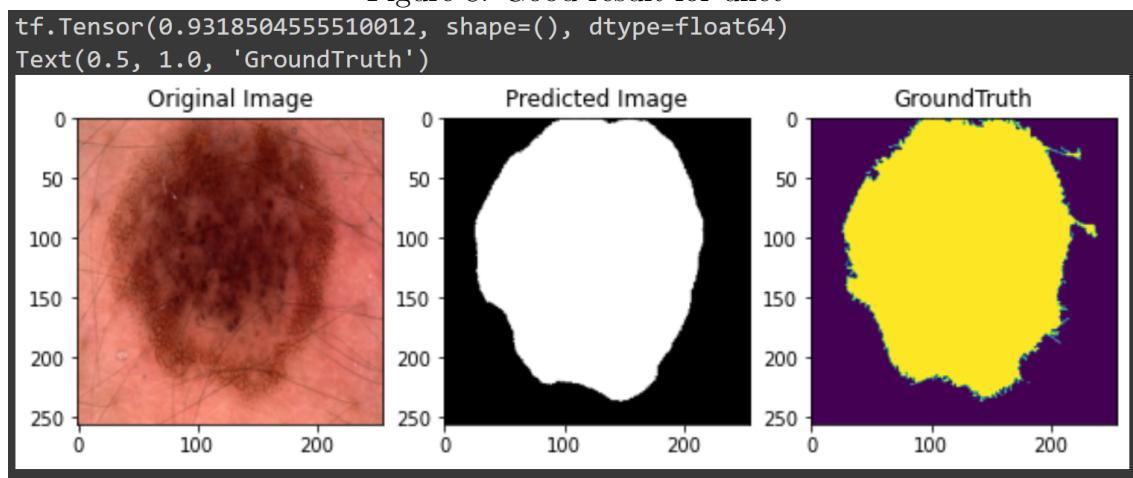


Figure 9: Good result for unet

Bad results

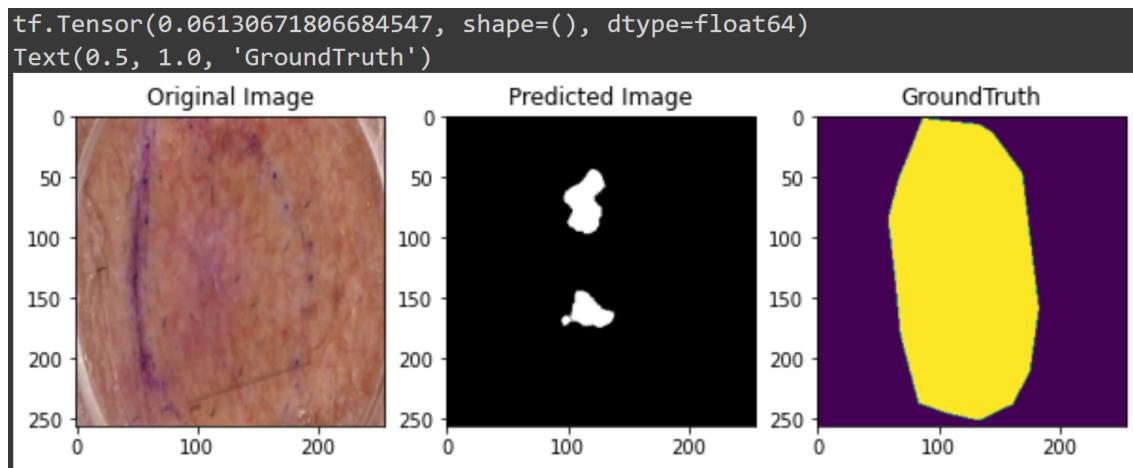


Figure 10: Bad result for unet

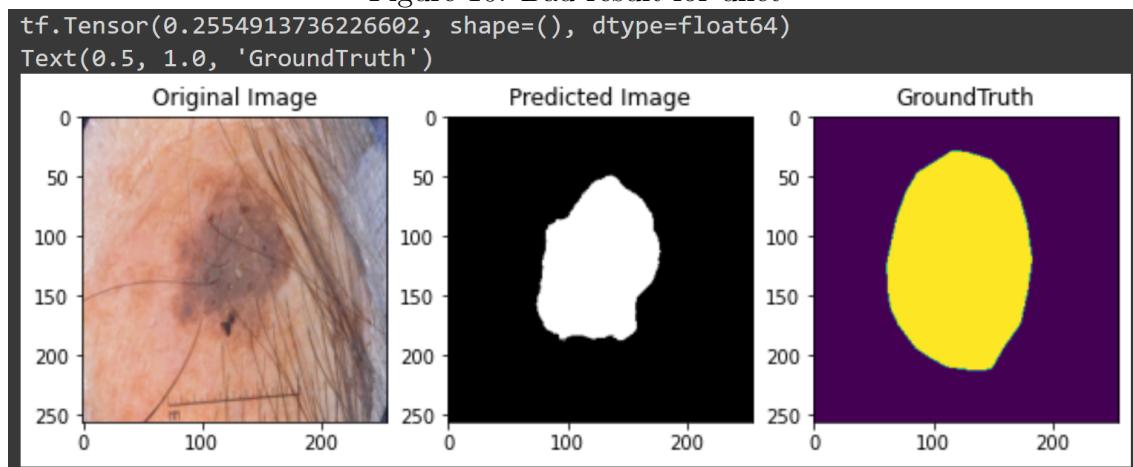


Figure 11: Bad result for unet

FCN

Training from scratch

Learning rate=0.1

Suffers from overfitting where training IoU is 78% and validation IoU is 56%

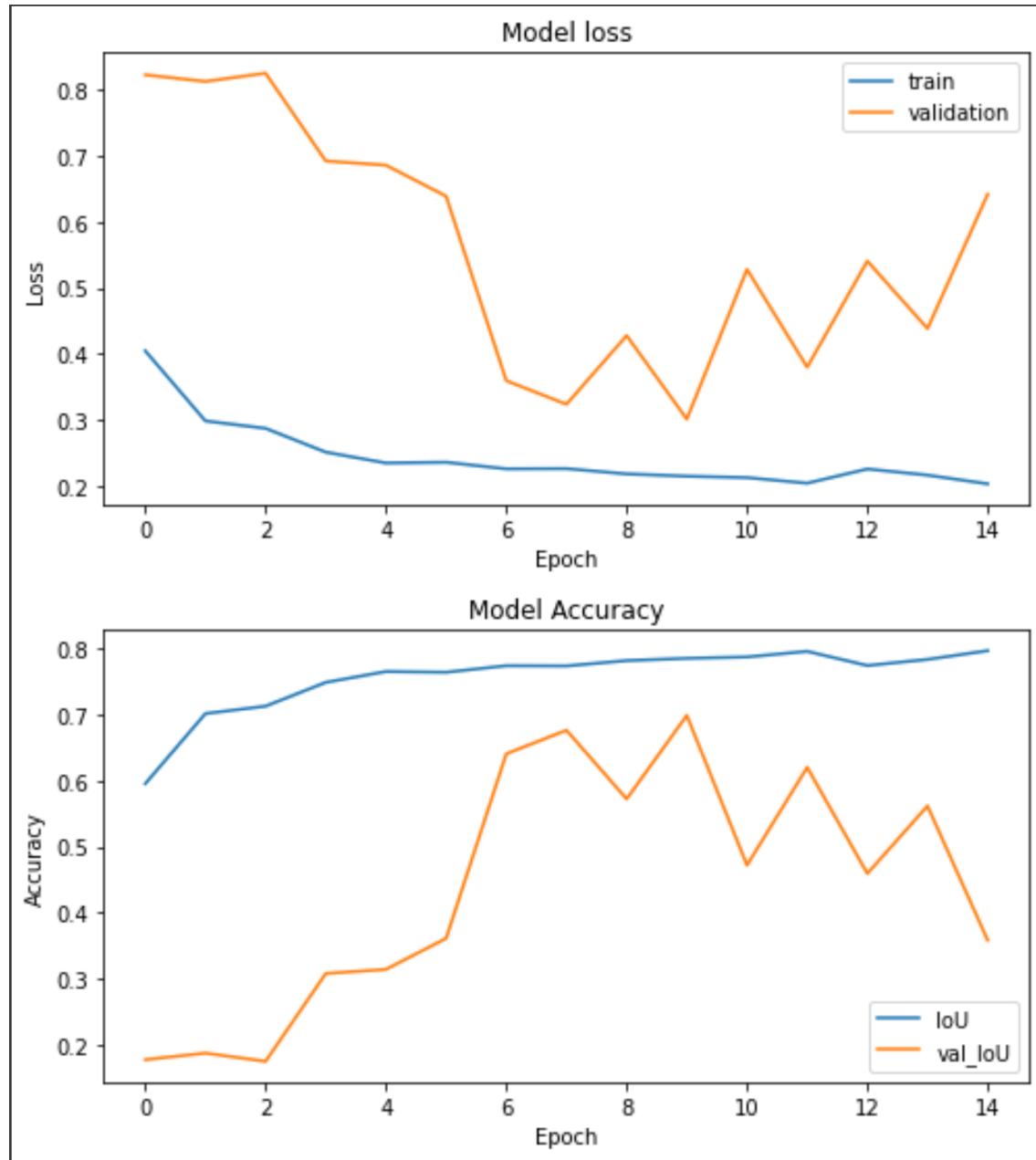


Figure 12: Fcn results after 15(early stopping) epoch with Lr=0.1

Learning rate=0.0001

Training IoU 44% Validation IoU 35% Overall moderate performance but still low IoU percentage and slow training speed and needs more training

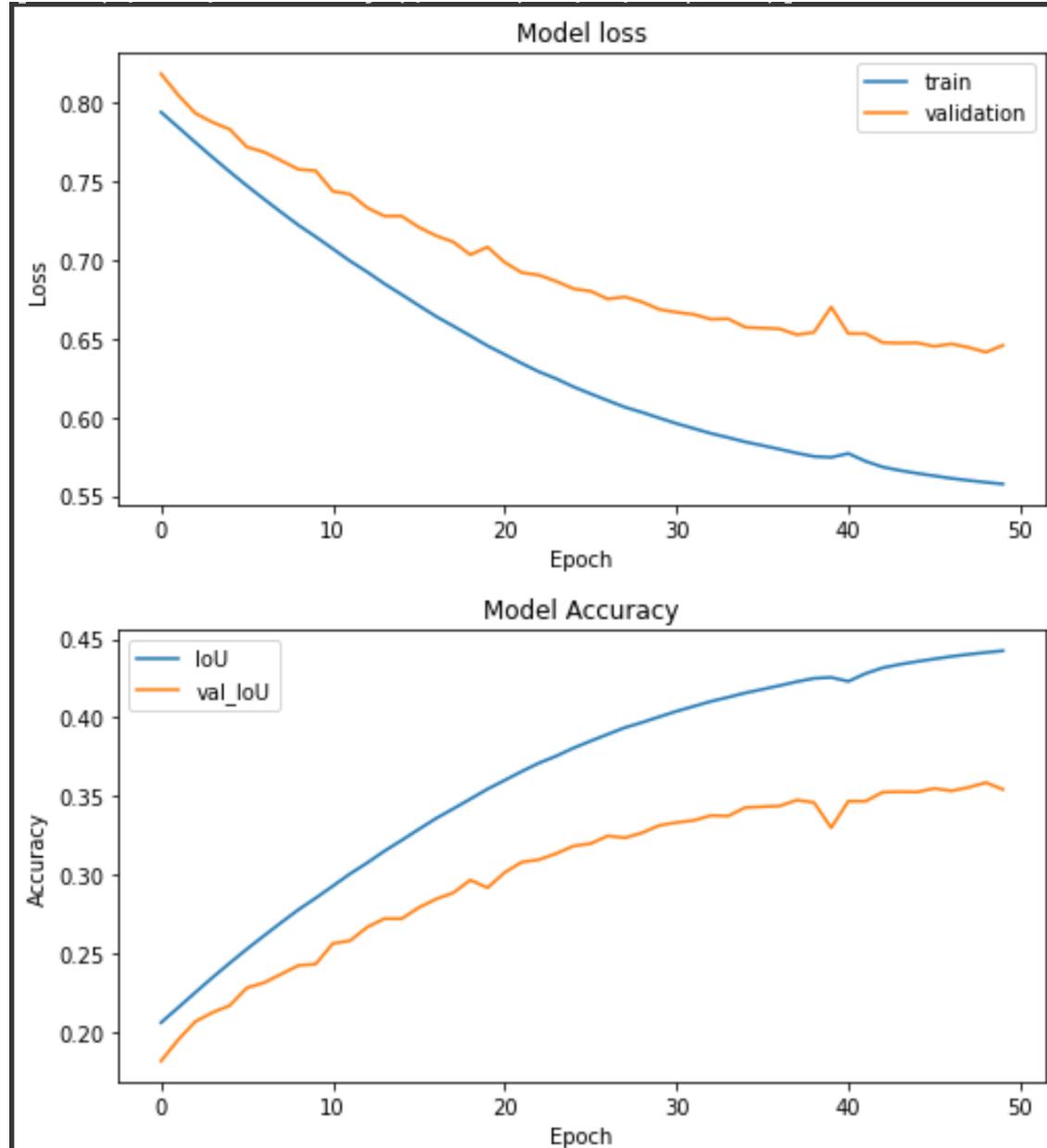


Figure 13: Fcn results after 15(early stopping) epoch with Lr=0.0001

Learning rate=0.003

Trains well and fast with no overfitting Training IoU 84% , Validation IoU 77%, Test IoU 75%

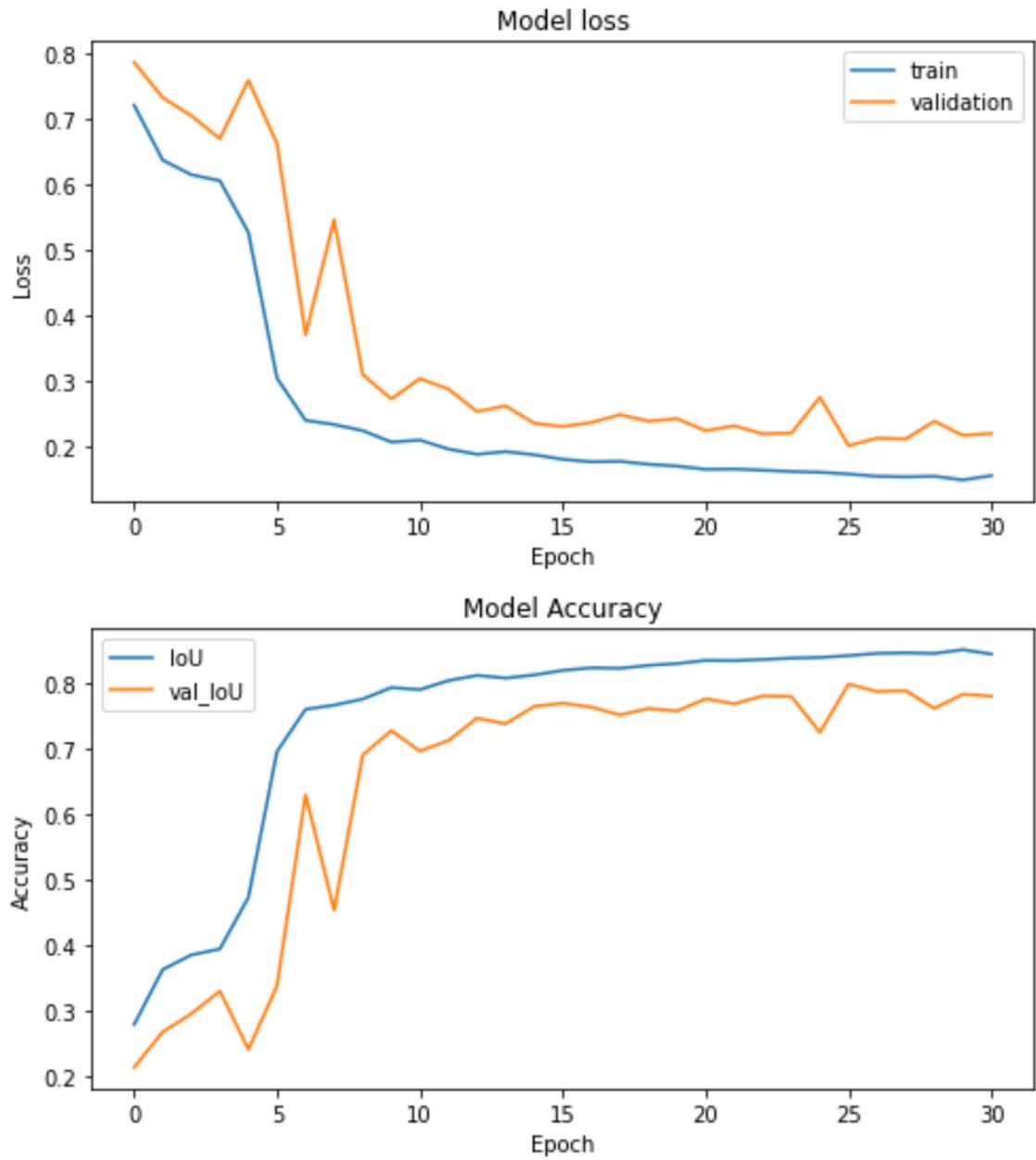


Figure 14: Fcn results after 31(early stopping) epoch with Lr=0.003

FCN cases

Good results

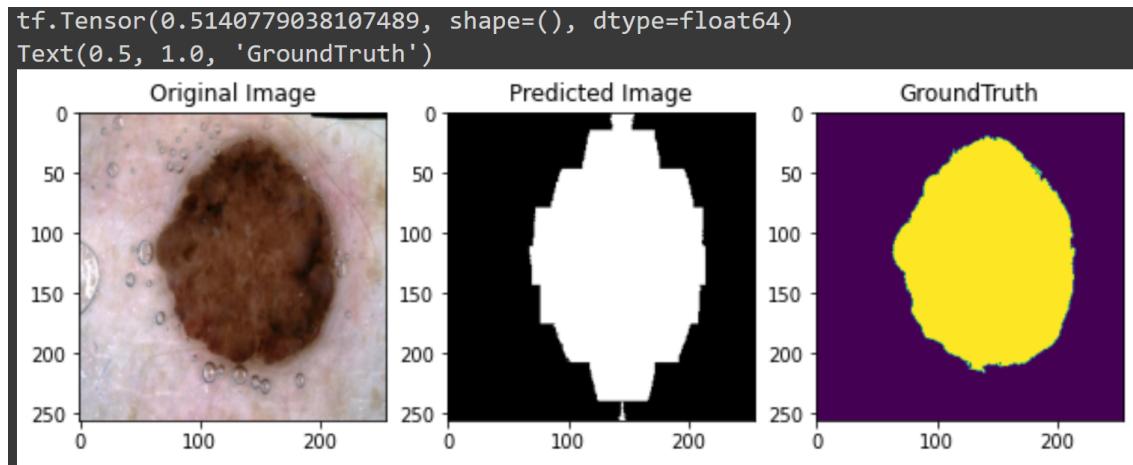


Figure 15: Good result for fcn

Bad results

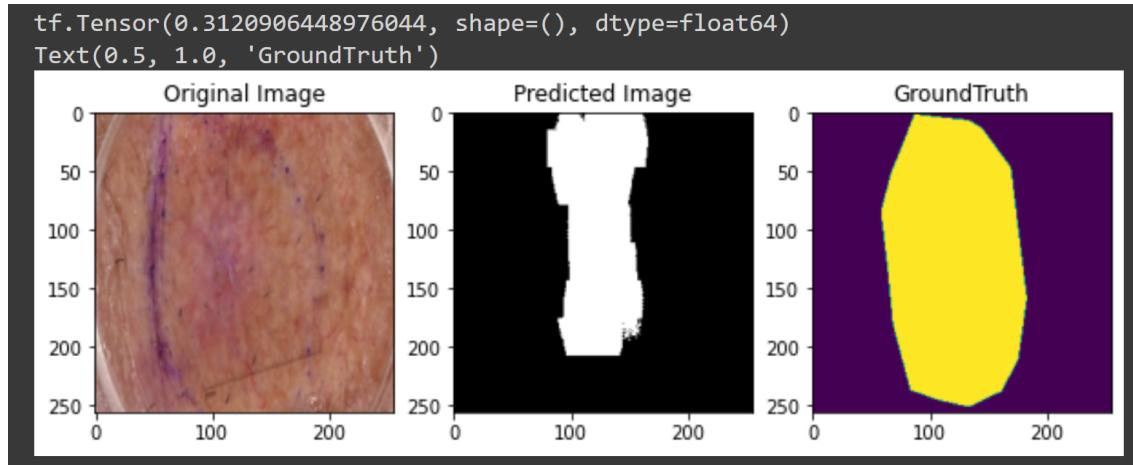


Figure 16: Bad result for fcn

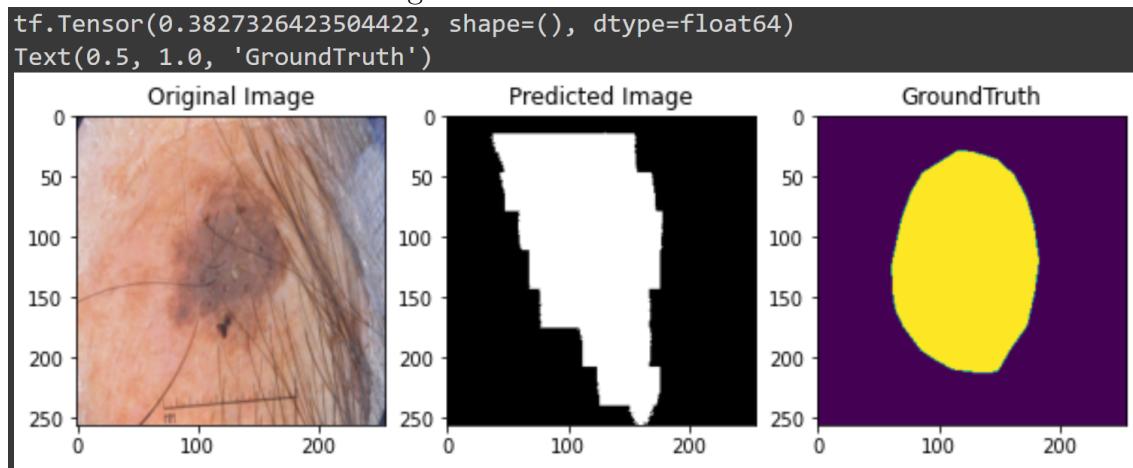


Figure 17: Bad result for fcn

FCN pretrained

```
#Loading VGG network
#Initializing weights to imagenet
img_input = Input(shape=(256, 256, 3))
vgg = vgg16.VGG16(include_top=False,weights = 'imagenet', input_tensor=img_input)
vgg.summary()
```

Figure 18: Loading VGG16 network and initializing weights to imagenet weights

Learning rate=0.1

```
#Learning rate 1/
model = Sequential()
model.add(vgg)
model.add(Conv2D(filters=512, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2D(filters=1, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2DTranspose(filters=1 ,padding='same', kernel_size=(64,64),strides=(32,32) ,activation="sigmoid"))
model.summary()
model.compile(loss=[jaccard_distance],
              optimizer=tf.keras.optimizers.Adam(learning_rate=0.1),
              metrics=[IoU])
```

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 8, 8, 512)	14714688
conv2d (Conv2D)	(None, 8, 8, 512)	262656
conv2d_1 (Conv2D)	(None, 8, 8, 512)	262656
conv2d_2 (Conv2D)	(None, 8, 8, 1)	513
conv2d_transpose (Conv2DTran	(None, 256, 256, 1)	4097

Total params: 15,244,610
Trainable params: 15,244,610
Non-trainable params: 0

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=5, restore_best_weights = True)
checkpoint = ModelCheckpoint("best_model.hdf5", monitor='loss', verbose=1, save_best_only=True, save_freq=5 * 128, mode='auto',
                            batch_size=16,
                            epochs=50, callbacks=[es, checkpoint])

2022-02-01 12:35:28.902663: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
Epoch 1/50
2022-02-01 12:35:30.856836: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005
125/125 [=====] - 28s 145ms/step - loss: 0.2132 - IoU: 0.7868 - val_loss: 0.1832 - val_IoU: 0.8168
Epoch 2/50
125/125 [=====] - 17s 136ms/step - loss: 0.2052 - IoU: 0.7948 - val_loss: 0.1832 - val_IoU: 0.8168
Epoch 3/50
125/125 [=====] - 17s 135ms/step - loss: 0.2052 - IoU: 0.7948 - val_loss: 0.1832 - val_IoU: 0.8168
Epoch 4/50
125/125 [=====] - 17s 136ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1835 - val_IoU: 0.8165
Epoch 5/50
125/125 [=====] - 17s 135ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1846 - val_IoU: 0.8154
Epoch 6/50
15/125 [==>.....] - ETA: 14s - loss: 0.1990 - IoU: 0.8010
Epoch 00006: loss improved from inf to 0.19905, saving model to best_model.hdf5
125/125 [=====] - 17s 138ms/step - loss: 0.2052 - IoU: 0.7948 - val_loss: 0.1835 - val_IoU: 0.8165
Epoch 7/50
125/125 [=====] - 17s 135ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1832 - val_IoU: 0.8168
Epoch 00007: early stopping
```

Figure 19: Learning rate is too high so it overshoots and doesn't learn

Learning rate=0.001

```

model = Sequential()
model.add(vgg)
model.add(Conv2D(filters=512, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2D(filters=1, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2DTranspose(filters=1 ,padding='same', kernel_size=(64,64),strides=(32,32) ,activation="sigmoid"))
model.summary()
model.compile(loss=[jaccard_distance],
              optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              metrics=[IoU])

Model: "sequential_1"
Layer (type)          Output Shape         Param #
vgg16 (Functional)    (None, 8, 512)       14714688
conv2d_3 (Conv2D)     (None, 8, 512)       262656
conv2d_4 (Conv2D)     (None, 8, 512)       262656
conv2d_5 (Conv2D)     (None, 8, 8, 1)      513
conv2d_transpose_1 (Conv2DTr) (None, 256, 256, 1) 4097
=====
Total params: 15,244,610
Trainable params: 15,244,610
Non-trainable params: 0
=====
Epoch 1/50
125/125 [=====] - 18s 137ms/step - loss: 0.3407 - IoU: 0.6593 - val_loss: 0.3358 - val_IoU: 0.6642
Epoch 2/50
125/125 [=====] - 17s 136ms/step - loss: 0.3245 - IoU: 0.6755 - val_loss: 0.3182 - val_IoU: 0.6818
Epoch 3/50
125/125 [=====] - 17s 135ms/step - loss: 0.3090 - IoU: 0.6910 - val_loss: 0.3014 - val_IoU: 0.6986
Epoch 4/50
125/125 [=====] - 17s 135ms/step - loss: 0.2945 - IoU: 0.7055 - val_loss: 0.2857 - val_IoU: 0.7143
Epoch 5/50
125/125 [=====] - 17s 135ms/step - loss: 0.2812 - IoU: 0.7188 - val_loss: 0.2714 - val_IoU: 0.7286
Epoch 6/50
15/125 [=>.....] - ETA: 14s - loss: 0.2741 - IoU: 0.7259
Epoch 00006: loss improved from inf to 0.27414, saving model to best_model.hdf5
125/125 [=====] - 17s 139ms/step - loss: 0.2694 - IoU: 0.7306 - val_loss: 0.2587 - val_IoU: 0.7413
Epoch 7/50
125/125 [=====] - 17s 135ms/step - loss: 0.2590 - IoU: 0.7410 - val_loss: 0.2474 - val_IoU: 0.7526
Epoch 8/50
125/125 [=====] - 17s 136ms/step - loss: 0.2500 - IoU: 0.7500 - val_loss: 0.2377 - val_IoU: 0.7623
Epoch 9/50
125/125 [=====] - 17s 135ms/step - loss: 0.2424 - IoU: 0.7576 - val_loss: 0.2294 - val_IoU: 0.7706
Epoch 10/50
125/125 [=====] - 17s 135ms/step - loss: 0.2359 - IoU: 0.7641 - val_loss: 0.2222 - val_IoU: 0.7778
Epoch 11/50
30/125 [=====] - ETA: 12s - loss: 0.2310 - IoU: 0.7690
Epoch 00011: loss improved from 0.27414 to 0.23102, saving model to best_model.hdf5
125/125 [=====] - 17s 139ms/step - loss: 0.2305 - IoU: 0.7695 - val_loss: 0.2162 - val_IoU: 0.7838
Epoch 12/50
125/125 [=====] - 17s 136ms/step - loss: 0.2261 - IoU: 0.7739 - val_loss: 0.2112 - val_IoU: 0.7888
Epoch 13/50
125/125 [=====] - 17s 135ms/step - loss: 0.2223 - IoU: 0.7777 - val_loss: 0.2070 - val_IoU: 0.7930
Epoch 14/50
125/125 [=====] - 17s 136ms/step - loss: 0.2193 - IoU: 0.7807 - val_loss: 0.2033 - val_IoU: 0.7967
Epoch 00036: loss improved from 0.20863 to 0.20629, saving model to best_model.hdf5
125/125 [=====] - 17s 139ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1842 - val_IoU: 0.8158
Epoch 37/50
125/125 [=====] - 17s 136ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1841 - val_IoU: 0.8159
Epoch 38/50
125/125 [=====] - 17s 135ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1840 - val_IoU: 0.8160
Epoch 39/50
125/125 [=====] - 17s 136ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1840 - val_IoU: 0.8160
Epoch 40/50
125/125 [=====] - 17s 135ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1839 - val_IoU: 0.8161
Epoch 41/50
120/125 [=====>..] - ETA: 0s - loss: 0.2044 - IoU: 0.7956
Epoch 00041: loss improved from 0.20629 to 0.20445, saving model to best_model.hdf5
125/125 [=====] - 17s 139ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1839 - val_IoU: 0.8161
Epoch 42/50
125/125 [=====] - 17s 136ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1838 - val_IoU: 0.8162
Epoch 43/50
125/125 [=====] - 17s 135ms/step - loss: 0.2051 - IoU: 0.7949 - val_loss: 0.1838 - val_IoU: 0.8162
Epoch 44/50
125/125 [=====] - 17s 136ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1838 - val_IoU: 0.8162
Epoch 45/50
125/125 [=====] - 17s 135ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1838 - val_IoU: 0.8162
Epoch 46/50
125/125 [=====] - 17s 135ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1837 - val_IoU: 0.8163
Epoch 47/50
10/125 [=>.....] - ETA: 15s - loss: 0.1875 - IoU: 0.8125
Epoch 00047: loss improved from 0.20445 to 0.18752, saving model to best_model.hdf5
125/125 [=====] - 17s 139ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1837 - val_IoU: 0.8163
Epoch 48/50
125/125 [=====] - 17s 135ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1837 - val_IoU: 0.8163
Epoch 49/50
125/125 [=====] - 17s 135ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1837 - val_IoU: 0.8163
Epoch 50/50
125/125 [=====] - 17s 135ms/step - loss: 0.2050 - IoU: 0.7950 - val_loss: 0.1837 - val_IoU: 0.8163
=====

score = model.evaluate(x_test, y_test.astype(np.float32), verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], score[1]*100))

```

IoU: 76.60%

Learning rate=0.001

Training IoU 79%, Validation IoU 81% ,Test IoU 76%

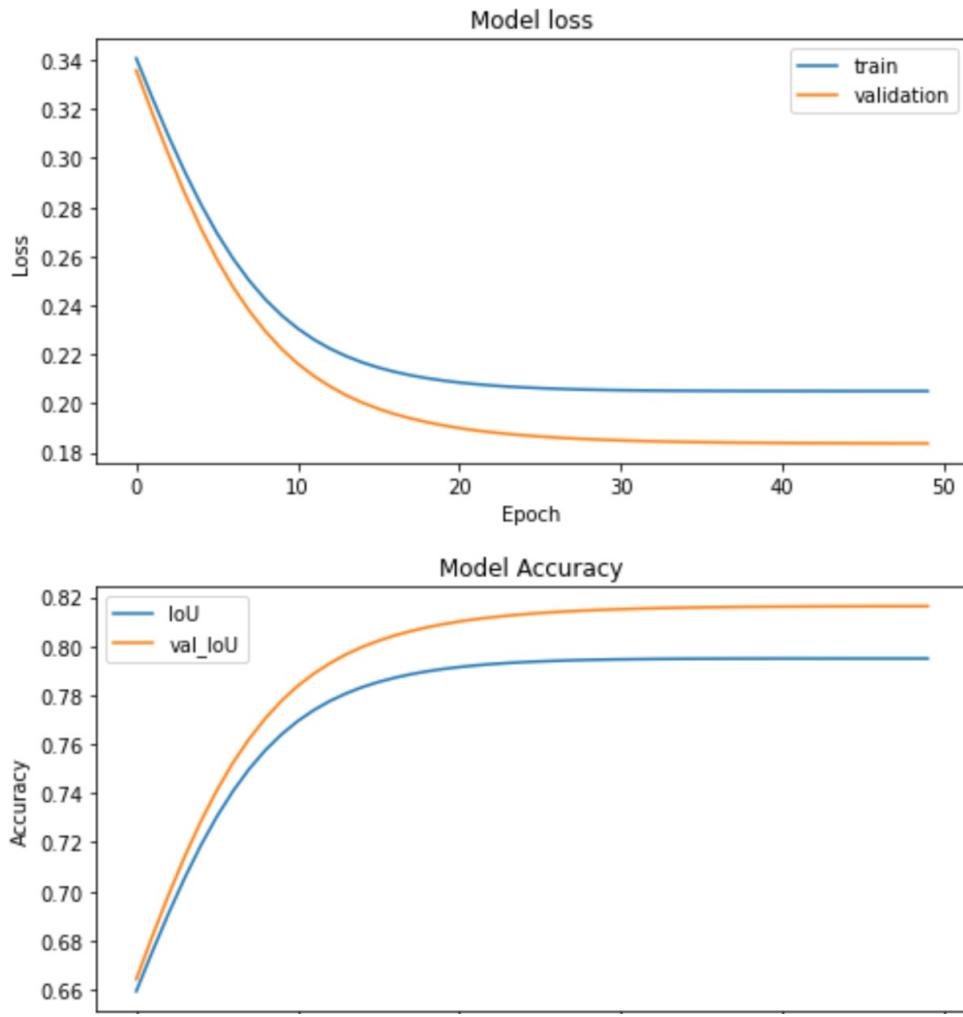


Figure 20: Trains well and smoothly but gets stuck near the end so we need to change LR

Learning rate=0.0001

```
#Learning rate 2/3
model = Sequential()
model.add(vgg)
model.add(Conv2D(filters=512, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2D(filters=1, kernel_size=(1,1),padding = 'same',activation="relu"))
model.add(Conv2DTranspose(filters=1 ,padding='same', kernel_size=(64,64),strides=(32,32) ,activation="sigmoid"))
model.summary()
model.compile(loss=[jaccard_distance],
              optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
              metrics=[IoU])

Model: "sequential_1"
Layer (type)          Output Shape         Param #
vgg16 (Functional)    (None, 8, 8, 512)      14714688
conv2d_3 (Conv2D)     (None, 8, 8, 512)      262656
conv2d_4 (Conv2D)     (None, 8, 8, 512)      262656
conv2d_5 (Conv2D)     (None, 8, 8, 1)       513
conv2d_transpose_1 (Conv2DTr) (None, 256, 256, 1) 4097
=====
Total params: 15,244,610
=====
Epoch 2/50
125/125 [=====>.....] - 17s 137ms/step - loss: 0.1393 - IoU: 0.8607 - val_loss: 0.1087 - val_IoU: 0.8913
Epoch 3/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.1046 - IoU: 0.8954 - val_loss: 0.1001 - val_IoU: 0.8999
Epoch 4/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0999 - IoU: 0.9001 - val_loss: 0.1077 - val_IoU: 0.8923
Epoch 5/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0964 - IoU: 0.9036 - val_loss: 0.0956 - val_IoU: 0.9044
Epoch 6/50
15/125 [==>.....] - ETA: 14s - loss: 0.0977 - IoU: 0.9023
Epoch 00006: loss improved from inf to 0.09765, saving model to best_model.hdf5
125/125 [=====>.....] - 18s 140ms/step - loss: 0.0921 - IoU: 0.9079 - val_loss: 0.1000 - val_IoU: 0.9000
Epoch 7/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0904 - IoU: 0.9096 - val_loss: 0.0972 - val_IoU: 0.9028
Epoch 8/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0881 - IoU: 0.9119 - val_loss: 0.0953 - val_IoU: 0.9047
Epoch 9/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0874 - IoU: 0.9126 - val_loss: 0.0984 - val_IoU: 0.9016
Epoch 10/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0865 - IoU: 0.9135 - val_loss: 0.0919 - val_IoU: 0.9081
Epoch 11/50
38/125 [=====>.....] - ETA: 12s - loss: 0.0831 - IoU: 0.9169
Epoch 00011: loss improved from 0.09765 to 0.08313, saving model to best_model.hdf5
125/125 [=====>.....] - 17s 140ms/step - loss: 0.0843 - IoU: 0.9157 - val_loss: 0.0904 - val_IoU: 0.9096
Epoch 12/50
125/125 [=====>.....] - 17s 137ms/step - loss: 0.0826 - IoU: 0.9174 - val_loss: 0.0877 - val_IoU: 0.9123
Epoch 13/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0809 - IoU: 0.9191 - val_loss: 0.0882 - val_IoU: 0.9118
Epoch 14/50
125/125 [=====>.....] - 17s 137ms/step - loss: 0.0811 - IoU: 0.9189 - val_loss: 0.0938 - val_IoU: 0.9062
Epoch 15/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0792 - IoU: 0.9208 - val_loss: 0.0872 - val_IoU: 0.9128
Epoch 00016: loss improved from 0.08313 to 0.0792, saving model to best_model.hdf5
125/125 [=====>.....] - 18s 140ms/step - loss: 0.0773 - IoU: 0.9227 - val_loss: 0.0872 - val_IoU: 0.9128
Epoch 17/50
125/125 [=====>.....] - 17s 137ms/step - loss: 0.0752 - IoU: 0.9248 - val_loss: 0.0868 - val_IoU: 0.9132
Epoch 18/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0732 - IoU: 0.9268 - val_loss: 0.0848 - val_IoU: 0.9152
Epoch 19/50
125/125 [=====>.....] - 17s 137ms/step - loss: 0.0719 - IoU: 0.9281 - val_loss: 0.0817 - val_IoU: 0.9183
Epoch 20/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0695 - IoU: 0.9305 - val_loss: 0.0849 - val_IoU: 0.9151
Epoch 21/50
60/125 [=====>.....] - ETA: 8s - loss: 0.0663 - IoU: 0.9337
Epoch 00021: loss improved from 0.07512 to 0.06628, saving model to best_model.hdf5
125/125 [=====>.....] - 18s 140ms/step - loss: 0.0677 - IoU: 0.9323 - val_loss: 0.0851 - val_IoU: 0.9149
Epoch 22/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0655 - IoU: 0.9345 - val_loss: 0.0838 - val_IoU: 0.9162
Epoch 23/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0629 - IoU: 0.9371 - val_loss: 0.0813 - val_IoU: 0.9187
Epoch 24/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0607 - IoU: 0.9393 - val_loss: 0.0817 - val_IoU: 0.9183
Epoch 25/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0596 - IoU: 0.9404 - val_loss: 0.0841 - val_IoU: 0.9159
Epoch 26/50
75/125 [=====>.....] - ETA: 6s - loss: 0.0573 - IoU: 0.9427
Epoch 00026: loss improved from 0.06628 to 0.05734, saving model to best_model.hdf5
125/125 [=====>.....] - 17s 140ms/step - loss: 0.0574 - IoU: 0.9426 - val_loss: 0.0783 - val_IoU: 0.9217
Epoch 27/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0559 - IoU: 0.9441 - val_loss: 0.0839 - val_IoU: 0.9161
Epoch 28/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0549 - IoU: 0.9451 - val_loss: 0.0771 - val_IoU: 0.9229
Epoch 29/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0536 - IoU: 0.9464 - val_loss: 0.0764 - val_IoU: 0.9236
Epoch 30/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0523 - IoU: 0.9477 - val_loss: 0.0751 - val_IoU: 0.9249
Epoch 31/50
90/125 [=====>.....] - ETA: 4s - loss: 0.0512 - IoU: 0.9488
Epoch 00031: loss improved from 0.05734 to 0.05124, saving model to best_model.hdf5
125/125 [=====>.....] - 17s 140ms/step - loss: 0.0518 - IoU: 0.9482 - val_loss: 0.0767 - val_IoU: 0.9233
Epoch 32/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0518 - IoU: 0.9482 - val_loss: 0.0754 - val_IoU: 0.9246
Epoch 33/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0501 - IoU: 0.9499 - val_loss: 0.0754 - val_IoU: 0.9246
Epoch 34/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0496 - IoU: 0.9504 - val_loss: 0.0763 - val_IoU: 0.9237
Epoch 35/50
125/125 [=====>.....] - 17s 136ms/step - loss: 0.0486 - IoU: 0.9514 - val_loss: 0.0765 - val_IoU: 0.9235
Restoring model weights from the end of the best epoch.
Epoch 00035: early stopping
```

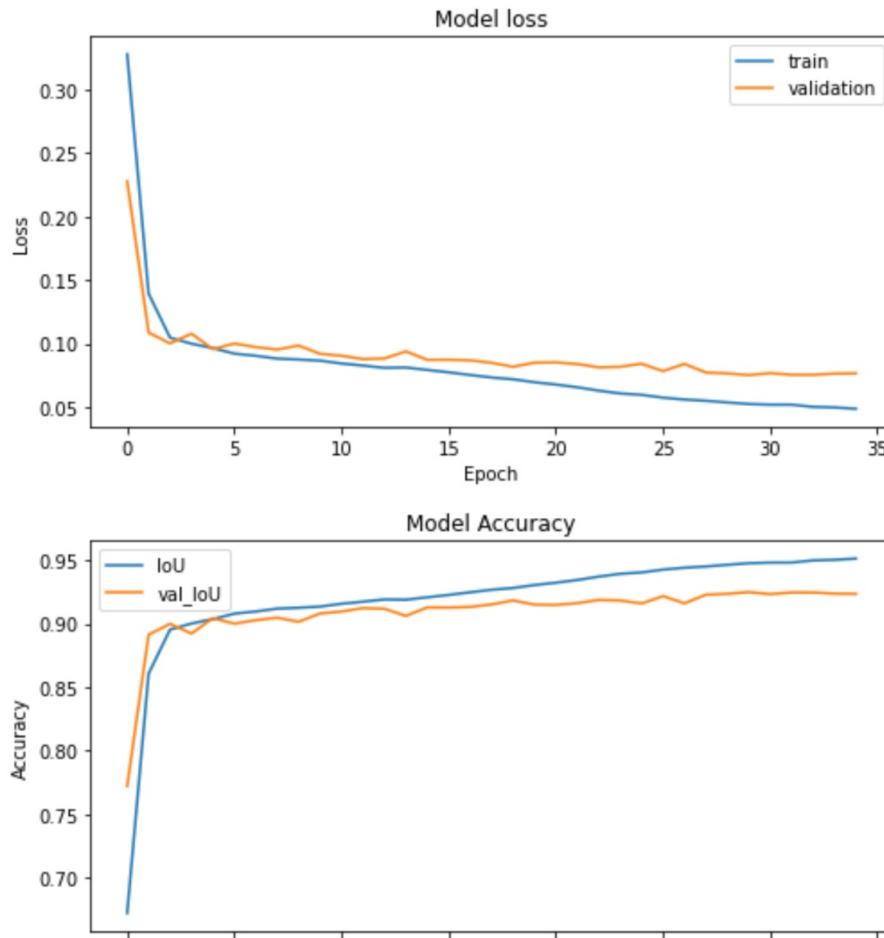
Learning rate=0.0001

Figure 21: Trains well and smoothly but gets stuck near the end

```
score = model.evaluate(x_test, y_test.astype(np.float32), verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], score[1]*100))
```

IoU: 89.17%