Irfan Ahmed

ICP 11

Due Date: 11/21/2020

Parameters changed and reasoning.

Optimizer was changed to 'Adagrad', validation spilt was increased to 0.25, the batch size was changed to 32, and epochs to 25. Also, added additional layers to model and compared to the unchanged model.

I ended up changing the optimizer along with adding additional layers to see if the val_dice_accu would improve or not. First, I ran the model unchanged to see what the accuracy would be. Then after adding additional sections the model was compiled and executed. The accuracy was best for the model that was provided by the instructor, and when I added additional layers and changed the optimizer and validation spilt the val_dice_acc was 0.7207 compared to 0.8683. The changes made were for comparative purpose to better under how to best tune any model. Few things to note is I was unsure if my model setup was even correct and that must be taken into consideration when looking at the results. Other thing to note is that optimizer 'adam' is the best choice to use when validating the model. Adding additional layers saw zero improvements in the overall val_dice_acc.

a. What you learned in the ICP

The goal of this ICP was to understand the U-Net architecture. The architecture is composed three sections. The contraction, bottleneck, and expansion section/layers. The contraction layer has many contraction blocks, where each block takes an input 3X3 convolution layer followed by 2X2 max pooling. The kernels double after each block, and the bottom layer mediates between the contraction layer and expansion layer. From this contraction and expansion layer we can see a semi u-shape, where the name is derived from.

b. ICP description what was the task you were performing

- Imported the required libraries/packages
- Installed Kaggle and used my token to get obtain access to the dataset
- Extracted the dataset from data-science-bowl-2018 and obtained the train data
- Initialized my img width, height and channels and then trained the x and y
- Two models were built and the first was unchanged from what was givein in class. The second model added more sections to compare the results.
- Compiled my model and looked at the results
- Used visualizing to look at the results.

c. Challenges you faced

One thing that was difficult for me was how to add more layers/sections. Still bit confused on some of the topic but I ended up just adding 9 additional layers to keep the pattern consistent and was able to run the model.

d. **Screen shots that shows the successful execution of each required step of your code**

Make 4 changes(for example changing number of layers, changing hyper parameters, addin
architecture given in the source code and train it on the same (or different) data set. Justify
report for each of the change you made. Visualize model performance and compare any dif

model performance.

Things I changed.

- The epochs was changed to 25
- The Max pooling layer was added
- Covolution layer was added c6
- optimizer Adagrad was applied to the comipler

Importing the required libraries and packages

```
[1]  import tensorflow as tf
     import os
     import sys
     import cv2
     import random
     import numpy as np
     from tqdm import tqdm
     from itertools import chain
     from skimage.io import imread, imshow
     from keras import backend as K
     import matplotlib.pyplot as plt
```

Installing kaggle pacakge. This will allow me to download the needed data to run the model

```
[2]  !pip install -q kaggle
```

Uploading my token

2)

Choose Files kaggle.json

- **kaggle.json**(application/json) - 63 bytes, last modified: 11/21/2020 - 100% done

```
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"irfanac","key":"f2743e6080e6edcffb6a841299eb3dce"
```

```
!mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle
```

[6]
```
! chmod 600 ~/.kaggle/kaggle.json
```

[7]
```
! kaggle datasets list
```

```
Warning: Looks like you're using an outdated API Version, please consider updati
ref                                                          title
-----------------------------------------------------------  ------------------------
unanimad/us-election-2020                                    US Election 2020
antgoldbloom/covid19-data-from-john-hopkins-university      COVID-19 data from John
manchunhui/us-election-2020-tweets                           US Election 2020 Tweets
headsortails/us-election-2020-presidential-debates           US Election 2020 - Presi
etsc9287/2020-general-election-polls                         Election, COVID, and Dem
radustoicescu/2020-united-states-presidential-election      2020 United States presi
shivamb/netflix-shows                                        Netflix Movies and TV Sh
terenceshin/covid19s-impact-on-airport-traffic               COVID-19's Impact on Air
sootersaalu/amazon-top-50-bestselling-books-2009-2019        Amazon Top 50 Bestsellin
nehaprabhavalkar/indian-food-101                             Indian Food 101
karangadiya/fifa19                                           FIFA 19 complete player
heeraldedhia/groceries-dataset                               Groceries dataset
andrewmvd/trip-advisor-hotel-reviews                         Trip Advisor Hotel Revie
docstein/brics-world-bank-indicators                         BRICS World Bank Indicat
omarhanyy/500-greatest-songs-of-all-time                     500 Greatest Songs of Al
google/tinyquickdraw                                         QuickDraw Sketches
datasnaek/youtube-new                                        Trending YouTube Video S
uciml/mushroom-classification                                Mushroom Classification
anikannal/solar-power-generation-data                        Solar Power Generation D
zynicide/wine-reviews                                        Wine Reviews
```

```
!pip install --upgrade kaggle
```

```
Requirement already up-to-date: kaggle in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied, skipping upgrade: python-dateutil in /usr/local/l
Requirement already satisfied, skipping upgrade: certifi in /usr/local/lib/pytho
Requirement already satisfied, skipping upgrade: urllib3 in /usr/local/lib/pytho
Requirement already satisfied, skipping upgrade: slugify in /usr/local/lib/pytho
Requirement already satisfied, skipping upgrade: requests in /usr/local/lib/pyth
Requirement already satisfied, skipping upgrade: python-slugify in /usr/local/li
Requirement already satisfied, skipping upgrade: six>=1.10 in /usr/local/lib/pyt
Requirement already satisfied, skipping upgrade: tqdm in /usr/local/lib/python3.
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in /usr/local
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in /usr/local/lib/
Requirement already satisfied, skipping upgrade: text-unidecode>=1.3 in /usr/loc
```

```
[9] !kaggle competitions download -c data-science-bowl-2018
```

```
Warning: Looks like you're using an outdated API Version, please consider updati
Downloading stage2_test_final.zip to /content
100% 276M/276M [00:03<00:00, 52.8MB/s]
100% 276M/276M [00:03<00:00, 78.9MB/s]
Downloading stage1_test.zip to /content
 55% 5.00M/9.10M [00:00<00:00, 7.25MB/s]
100% 9.10M/9.10M [00:00<00:00, 12.5MB/s]
Downloading stage1_sample_submission.csv.zip to /content
  0% 0.00/2.62k [00:00<?, ?B/s]
100% 2.62k/2.62k [00:00<00:00, 2.41MB/s]
Downloading stage2_sample_submission_final.csv.zip to /content
  0% 0.00/112k [00:00<?, ?B/s]
100% 112k/112k [00:00<00:00, 115MB/s]
Downloading stage1_solution.csv.zip to /content
  0% 0.00/386k [00:00<?, ?B/s]
100% 386k/386k [00:00<00:00, 55.0MB/s]
Downloading stage1_train_labels.csv.zip to /content
  0% 0.00/2.67M [00:00<?, ?B/s]
100% 2.67M/2.67M [00:00<00:00, 88.1MB/s]
Downloading stage1_train.zip to /content
 94% 74.0M/79.1M [00:02<00:00, 20.7MB/s]
100% 79.1M/79.1M [00:02<00:00, 33.6MB/s]
```

```
[10] ! mkdir train
```

```
! unzip /content/stage1_train.zip -d train
```

```
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
  inflating: train/d6a880b1f6056f3086679de5c810e7af87cdf3bbbd0533a83e3681817fce
   creating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
   creating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
  inflating: train/d751ccb64fa767a65a966061218438bd1860695d96bbef11fdb2f0d3b8de
   creating: train/d7d12a2acc47a94961aeb56fd56e8a0873016af75f5dd10915de9db8af8e
   creating: train/d7d12a2acc47a94961aeb56fd56e8a0873016af75f5dd10915de9db8af8e
  inflating: train/d7d12a2acc47a94961aeb56fd56e8a0873016af75f5dd10915de9db8af8e
  inflating: train/d7d12a2acc47a94961aeb56fd56e8a0873016af75f5dd10915de9db8af8e
  inflating: train/d7d12a2acc47a94961aeb56fd56e8a0873016af75f5dd10915de9db8af8e
  inflating: train/d7d12a2acc47a94961aeb56fd56e8a0873016af75f5dd10915de9db8af8e
```

Setting up my model

```
IMG_WIDTH = 128
IMG_HEIGHT = 128
IMG_CHANNELS = 3

DATA_PATH = '/content/train/'


seed = 42
random.seed = seed
np.random.seed = seed

image_ids = next(os.walk(DATA_PATH))[1]

X = np.zeros((len(image_ids), IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS), dtype=np.ui
Y = np.zeros((len(image_ids), IMG_HEIGHT, IMG_WIDTH, 1), dtype=np.bool)

for n, id_ in tqdm(enumerate(image_ids), total=len(image_ids)):
    path = DATA_PATH + id_
    img = imread(path + '/images/' + id_ + '.png')[:,:,:IMG_CHANNELS]
    img = cv2.resize(img, (IMG_HEIGHT, IMG_WIDTH), )
    X[n] = img
    mask = np.zeros((IMG_HEIGHT, IMG_WIDTH, 1), dtype=np.bool)
    for mask_file in next(os.walk(path + '/masks/'))[2]:
        mask_ = imread(path + '/masks/' + mask_file)
        mask_ = np.expand_dims(cv2.resize(mask_, (IMG_HEIGHT, IMG_WIDTH), inter
        mask = np.maximum(mask, mask_)
    Y[n] = mask

x_train=X

y_train=Y
```

```
100%|██████████| 670/670 [01:04<00:00, 10.36it/s]
```

This is the first run without

```python
# Build U-Net model
inputs = tf.keras.layers.Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
s = tf.keras.layers.Lambda(lambda x: x / 255)(inputs)

c1 = tf.keras.layers.Conv2D(16, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(s)
c1 = tf.keras.layers.Dropout(0.1)(c1)
c1 = tf.keras.layers.Conv2D(16, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c1)
p1 = tf.keras.layers.MaxPooling2D((2, 2))(c1)

c2 = tf.keras.layers.Conv2D(32, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(p1)
c2 = tf.keras.layers.Dropout(0.1)(c2)
c2 = tf.keras.layers.Conv2D(32, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c2)
p2 = tf.keras.layers.MaxPooling2D((2, 2))(c2)

c3 = tf.keras.layers.Conv2D(64, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(p2)
c3 = tf.keras.layers.Dropout(0.2)(c3)
c3 = tf.keras.layers.Conv2D(64, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c3)
p3 = tf.keras.layers.MaxPooling2D((2, 2))(c3)

c4 = tf.keras.layers.Conv2D(128, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(p3)
c4 = tf.keras.layers.Dropout(0.2)(c4)
c4 = tf.keras.layers.Conv2D(128, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c4)
p4 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(c4)

c5 = tf.keras.layers.Conv2D(256, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(p4)
c5 = tf.keras.layers.Dropout(0.3)(c5)
c5 = tf.keras.layers.Conv2D(256, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c5)

u6 = tf.keras.layers.Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(c5)
u6 = tf.keras.layers.concatenate([u6, c4])
c6 = tf.keras.layers.Conv2D(128, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(u6)
c6 = tf.keras.layers.Dropout(0.2)(c6)
c6 = tf.keras.layers.Conv2D(128, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c6)

u7 = tf.keras.layers.Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(c6)
u7 = tf.keras.layers.concatenate([u7, c3])
c7 = tf.keras.layers.Conv2D(64, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(u7)
c7 = tf.keras.layers.Dropout(0.2)(c7)
c7 = tf.keras.layers.Conv2D(64, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c7)

u8 = tf.keras.layers.Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same')(c7)
u8 = tf.keras.layers.concatenate([u8, c2])
c8 = tf.keras.layers.Conv2D(32, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(u8)
c8 = tf.keras.layers.Dropout(0.1)(c8)
c8 = tf.keras.layers.Conv2D(32, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c8)

u9 = tf.keras.layers.Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same')(c8)
u9 = tf.keras.layers.concatenate([u9, c1], axis=3)
c9 = tf.keras.layers.Conv2D(16, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(u9)
c9 = tf.keras.layers.Dropout(0.1)(c9)
c9 = tf.keras.layers.Conv2D(16, (3, 3), activation=tf.keras.activations.elu, kernel_initializer='he_normal',
                            padding='same')(c9)

outputs = tf.keras.layers.Conv2D(1, (1, 1), activation='sigmoid')(c9)
```

```
[15] def dice_acc(y_true, y_pred):
         smooth = 1.
         y_true_f = K.flatten(y_true)
         y_pred_f = K.flatten(y_pred)
         intersection = K.sum(y_true_f * y_pred_f)
         return (2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + smooth)
```

```
[17] model = tf.keras.Model(inputs=[inputs], outputs=[outputs])
     model.compile(optimizer='adam', loss='binary_crossentropy', metrics=[dice_acc])
     model.summary()
```

```
Model: "functional_3"
______________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==============================================================================================
input_2 (InputLayer)            [(None, 128, 128, 3) 0
______________________________________________________________________________________________
lambda_1 (Lambda)               (None, 128, 128, 3)  0           input_2[0][0]
______________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, 128, 128, 16) 448         lambda_1[0][0]
______________________________________________________________________________________________
dropout_11 (Dropout)            (None, 128, 128, 16) 0           conv2d_23[0][0]
______________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, 128, 128, 16) 2320        dropout_11[0][0]
______________________________________________________________________________________________
max_pooling2d_5 (MaxPooling2D)  (None, 64, 64, 16)   0           conv2d_24[0][0]
______________________________________________________________________________________________
conv2d_25 (Conv2D)              (None, 64, 64, 32)   4640        max_pooling2d_5[0][0]
______________________________________________________________________________________________
dropout_12 (Dropout)            (None, 64, 64, 32)   0           conv2d_25[0][0]
______________________________________________________________________________________________
conv2d_26 (Conv2D)              (None, 64, 64, 32)   9248        dropout_12[0][0]
______________________________________________________________________________________________
max_pooling2d_6 (MaxPooling2D)  (None, 32, 32, 32)   0           conv2d_26[0][0]
______________________________________________________________________________________________
conv2d_27 (Conv2D)              (None, 32, 32, 64)   18496       max_pooling2d_6[0][0]
______________________________________________________________________________________________
dropout_13 (Dropout)            (None, 32, 32, 64)   0           conv2d_27[0][0]
______________________________________________________________________________________________
conv2d_28 (Conv2D)              (None, 32, 32, 64)   36928       dropout_13[0][0]
______________________________________________________________________________________________
max_pooling2d_7 (MaxPooling2D)  (None, 16, 16, 64)   0           conv2d_28[0][0]
______________________________________________________________________________________________
conv2d_29 (Conv2D)              (None, 16, 16, 128)  73856       max_pooling2d_7[0][0]
______________________________________________________________________________________________
dropout_14 (Dropout)            (None, 16, 16, 128)  0           conv2d_29[0][0]
______________________________________________________________________________________________
conv2d_30 (Conv2D)              (None, 16, 16, 128)  147584      dropout_14[0][0]
```

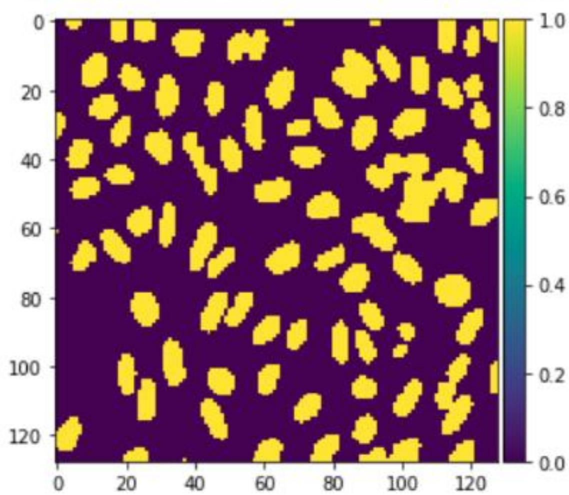
```
                                                                       conv2d_26[0][0]
__________________________________________________________________________________________________
conv2d_37 (Conv2D)              (None, 64, 64, 32)   18464       concatenate_7[0][0]
__________________________________________________________________________________________________
dropout_18 (Dropout)            (None, 64, 64, 32)   0           conv2d_37[0][0]
__________________________________________________________________________________________________
conv2d_38 (Conv2D)              (None, 64, 64, 32)   9248        dropout_18[0][0]
__________________________________________________________________________________________________
conv2d_transpose_8 (Conv2DTrans (None, 128, 128, 16) 2064        conv2d_38[0][0]
__________________________________________________________________________________________________
concatenate_8 (Concatenate)     (None, 128, 128, 32) 0           conv2d_transpose_8[0
                                                                 conv2d_24[0][0]
__________________________________________________________________________________________________
conv2d_39 (Conv2D)              (None, 128, 128, 16) 4624        concatenate_8[0][0]
__________________________________________________________________________________________________
dropout_19 (Dropout)            (None, 128, 128, 16) 0           conv2d_39[0][0]
__________________________________________________________________________________________________
conv2d_40 (Conv2D)              (None, 128, 128, 16) 2320        dropout_19[0][0]
__________________________________________________________________________________________________
conv2d_41 (Conv2D)              (None, 128, 128, 1)  17          conv2d_40[0][0]
==================================================================================================
Total params: 1,941,105
Trainable params: 1,941,105
Non-trainable params: 0
__________________________________________________________________________________________________
```

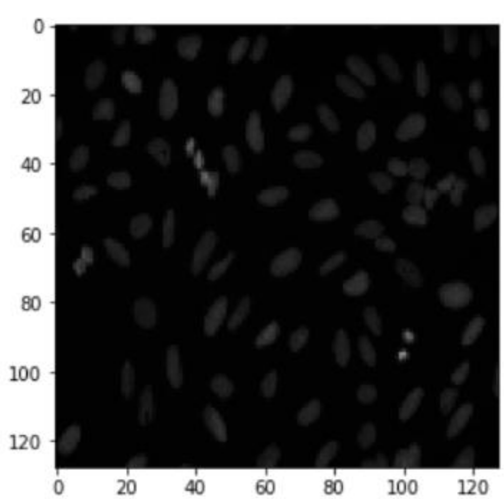
```
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)

# Create checkpoint callback
cp_callback = tf.keras.callbacks.ModelCheckpoint(checkpoint_path,
                                                 save_weights_only=True,
                                                 verbose=1)

callbacks = [
  tf.keras.callbacks.EarlyStopping(patience=50, monitor='val_loss', verbose=1), #pati
  tf.keras.callbacks.TensorBoard(log_dir='./logs'),
  cp_callback
]

results_1 = model.fit(x_train, y_train, validation_split=0.01, batch_size=16, epochs=
#results_2 = model.fit(x_train, y_train, validation_split=0.25, batch_size=32, epochs
```

```
41/42 [============================>.] - ETA: 0s - loss: 0.0827 - dice_acc: 0.8550
Epoch 00014: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0828 - dice_acc: 0.8549
Epoch 15/25
41/42 [============================>.] - ETA: 0s - loss: 0.0793 - dice_acc: 0.8610
Epoch 00015: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0792 - dice_acc: 0.8600
Epoch 16/25
41/42 [============================>.] - ETA: 0s - loss: 0.0769 - dice_acc: 0.8647
Epoch 00016: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0774 - dice_acc: 0.8645
Epoch 17/25
41/42 [============================>.] - ETA: 0s - loss: 0.0772 - dice_acc: 0.8653
Epoch 00017: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0771 - dice_acc: 0.8652
Epoch 18/25
41/42 [============================>.] - ETA: 0s - loss: 0.0767 - dice_acc: 0.8678
Epoch 00018: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0770 - dice_acc: 0.8673
Epoch 19/25
41/42 [============================>.] - ETA: 0s - loss: 0.0765 - dice_acc: 0.8650
Epoch 00019: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0763 - dice_acc: 0.8657
Epoch 20/25
41/42 [============================>.] - ETA: 0s - loss: 0.0758 - dice_acc: 0.8677
Epoch 00020: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 45ms/step - loss: 0.0755 - dice_acc: 0.8688
Epoch 21/25
41/42 [============================>.] - ETA: 0s - loss: 0.0743 - dice_acc: 0.8718
Epoch 00021: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 44ms/step - loss: 0.0742 - dice_acc: 0.8723
Epoch 22/25
41/42 [============================>.] - ETA: 0s - loss: 0.0745 - dice_acc: 0.8698
Epoch 00022: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 45ms/step - loss: 0.0745 - dice_acc: 0.8698
Epoch 23/25
41/42 [============================>.] - ETA: 0s - loss: 0.0752 - dice_acc: 0.8690
Epoch 00023: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 46ms/step - loss: 0.0752 - dice_acc: 0.8689
Epoch 24/25
42/42 [==============================] - ETA: 0s - loss: 0.0742 - dice_acc: 0.8692
Epoch 00024: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 45ms/step - loss: 0.0742 - dice_acc: 0.8692
Epoch 25/25
41/42 [============================>.] - ETA: 0s - loss: 0.0762 - dice_acc: 0.8685
Epoch 00025: saving model to training_1/cp.ckpt
42/42 [==============================] - 2s 45ms/step - loss: 0.0761 - dice_acc: 0.8683
```

```
idx = random.randint(0, len(x_train))
x=np.array(x_train[idx])
x=np.expand_dims(x, axis=0)
predict = model.predict(x, verbose=1)

predict = (predict > 0.5).astype(np.uint8)

imshow(np.squeeze(predict[0]))
plt.show()

imshow(x_train[idx])

plt.show()
```

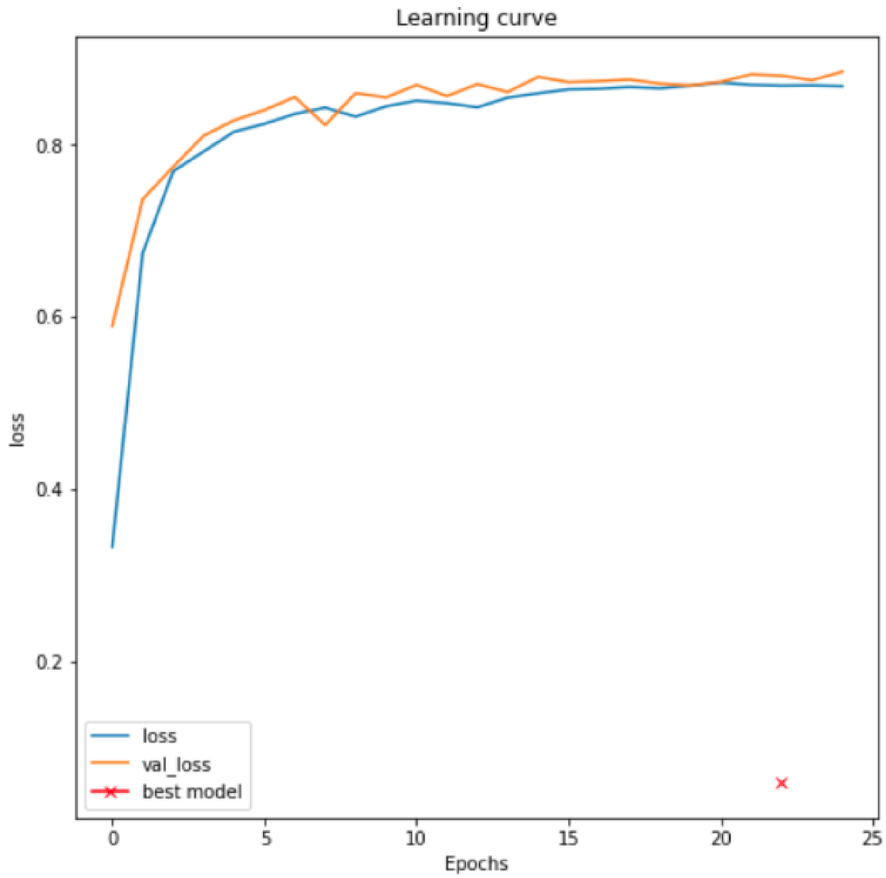
```
1/1 [==============================] - 0s 3ms/step
/usr/local/lib/python3.6/dist-packages/skimage/io/_plugins/matplotlib_plugin.py:150: UserWarni
  lo, hi, cmap = _get_display_range(image)
```





```
plt.figure(figsize=(8, 8))
plt.title("Learning curve")
plt.plot(results_1.history["dice_acc"], label="loss")
plt.plot(results_1.history["val_dice_acc"], label="val_loss")
plt.plot( np.argmin(results_1.history["val_loss"]), np.min(results_1.history["val_loss"]), marke
plt.xlabel("Epochs")
plt.ylabel("loss")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7ff142bb7f28>
```



The images below are for result 2 where changes were made, and the results compared using visualization.

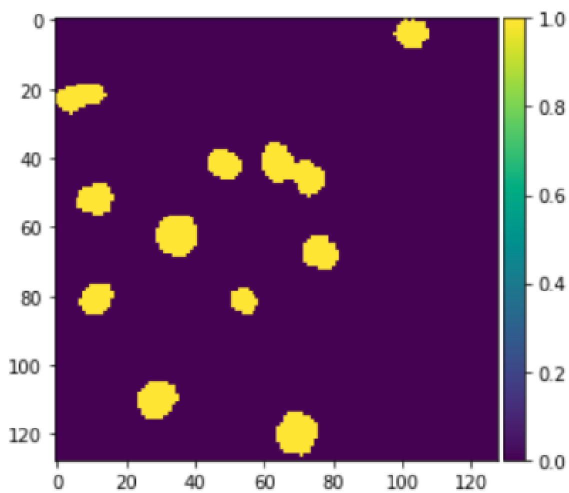
```python
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)

# Create checkpoint callback
cp_callback = tf.keras.callbacks.ModelCheckpoint(checkpoint_path,
                                                 save_weights_only=True,
                                                 verbose=1)


callbacks = [
  tf.keras.callbacks.EarlyStopping(patience=50, monitor='val_loss', verbose=1), #patience was ch
  tf.keras.callbacks.TensorBoard(log_dir='./logs'),
  cp_callback
]

#results_1 = model.fit(x_train, y_train, validation_split=0.01, batch_size=16, epochs=25,callbac
results_2 = model.fit(x_train, y_train, validation_split=0.25, batch_size=32, epochs=25,callback
```

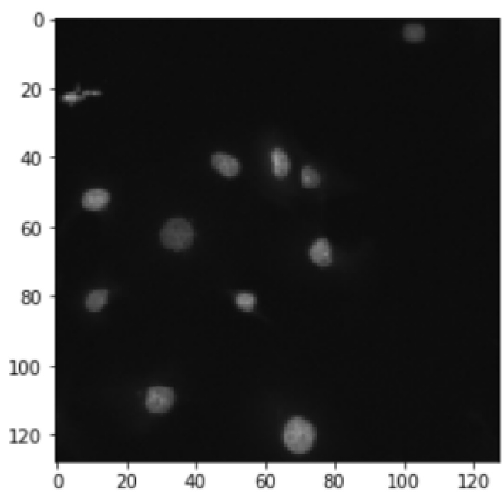
```
16/16 [==============================] - ETA: 0s - loss: 0.1729 - dice_acc: 0.6930
Epoch 00011: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 128ms/step - loss: 0.1729 - dice_acc: 0.6930 - val_l
Epoch 12/25
16/16 [==============================] - ETA: 0s - loss: 0.1720 - dice_acc: 0.6948
Epoch 00012: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 129ms/step - loss: 0.1720 - dice_acc: 0.6948 - val_l
Epoch 13/25
16/16 [==============================] - ETA: 0s - loss: 0.1711 - dice_acc: 0.6953
Epoch 00013: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 129ms/step - loss: 0.1711 - dice_acc: 0.6953 - val_l
Epoch 14/25
16/16 [==============================] - ETA: 0s - loss: 0.1713 - dice_acc: 0.6944
Epoch 00014: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1713 - dice_acc: 0.6944 - val_l
Epoch 15/25
16/16 [==============================] - ETA: 0s - loss: 0.1698 - dice_acc: 0.7000
Epoch 00015: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1698 - dice_acc: 0.7000 - val_l
Epoch 16/25
16/16 [==============================] - ETA: 0s - loss: 0.1698 - dice_acc: 0.6984
Epoch 00016: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 129ms/step - loss: 0.1698 - dice_acc: 0.6984 - val_l
Epoch 17/25
16/16 [==============================] - ETA: 0s - loss: 0.1692 - dice_acc: 0.7009
Epoch 00017: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1692 - dice_acc: 0.7009 - val_l
Epoch 18/25
16/16 [==============================] - ETA: 0s - loss: 0.1686 - dice_acc: 0.7020
Epoch 00018: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1686 - dice_acc: 0.7020 - val_l
Epoch 19/25
16/16 [==============================] - ETA: 0s - loss: 0.1680 - dice_acc: 0.7028
Epoch 00019: saving model to training_1/cp.ckpt
```

```
16/16 [==============================] - 2s 129ms/step - loss: 0.1711 - dice_acc: 0.6953 - val_l
Epoch 14/25
16/16 [==============================] - ETA: 0s - loss: 0.1713 - dice_acc: 0.6944
Epoch 00014: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1713 - dice_acc: 0.6944 - val_l
Epoch 15/25
16/16 [==============================] - ETA: 0s - loss: 0.1698 - dice_acc: 0.7000
Epoch 00015: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1698 - dice_acc: 0.7000 - val_l
Epoch 16/25
16/16 [==============================] - ETA: 0s - loss: 0.1698 - dice_acc: 0.6984
Epoch 00016: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 129ms/step - loss: 0.1698 - dice_acc: 0.6984 - val_l
Epoch 17/25
16/16 [==============================] - ETA: 0s - loss: 0.1692 - dice_acc: 0.7009
Epoch 00017: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1692 - dice_acc: 0.7009 - val_l
Epoch 18/25
16/16 [==============================] - ETA: 0s - loss: 0.1686 - dice_acc: 0.7020
Epoch 00018: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 130ms/step - loss: 0.1686 - dice_acc: 0.7020 - val_l
Epoch 19/25
16/16 [==============================] - ETA: 0s - loss: 0.1680 - dice_acc: 0.7028
Epoch 00019: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 128ms/step - loss: 0.1680 - dice_acc: 0.7028 - val_l
Epoch 20/25
16/16 [==============================] - ETA: 0s - loss: 0.1676 - dice_acc: 0.7025
Epoch 00020: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 128ms/step - loss: 0.1676 - dice_acc: 0.7025 - val_l
Epoch 21/25
16/16 [==============================] - ETA: 0s - loss: 0.1671 - dice_acc: 0.7064
Epoch 00021: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 128ms/step - loss: 0.1671 - dice_acc: 0.7064 - val_l
Epoch 22/25
16/16 [==============================] - ETA: 0s - loss: 0.1662 - dice_acc: 0.7061
Epoch 00022: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 128ms/step - loss: 0.1662 - dice_acc: 0.7061 - val_l
Epoch 23/25
16/16 [==============================] - ETA: 0s - loss: 0.1658 - dice_acc: 0.7082
Epoch 00023: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 128ms/step - loss: 0.1658 - dice_acc: 0.7082 - val_l
Epoch 24/25
16/16 [==============================] - ETA: 0s - loss: 0.1655 - dice_acc: 0.7081
Epoch 00024: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 127ms/step - loss: 0.1655 - dice_acc: 0.7081 - val_l
Epoch 25/25
16/16 [==============================] - ETA: 0s - loss: 0.1642 - dice_acc: 0.7093
Epoch 00025: saving model to training_1/cp.ckpt
16/16 [==============================] - 2s 126ms/step - loss: 0.1642 - dice_acc: 0.7093 - val_l
```

```
idx = random.randint(0, len(x_train))
x=np.array(x_train[idx])
x=np.expand_dims(x, axis=0)
predict = model.predict(x, verbose=1)

predict = (predict > 0.5).astype(np.uint8)

imshow(np.squeeze(predict[0]))
plt.show()

imshow(x_train[idx])

plt.show()
```

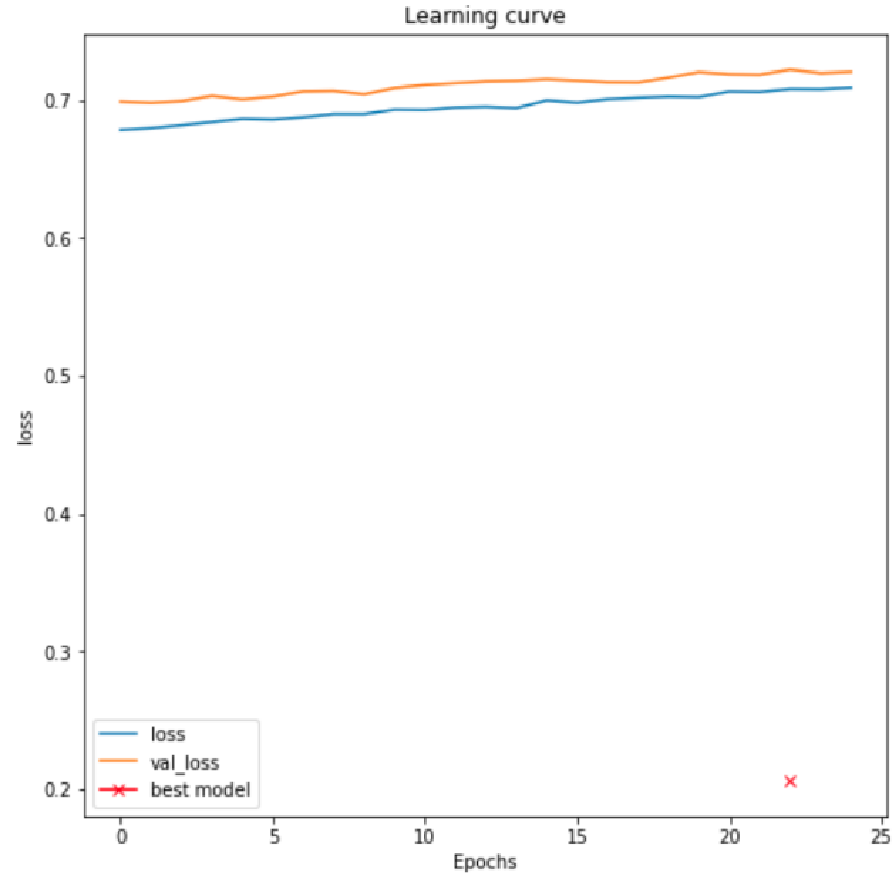
```
1/1 [==============================] - 0s 3ms/step
/usr/local/lib/python3.6/dist-packages/skimage/io/_plugins/matplotlib_plugin.py:150: UserWarning
  lo, hi, cmap = _get_display_range(image)
```





```
plt.figure(figsize=(8, 8))
plt.title("Learning curve")
plt.plot(results_2.history["dice_acc"], label="loss")
plt.plot(results_2.history["val_dice_acc"], label="val_loss")
plt.plot( np.argmin(results_1.history["val_loss"]), np.min(results_1.history["val_loss"]), marke
plt.xlabel("Epochs")
plt.ylabel("loss")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7ff1421b44a8>
```



e. **Output file link if applicable**

https://github.com/UMKC-APL-BigDataAnalytics/icp-11-irfancheemaa

f. **Video link (YouTube or any other publicly available video platform)**

https://umkc.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f8816e88-b9fb-4150-b3e4-ac7c0158c0fd

g. Any inside about the data or the ICP in general

None at the moment.