Attention U-Net : Medical Image Segmentation

Design Project ID-34

Neural Networks and Fuzzy Logic

Implementation of the Research Paper - Attention U-Net : Learning Where to Look for the Pancreas

PROBLEM STATEMENT

- **Existing Problems :**
    - Manual Segmentation has high error rate.
    - If CT is dense, manual segmentation takes time and is a tedious task.
- Due to this **Automated medical image segmentation** has been extensively studied in the image analysis community as :
    - Accurate and reliable solutions.
    - Support decision making through fast and automatic extraction of quantitative measurements
- **Examples** :

Pancreas Segmentation out of abdomen CT and BP Brachial Plexus Nerve out of ultrasound.

DATASET

- **Initial Dataset:**
    - Abdomen CT task to segment pancreas.
    - Images in .tcia zip and dicom format.
    - Masks in .nii format.
        - Tough to process,load, resize etc.
        - Huge python codes required to convert into uint8 , float32 etc.
- **Final Dataset:**
    - Ultrasound nerve images(from Kaggle).
        - 5635 samples of images and their masks..
        - 5580 images in test data.
        - .tif format which is easy to work with.
    - Mask and images in same folder.
        - Rather than segregating, converted into .npy files.
    - Dataset contains two files
        - Xtrain (images).
        - Ytrain (masks)
        - 20% validation data.

DATA PROCESSING

- Original images in dataset consisted of size 420 x 580 pixels.

- Images resized to 96 x 96 pixels.
- Mean Centering & Data Standardization
    - each feature have a similar range
    - gradients don't go out of control
    - only one global learning rate multiplier

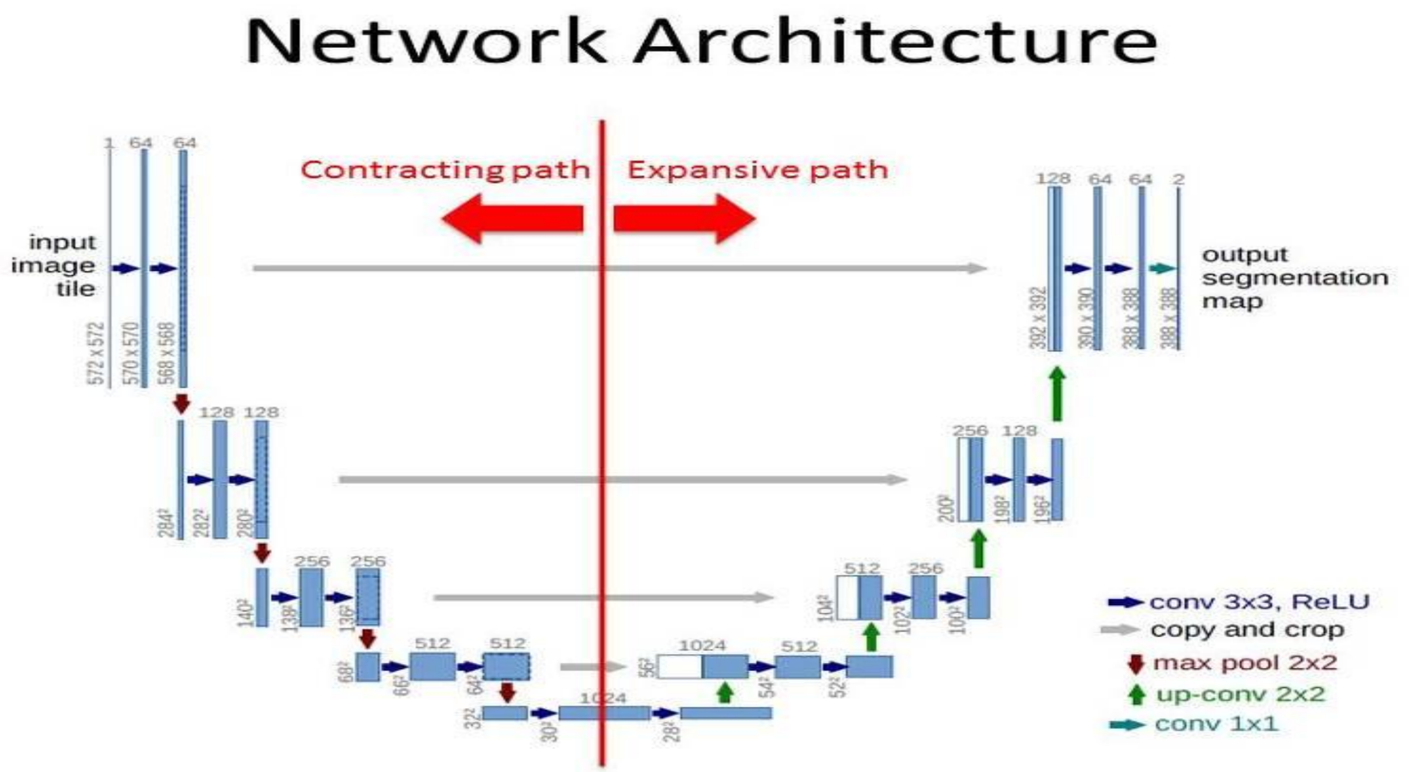
MODEL

- **Attention U-Net**
  - Improves model sensitivity and accuracy by attaching attention gates on top of the standard U-Net.
  - Automatically learns to focus on target structures of varying shapes and sizes.
- **Attention gates**
  - Commonly used in natural image analysis,knowledge graphs and language processing(NLP) for image captioning, machine translation and classification.
  - Improve model sensitivity and accuracy for dense label predictions by suppressing feature activation in irrelevant regions.
- **U-Net**
  - Built upon the Fully Convolutional Network and modified in a way that it yields better segmentation in medical imaging.
  - U-Net architecture is separated in 3 parts:
    - The contracting/downsampling path
    - Bottleneck
    - The expanding/upsampling path

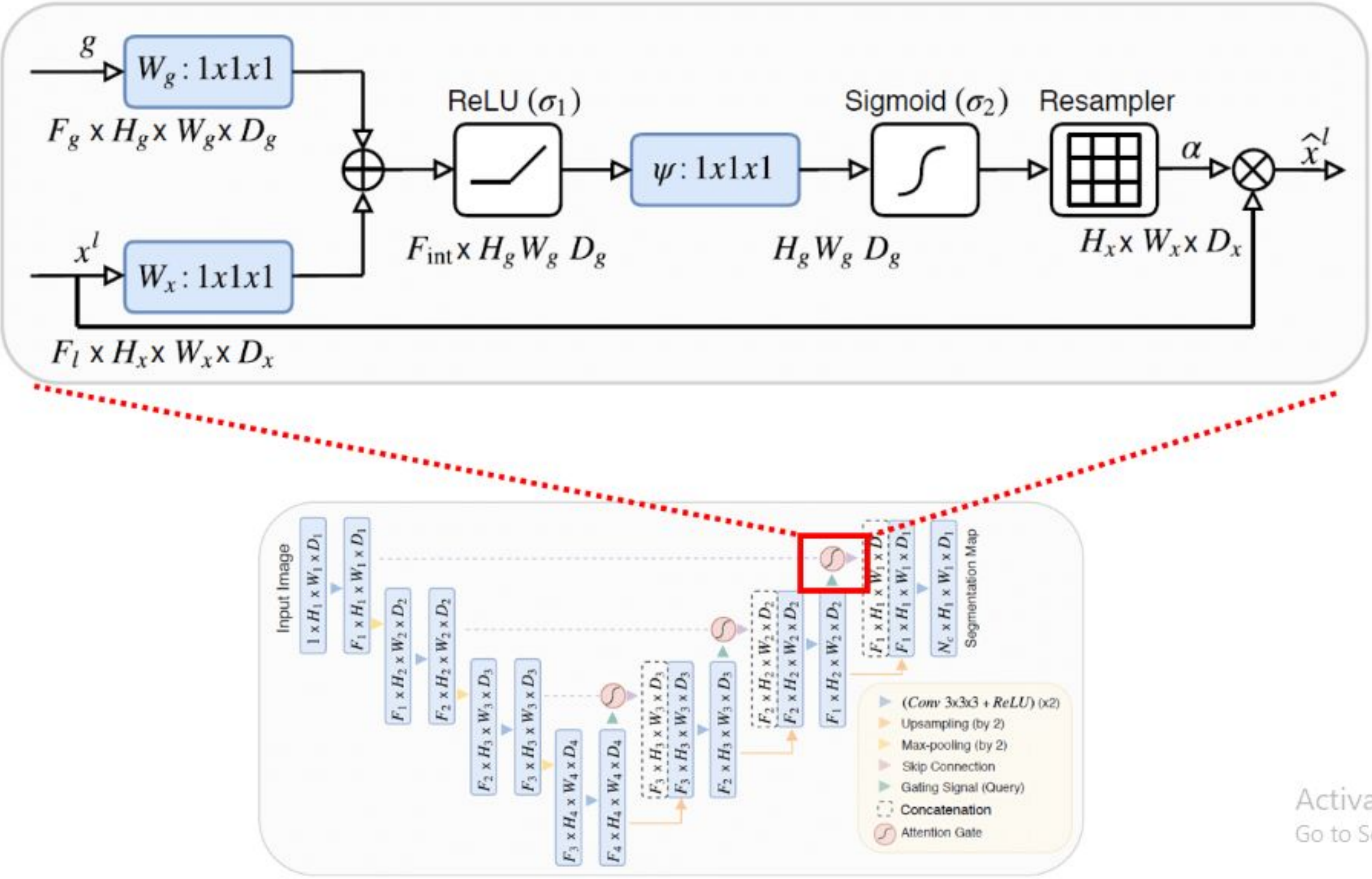
MODEL

- **Adam Optimizer**
  - combining the advantages of two other extensions of stochastic gradient descent:
    - Adaptive Gradient Algorithm.
    - Root Mean Square Propagation.
- **Dice coefficient**
  - Statistical tool which measures the similarity between two sets of data.
  - Equation:
    - $2 * |X| \cap |Y| / (|X| + |Y|)$
  - the range of DSC is between 0 and 1, the larger the better.
- **Dice Loss**
  - Dice Loss= 1-(Dice coeff)
  - Maximize the overlap between two sets.
  - considers the loss information both locally and globally, which is critical for high accuracy.

U-NET Diagram



Attention U-Net



QUANTITATIVE RESULTS

- Measurement and Comparison of performance is done using Dice Coefficient.
- Dice coefficient(Epoch 1)=0.0451
- Dice Coefficient (Epoch 2)=0.2733

KerasAttention-UNet.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
activation_21 (Activation)      (None, 96, 96, 1)    0          conv2d_72[0][0]
[39] ==================================================================================================
Total params: 7,911,460
Trainable params: 7,911,460
Non-trainable params: 0
__________________________________________________________________________________________________

[40] results = model.fit(imgs_train, imgs_mask_train, batch_size=32, epochs=20, verbose=1, shuffle=True,
                         validation_split=0.2,callbacks=[model_checkpoint])

Train on 4508 samples, validate on 1127 samples
Epoch 1/20
4508/4508 [==============================] - 69s 15ms/step - loss: 0.9049 - accuracy: 0.9592 - dice_coef: 0.0451 - val_loss: 0.8909 - val_accuracy: 0.7642 - val_c
Epoch 2/20
4508/4508 [==============================] - 51s 11ms/step - loss: 0.7725 - accuracy: 0.9645 - dice_coef: 0.1527 - val_loss: 0.8532 - val_accuracy: 0.8420 - val_c
Epoch 3/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7465 - accuracy: 0.9682 - dice_coef: 0.1860 - val_loss: 0.8203 - val_accuracy: 0.8861 - val_c
Epoch 4/20
4508/4508 [==============================] - 52s 12ms/step - loss: 0.7371 - accuracy: 0.9697 - dice_coef: 0.1946 - val_loss: 0.7926 - val_accuracy: 0.9170 - val_c
Epoch 5/20
4508/4508 [==============================] - 52s 12ms/step - loss: 0.7264 - accuracy: 0.9711 - dice_coef: 0.2040 - val_loss: 0.7565 - val_accuracy: 0.9430 - val_c
Epoch 6/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7181 - accuracy: 0.9724 - dice_coef: 0.2117 - val_loss: 0.7533 - val_accuracy: 0.9437 - val_c
Epoch 7/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7111 - accuracy: 0.9729 - dice_coef: 0.2186 - val_loss: 0.7320 - val_accuracy: 0.9569 - val_c
Epoch 8/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7042 - accuracy: 0.9739 - dice_coef: 0.2247 - val_loss: 0.7103 - val_accuracy: 0.9653 - val_c
Epoch 9/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6972 - accuracy: 0.9746 - dice_coef: 0.2328 - val_loss: 0.7101 - val_accuracy: 0.9639 - val_c
Epoch 10/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6923 - accuracy: 0.9749 - dice_coef: 0.2386 - val_loss: 0.6992 - val_accuracy: 0.9668 - val_c
Epoch 11/20
```

QUANTITATIVE RESULTS

KerasAttention-UNet.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text RAM Disk Editing

```
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7465 - accuracy: 0.9682 - dice_coef: 0.1860 - val_loss: 0.8203 - val_accuracy: 0.8861 - val_c
Epoch 4/20
4508/4508 [==============================] - 52s 12ms/step - loss: 0.7371 - accuracy: 0.9697 - dice_coef: 0.1946 - val_loss: 0.7926 - val_accuracy: 0.9170 - val_c
Epoch 5/20
4508/4508 [==============================] - 52s 12ms/step - loss: 0.7264 - accuracy: 0.9711 - dice_coef: 0.2040 - val_loss: 0.7565 - val_accuracy: 0.9430 - val_c
Epoch 6/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7181 - accuracy: 0.9724 - dice_coef: 0.2117 - val_loss: 0.7533 - val_accuracy: 0.9437 - val_c
Epoch 7/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7111 - accuracy: 0.9729 - dice_coef: 0.2186 - val_loss: 0.7320 - val_accuracy: 0.9569 - val_c
Epoch 8/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7042 - accuracy: 0.9739 - dice_coef: 0.2247 - val_loss: 0.7103 - val_accuracy: 0.9653 - val_c
Epoch 9/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6972 - accuracy: 0.9746 - dice_coef: 0.2328 - val_loss: 0.7101 - val_accuracy: 0.9639 - val_c
Epoch 10/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6923 - accuracy: 0.9749 - dice_coef: 0.2386 - val_loss: 0.6992 - val_accuracy: 0.9668 - val_c
Epoch 11/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6900 - accuracy: 0.9745 - dice_coef: 0.2412 - val_loss: 0.6958 - val_accuracy: 0.9673 - val_c
Epoch 12/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6844 - accuracy: 0.9755 - dice_coef: 0.2485 - val_loss: 0.7044 - val_accuracy: 0.9619 - val_c
Epoch 13/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6801 - accuracy: 0.9755 - dice_coef: 0.2531 - val_loss: 0.6865 - val_accuracy: 0.9700 - val_c
Epoch 14/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6768 - accuracy: 0.9758 - dice_coef: 0.2570 - val_loss: 0.6892 - val_accuracy: 0.9672 - val_c
Epoch 15/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6753 - accuracy: 0.9755 - dice_coef: 0.2590 - val_loss: 0.6750 - val_accuracy: 0.9723 - val_c
Epoch 16/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6719 - accuracy: 0.9758 - dice_coef: 0.2635 - val_loss: 0.6768 - val_accuracy: 0.9704 - val_c
Epoch 17/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6717 - accuracy: 0.9755 - dice_coef: 0.2640 - val_loss: 0.6736 - val_accuracy: 0.9713 - val_c
Epoch 18/20
1088/4508 [======>.......................] - ETA: 36s - loss: 0.6649 - accuracy: 0.9768 - dice_coef: 0.2715
```

QUANTITATIVE RESULTS

KerasAttention-UNet.ipynb

File Edit View Insert Runtime Tools Help All changes saved

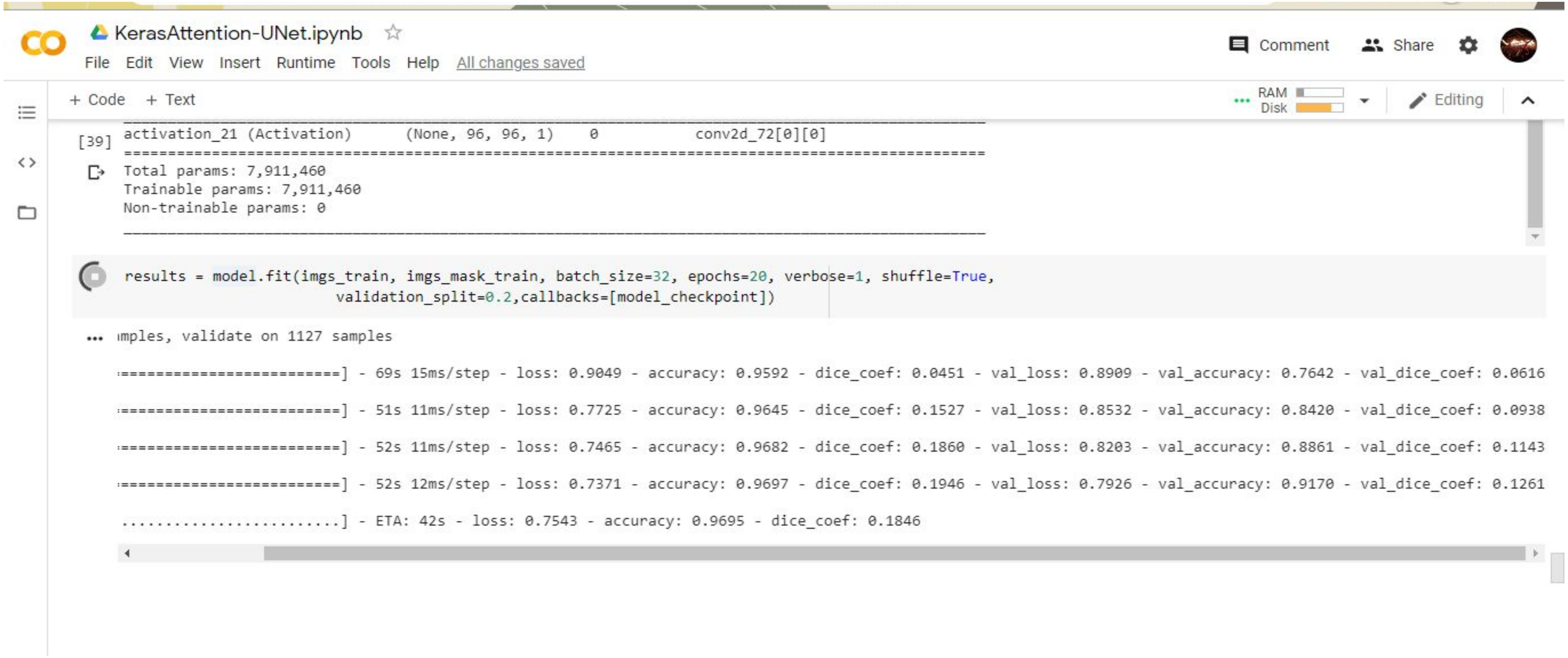
Comment Share

+ Code + Text

RAM Disk Editing

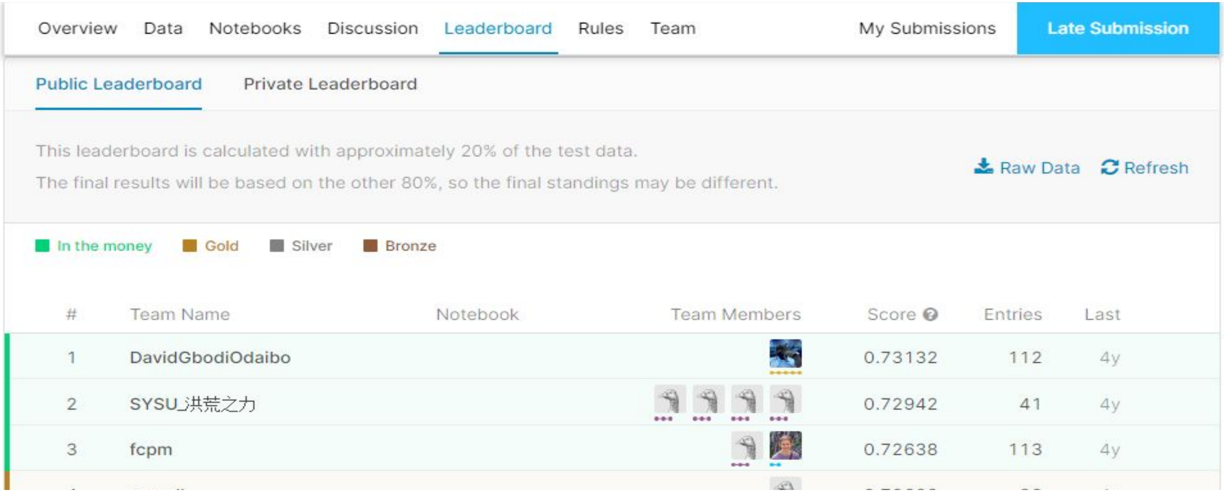
```
[40] 4508/4508 [==============================] - 52s 12ms/step - loss: 0.7264 - accuracy: 0.9711 - dice_coef: 0.2040 - val_loss: 0.7565 - val_accuracy: 0.9430 - val_c
Epoch 6/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7181 - accuracy: 0.9724 - dice_coef: 0.2117 - val_loss: 0.7533 - val_accuracy: 0.9437 - val_c
Epoch 7/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7111 - accuracy: 0.9729 - dice_coef: 0.2186 - val_loss: 0.7320 - val_accuracy: 0.9569 - val_c
Epoch 8/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.7042 - accuracy: 0.9739 - dice_coef: 0.2247 - val_loss: 0.7103 - val_accuracy: 0.9653 - val_c
Epoch 9/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6972 - accuracy: 0.9746 - dice_coef: 0.2328 - val_loss: 0.7101 - val_accuracy: 0.9639 - val_c
Epoch 10/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6923 - accuracy: 0.9749 - dice_coef: 0.2386 - val_loss: 0.6992 - val_accuracy: 0.9668 - val_c
Epoch 11/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6900 - accuracy: 0.9745 - dice_coef: 0.2412 - val_loss: 0.6958 - val_accuracy: 0.9673 - val_c
Epoch 12/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6844 - accuracy: 0.9755 - dice_coef: 0.2485 - val_loss: 0.7044 - val_accuracy: 0.9619 - val_c
Epoch 13/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6801 - accuracy: 0.9755 - dice_coef: 0.2531 - val_loss: 0.6865 - val_accuracy: 0.9700 - val_c
Epoch 14/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6768 - accuracy: 0.9758 - dice_coef: 0.2570 - val_loss: 0.6892 - val_accuracy: 0.9672 - val_c
Epoch 15/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6753 - accuracy: 0.9755 - dice_coef: 0.2590 - val_loss: 0.6750 - val_accuracy: 0.9723 - val_c
Epoch 16/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6719 - accuracy: 0.9758 - dice_coef: 0.2635 - val_loss: 0.6768 - val_accuracy: 0.9704 - val_c
Epoch 17/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6717 - accuracy: 0.9755 - dice_coef: 0.2640 - val_loss: 0.6736 - val_accuracy: 0.9713 - val_c
Epoch 18/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6670 - accuracy: 0.9761 - dice_coef: 0.2700 - val_loss: 0.6702 - val_accuracy: 0.9730 - val_c
Epoch 19/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6656 - accuracy: 0.9763 - dice_coef: 0.2714 - val_loss: 0.6707 - val_accuracy: 0.9711 - val_c
Epoch 20/20
4508/4508 [==============================] - 52s 11ms/step - loss: 0.6642 - accuracy: 0.9762 - dice_coef: 0.2733 - val_loss: 0.6720 - val_accuracy: 0.9693 - val_c
```

QUANTITATIVE RESULTS

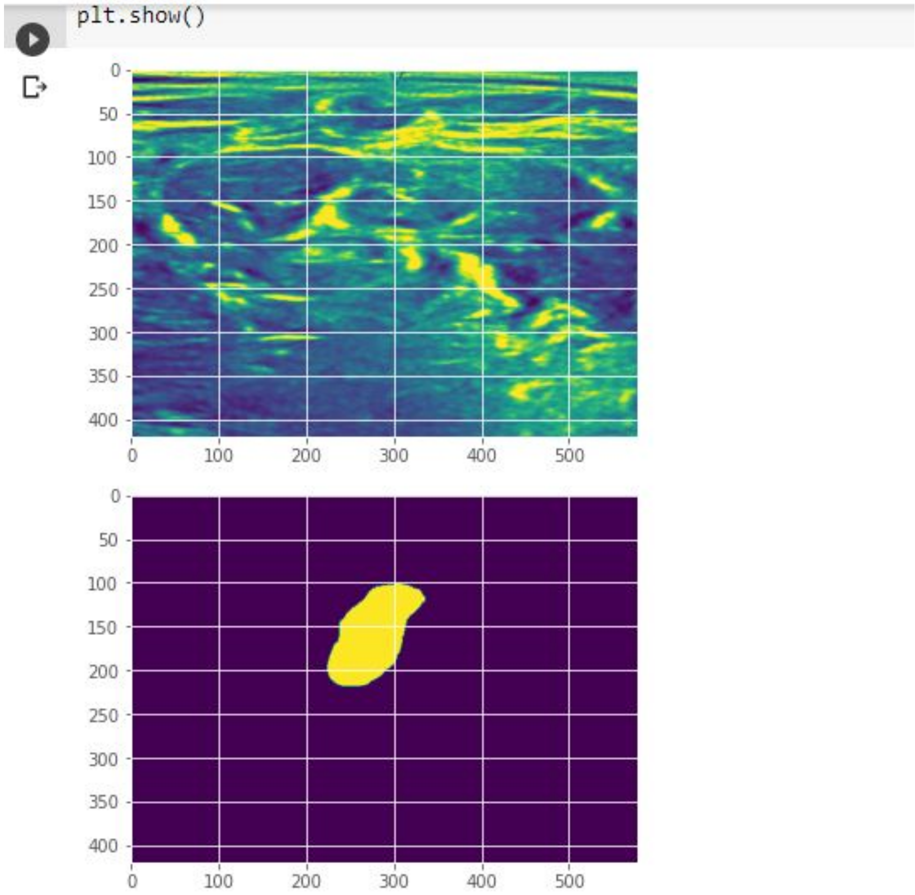


QUANTITATIVE RESULTS

- We compared our results with kaggle competition -top 3 entries.



QUALITATIVE RESULTS



- Kaggle Dataset Entry
- Training Sample: X_train and corresponding mask Y_train

QUALITATIVE RESULTS

- Sample Predictions: