Brain Tumor Classification

Arefa Parwary, Riyad Bin Rafiq, Shoham Weiss

Contents

- Dataset
- Transfer learning
- UNET and Saliency map
- Custom CNN
- Limitation and future work

Dataset

- Brain Tumor Image Dataset (MRI Images) from Kaggle.
- Gathered by Southern Medical University, Guangzhou, China.
- Contains 3064 T1-weighted contrast-enhanced MRI images from 233 patients:
  - Meningioma [708 slices]
  - Glioma [1426 slices]
  - Pituitary tumor [930 slices]

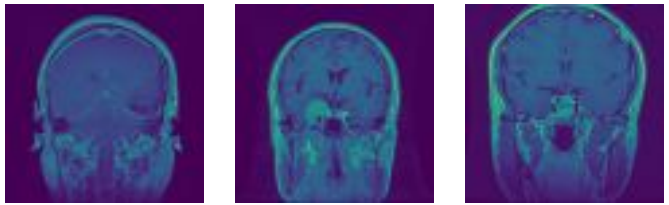


Figure 1. Left: Meningioma - usually appear as an enhancing mass on the outside lining of the brain tissue, which may or may not brighten with contrast. Malignant meningiomas can also invade the brain tissue. Middle: Glioma - called intra-axial brain tumors because they grow within the substance of the brain and often mix with normal brain tissue. Right: Pituitary tumor - and abnormal growth in the pituitary gland. The pituitary is a small gland in the brain. It is located behind the back of the nose.

Deep Learning Approaches

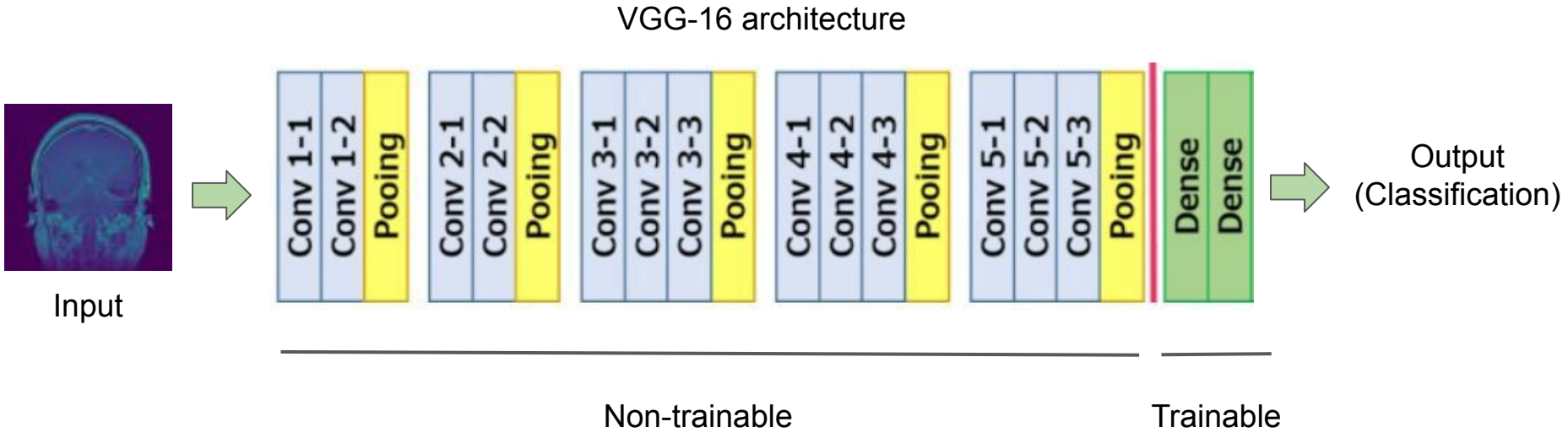
Classification:

- Transfer Learning
- Custom CNN

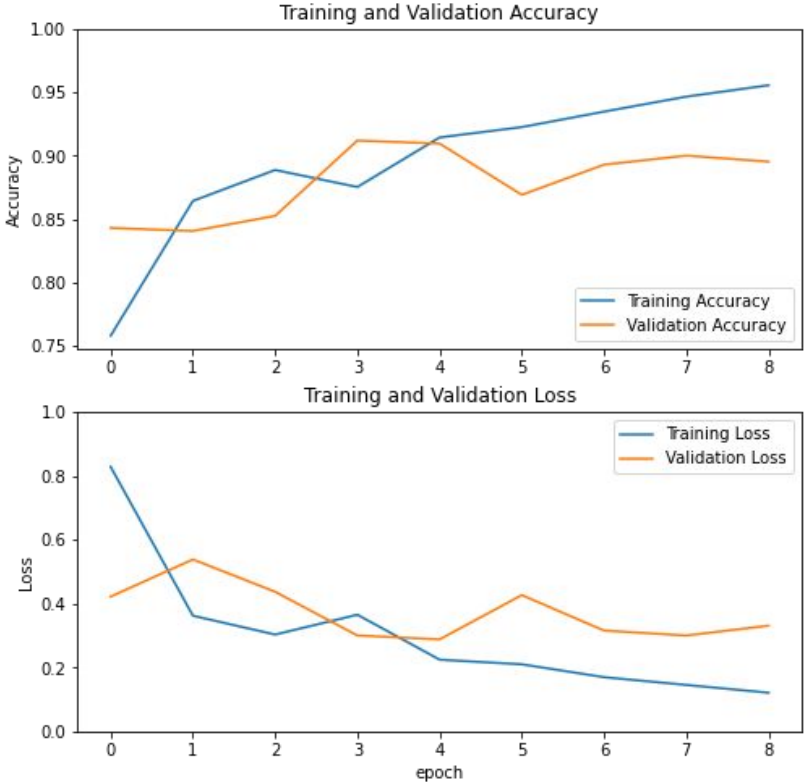
Localization:

- Custom UNET
- Gradient Tape

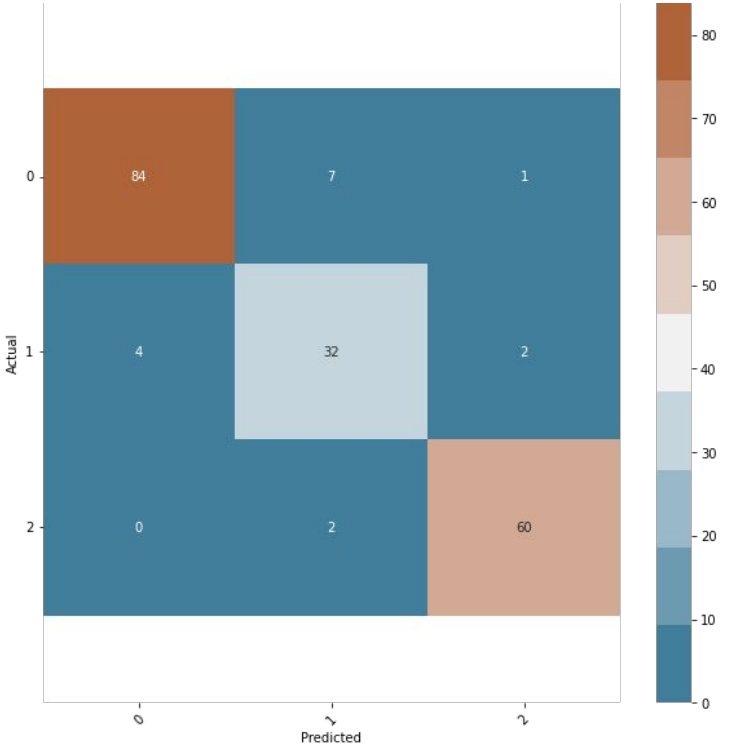
Transfer Learning: Methods



Transfer Learning: Results



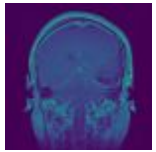
Transfer Learning: Results (Contd.)



Test set accuracy: 93.23%

Class	Precision	Recall	F1-score
Glioma	95%	91%	93%
Meningioma	78%	84%	81%
Pituitary tumor	95%	97%	96%
Average	**90%**	**91%**	**90%**

UNET Saliency Map: General Approach



Step 1: Train UNET

Input image
(512x512x3)

Encoder
(Conv Downstep)

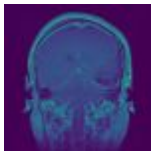
Latent (1x1x512)

Decoder
(Deconv Upstep)

Skip step
(512x512x1)
untrainable

Positional
information

Colorer



Output image
(512x512x3)

Step 2: Further train
encoder as classifier

Input image
(512x512x3)

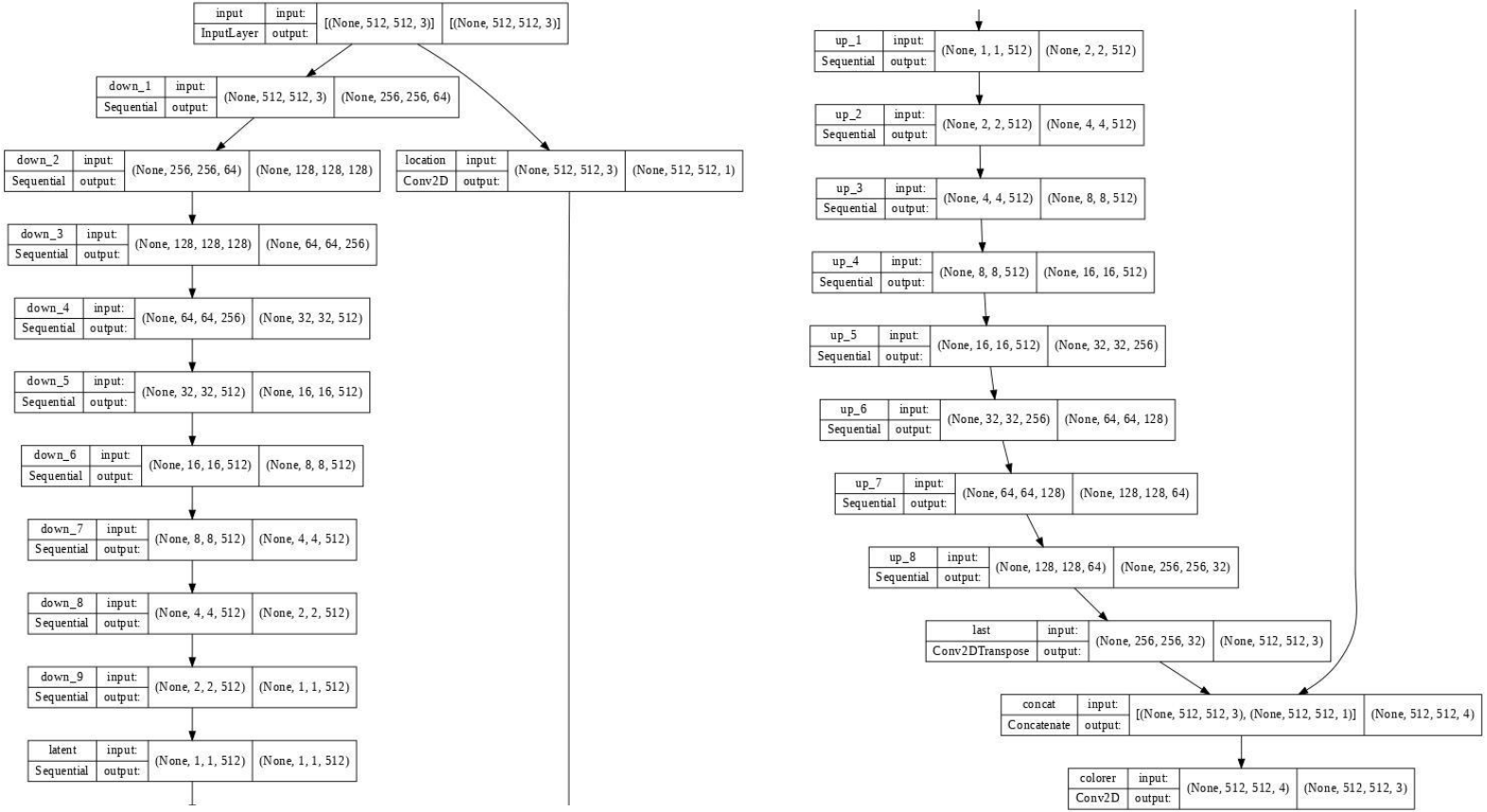
Further train as classifier

Encoder
(Conv Downstep)

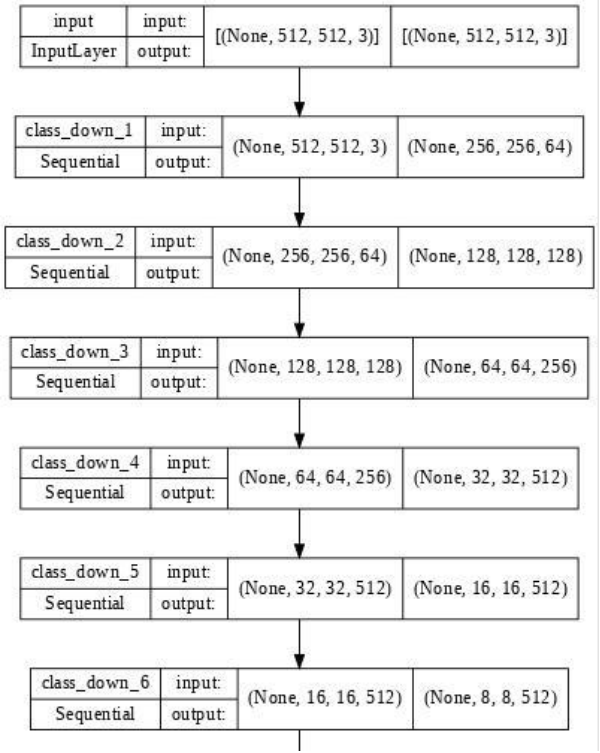
Latent to 3 classes
(1x1x512)=>1x1x3)

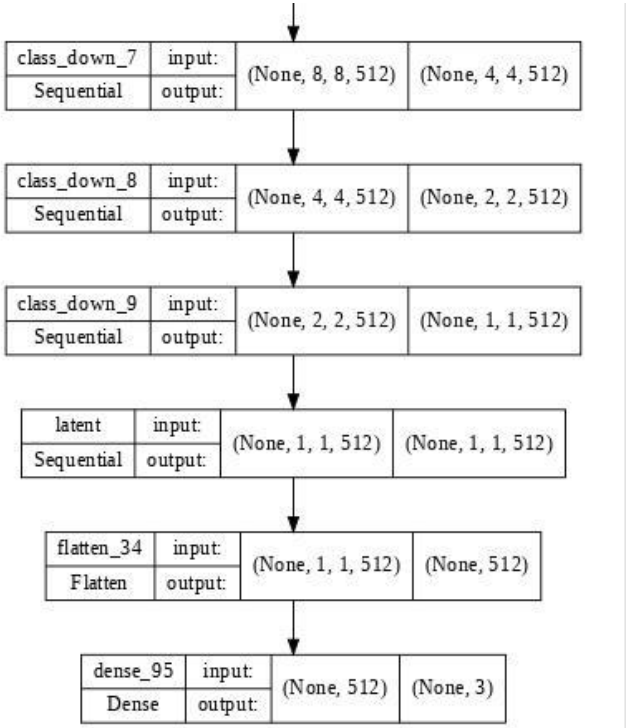
Output
(Classification)

UNET Saliency Map: UNET Layers

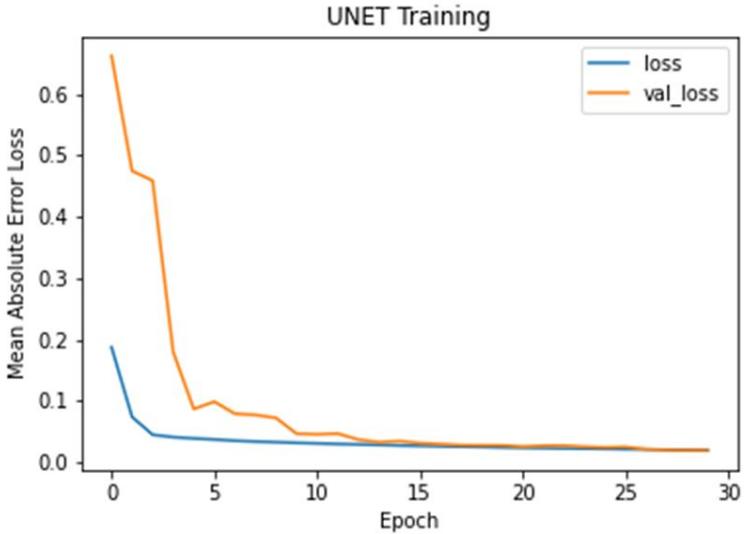


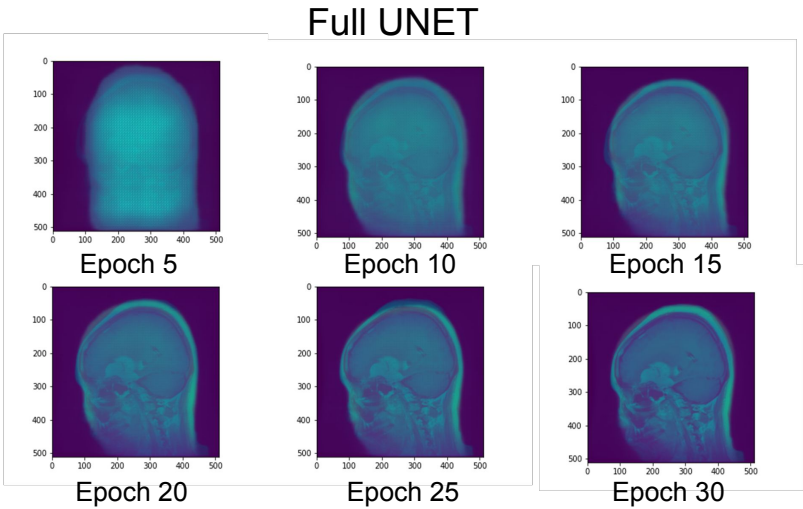
UNET Saliency Map: UNET Classifier Layers



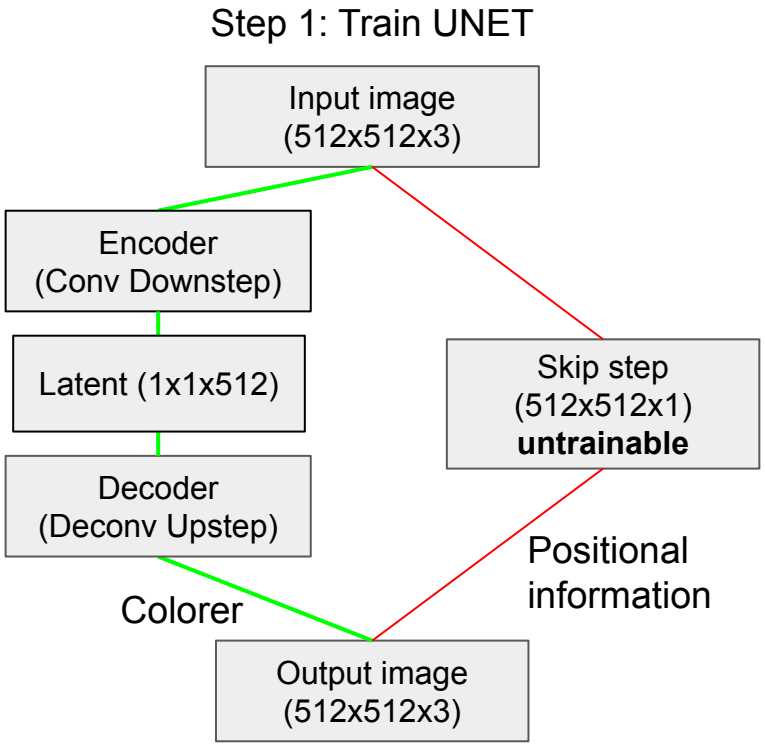


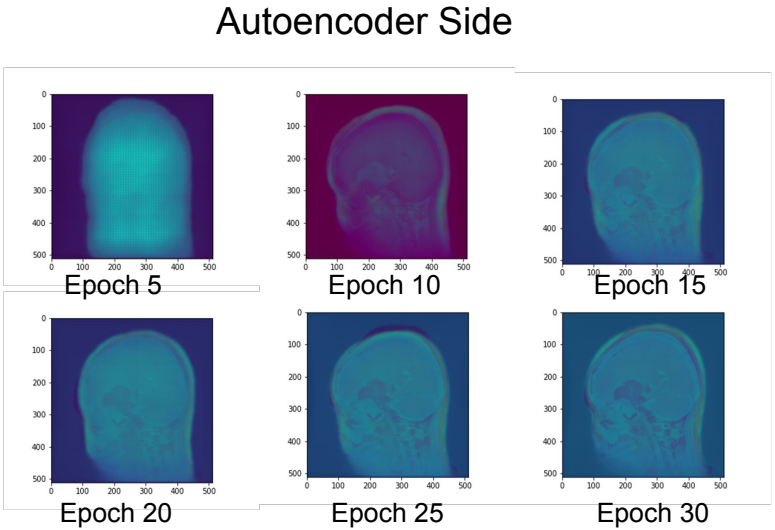
UNET Saliency Map: UNET Results



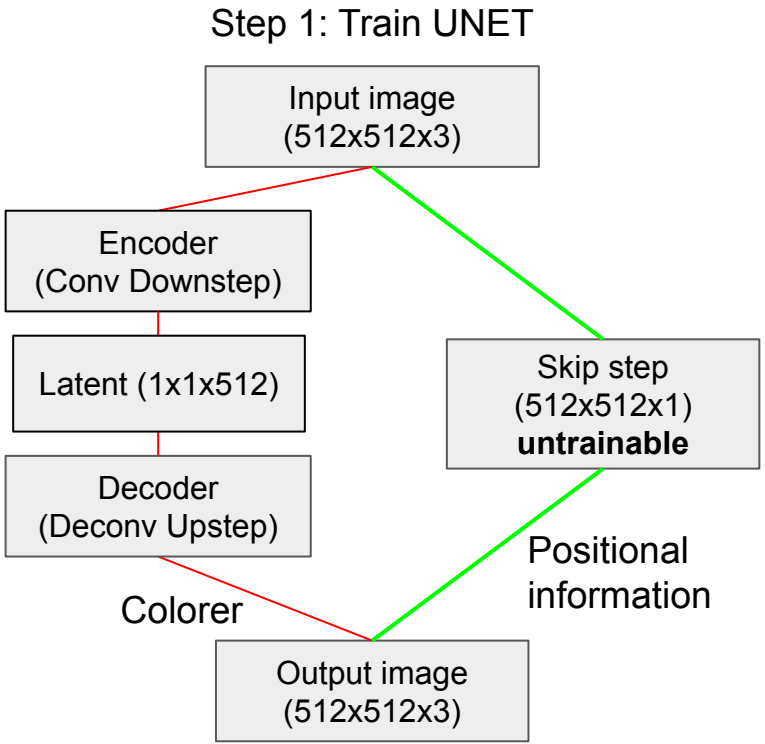


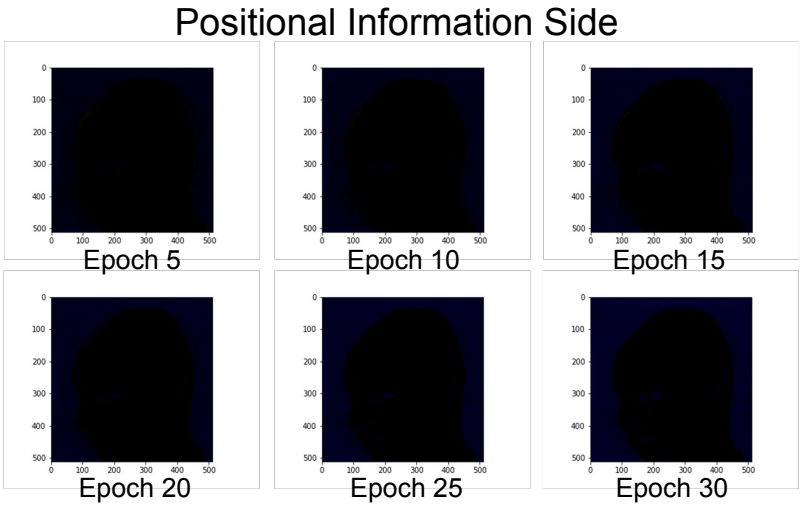
UNET Saliency Map: UNET Results



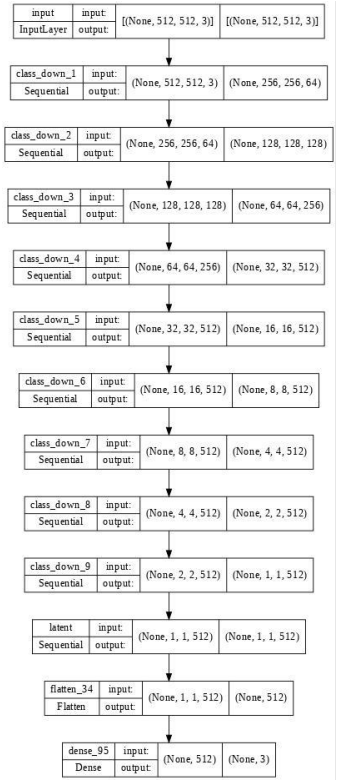


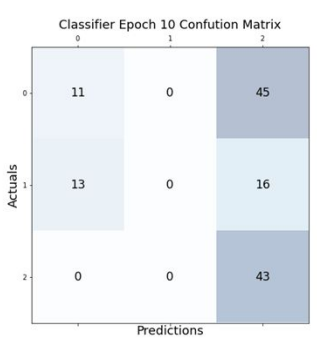
UNET Saliency Map: UNET Results

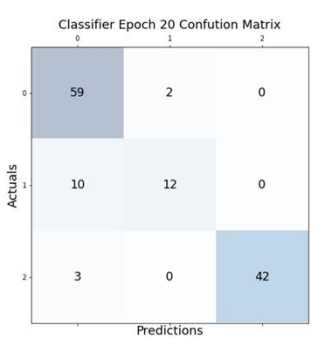


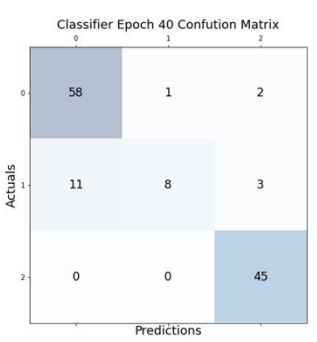


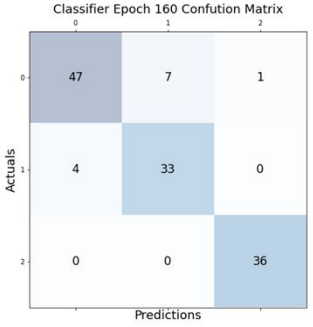
UNET Saliency Map: Encoder Training



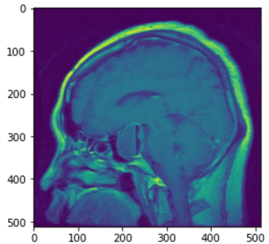




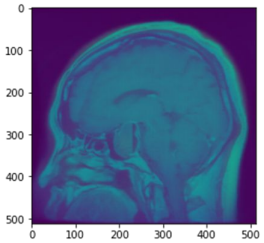




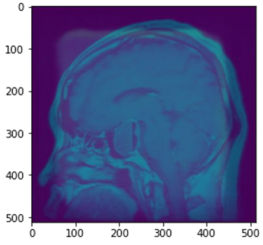
UNET Saliency Map: Encoder Training



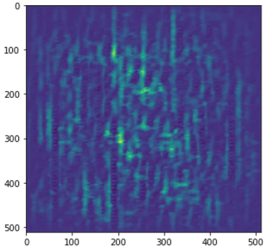
Original



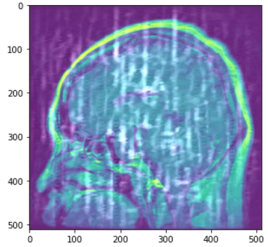
UNET



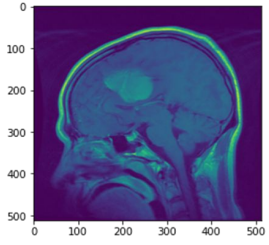
Classifier UNET



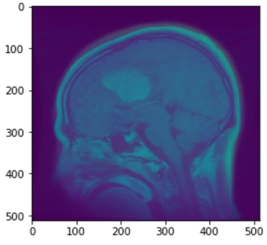
Gradients



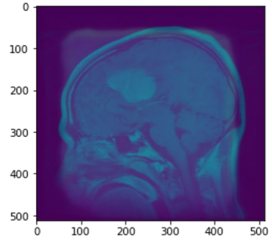
Gradients
on Original



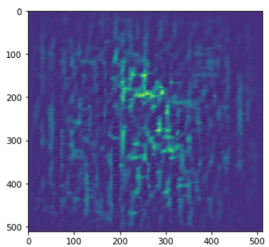
Original



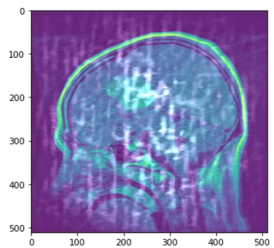
UNET



Classifier UNET

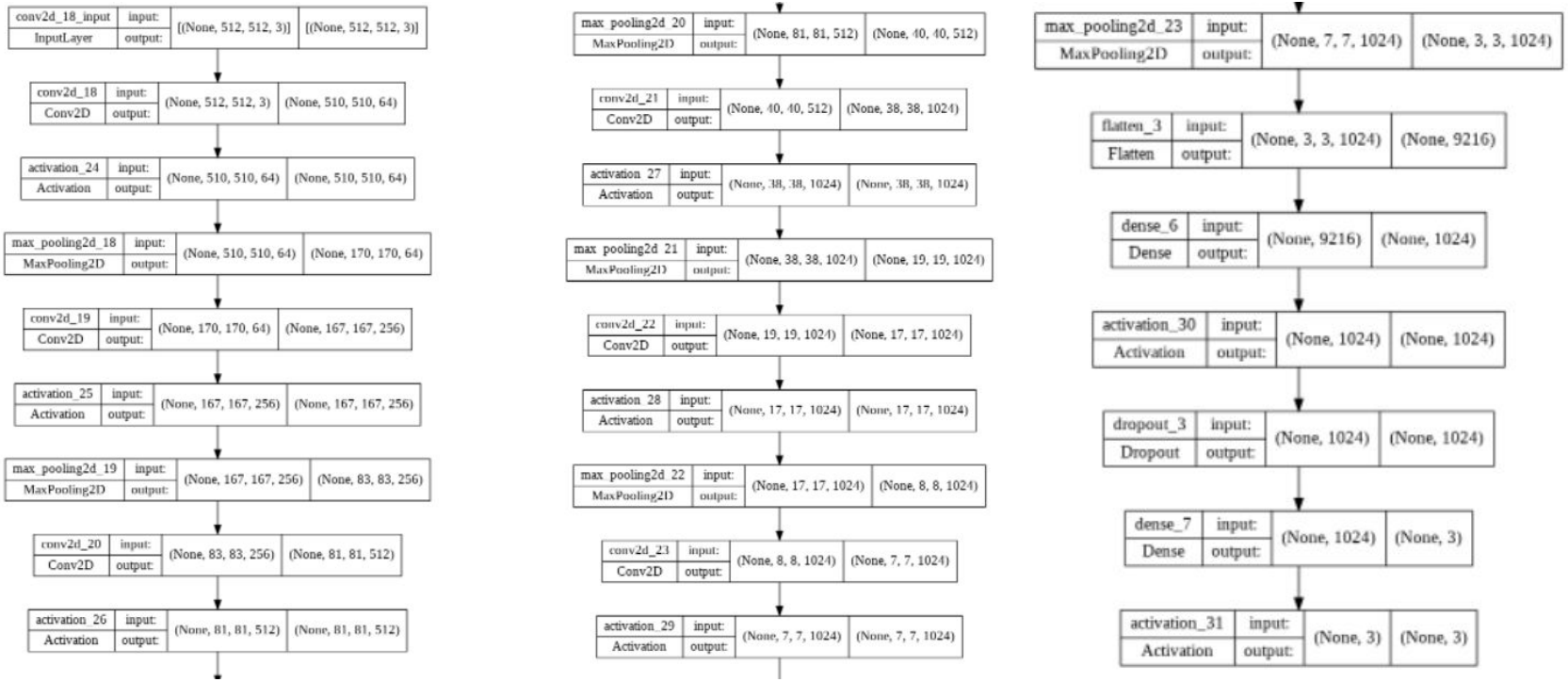


Gradients



Gradients
on Original

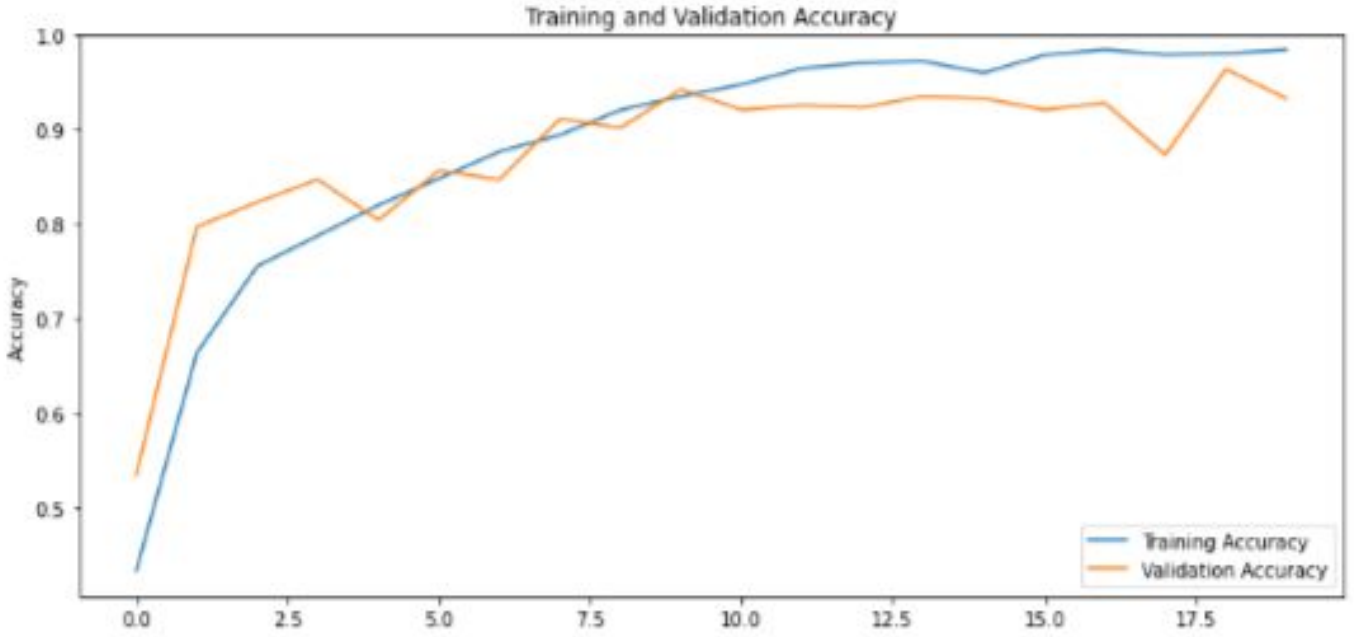
CNN


conv2d_18_input | input: | [(None, 512, 512, 3)] | [(None, 512, 512, 3)]
InputLayer | output:
conv2d_18 | input: | (None, 512, 512, 3) | (None, 510, 510, 64)
Conv2D | output:
activation_24 | input: | (None, 510, 510, 64) | (None, 510, 510, 64)
Activation | output:
max_pooling2d_18 | input: | (None, 510, 510, 64) | (None, 170, 170, 64)
MaxPooling2D | output:
conv2d_19 | input: | (None, 170, 170, 64) | (None, 167, 167, 256)
Conv2D | output:
activation_25 | input: | (None, 167, 167, 256) | (None, 167, 167, 256)
Activation | output:
max_pooling2d_19 | input: | (None, 167, 167, 256) | (None, 83, 83, 256)
MaxPooling2D | output:
conv2d_20 | input: | (None, 83, 83, 256) | (None, 81, 81, 512)
Conv2D | output:
activation_26 | input: | (None, 81, 81, 512) | (None, 81, 81, 512)
Activation | output:
max_pooling2d_20 | input: | (None, 81, 81, 512) | (None, 40, 40, 512)
MaxPooling2D | output:
conv2d_21 | input: | (None, 40, 40, 512) | (None, 38, 38, 1024)
Conv2D | output:
activation_27 | input: | (None, 38, 38, 1024) | (None, 38, 38, 1024)
Activation | output:
max_pooling2d_21 | input: | (None, 38, 38, 1024) | (None, 19, 19, 1024)
MaxPooling2D | output:
conv2d_22 | input: | (None, 19, 19, 1024) | (None, 17, 17, 1024)
Conv2D | output:
activation_28 | input: | (None, 17, 17, 1024) | (None, 17, 17, 1024)
Activation | output:
max_pooling2d_22 | input: | (None, 17, 17, 1024) | (None, 8, 8, 1024)
MaxPooling2D | output:
conv2d_23 | input: | (None, 8, 8, 1024) | (None, 7, 7, 1024)
Conv2D | output:
activation_29 | input: | (None, 7, 7, 1024) | (None, 7, 7, 1024)
Activation | output:
max_pooling2d_23 | input: | (None, 7, 7, 1024) | (None, 3, 3, 1024)
MaxPooling2D | output:
flatten_3 | input: | (None, 3, 3, 1024) | (None, 9216)
Flatten | output:
dense_6 | input: | (None, 9216) | (None, 1024)
Dense | output:
activation_30 | input: | (None, 1024) | (None, 1024)
Activation | output:
dropout_3 | input: | (None, 1024) | (None, 1024)
Dropout | output:
dense_7 | input: | (None, 1024) | (None, 3)
Dense | output:
activation_31 | input: | (None, 3) | (None, 3)
Activation | output:

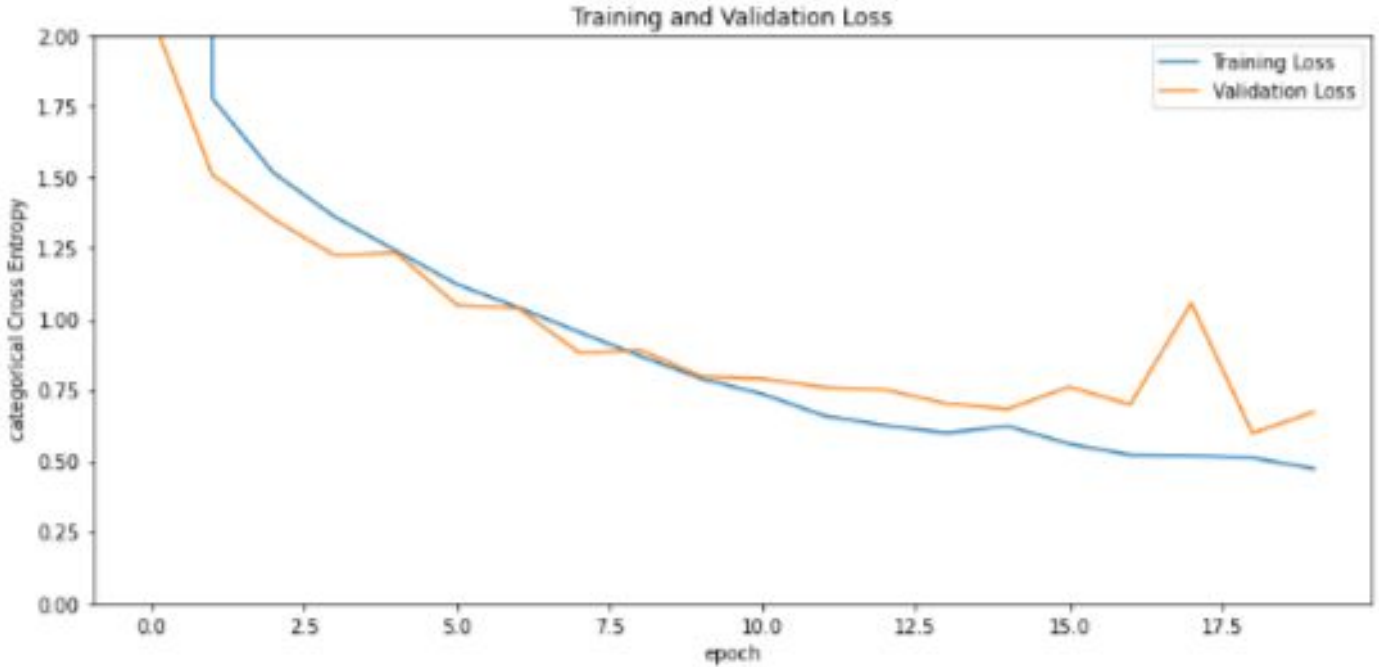
CNN: Results

```
Epoch 1/20
77/77 [==============================] - 49s 567ms/step - loss: 63.9858 - accuracy: 0.4343 - val_loss: 2.0782 - val_accuracy: 0.5357
Epoch 2/20
77/77 [==============================] - 44s 538ms/step - loss: 1.7791 - accuracy: 0.6644 - val_loss: 1.5092 - val_accuracy: 0.7976
Epoch 3/20
77/77 [==============================] - 44s 538ms/step - loss: 1.5160 - accuracy: 0.7561 - val_loss: 1.3527 - val_accuracy: 0.8238
Epoch 4/20
77/77 [==============================] - 44s 539ms/step - loss: 1.3607 - accuracy: 0.7883 - val_loss: 1.2247 - val_accuracy: 0.8476
Epoch 5/20
77/77 [==============================] - 44s 541ms/step - loss: 1.2416 - accuracy: 0.8206 - val_loss: 1.2347 - val_accuracy: 0.8048
Epoch 6/20
77/77 [==============================] - 44s 540ms/step - loss: 1.1225 - accuracy: 0.8487 - val_loss: 1.0499 - val_accuracy: 0.8571
Epoch 7/20
77/77 [==============================] - 44s 540ms/step - loss: 1.0410 - accuracy: 0.8772 - val_loss: 1.0382 - val_accuracy: 0.8476
Epoch 8/20
77/77 [==============================] - 43s 535ms/step - loss: 0.9537 - accuracy: 0.8948 - val_loss: 0.8809 - val_accuracy: 0.9119
Epoch 9/20
77/77 [==============================] - 44s 539ms/step - loss: 0.8691 - accuracy: 0.9213 - val_loss: 0.8906 - val_accuracy: 0.9024
Epoch 10/20
77/77 [==============================] - 44s 537ms/step - loss: 0.7896 - accuracy: 0.9356 - val_loss: 0.7971 - val_accuracy: 0.9429
Epoch 11/20
77/77 [==============================] - 44s 538ms/step - loss: 0.7352 - accuracy: 0.9482 - val_loss: 0.7893 - val_accuracy: 0.9214
Epoch 12/20
77/77 [==============================] - 43s 536ms/step - loss: 0.6592 - accuracy: 0.9653 - val_loss: 0.7594 - val_accuracy: 0.9262
Epoch 13/20
77/77 [==============================] - 44s 538ms/step - loss: 0.6247 - accuracy: 0.9710 - val_loss: 0.7505 - val_accuracy: 0.9238
Epoch 14/20
77/77 [==============================] - 44s 538ms/step - loss: 0.5988 - accuracy: 0.9727 - val_loss: 0.7023 - val_accuracy: 0.9357
Epoch 15/20
77/77 [==============================] - 44s 540ms/step - loss: 0.6227 - accuracy: 0.9604 - val_loss: 0.6826 - val_accuracy: 0.9333
Epoch 16/20
77/77 [==============================] - 43s 536ms/step - loss: 0.5600 - accuracy: 0.9792 - val_loss: 0.7611 - val_accuracy: 0.9214
Epoch 17/20
77/77 [==============================] - 44s 538ms/step - loss: 0.5211 - accuracy: 0.9849 - val_loss: 0.6985 - val_accuracy: 0.9286
Epoch 18/20
77/77 [==============================] - 43s 536ms/step - loss: 0.5190 - accuracy: 0.9796 - val_loss: 1.0567 - val_accuracy: 0.8738
Epoch 19/20
77/77 [==============================] - 44s 541ms/step - loss: 0.5106 - accuracy: 0.9808 - val_loss: 0.5980 - val_accuracy: 0.9643
Epoch 20/20
77/77 [==============================] - 44s 537ms/step - loss: 0.4729 - accuracy: 0.9849 - val_loss: 0.6723 - val_accuracy: 0.9333
```

Training and validation accuracy



Training and validation loss

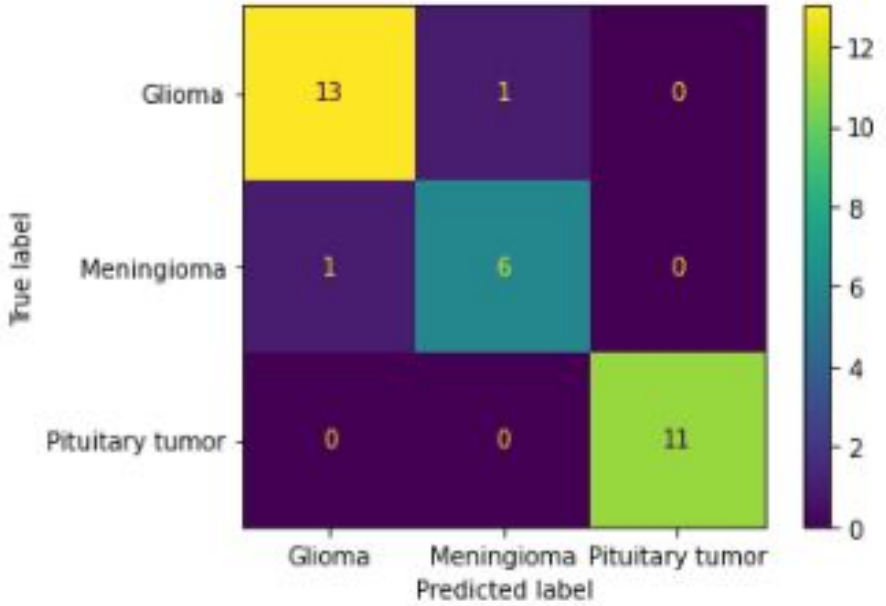


Evaluation on the test data

```
model.evaluate(test_ds)
```

```
6/6 [==============================] - 4s 343ms/step - loss: 0.7461 - accuracy: 0.9219
[0.7461099624633789, 0.921875]
```

CNN: Confusion matrix



CNN: Accuracy, Precision, Recall & F1 Score

	precision	recall	f1-score	support
0	0.93	0.93	0.93	14
1	0.86	0.86	0.86	7
2	1.00	1.00	1.00	11
accuracy			0.94	32
macro avg	0.93	0.93	0.93	32
weighted avg	0.94	0.94	0.94	32

Limitation and Future work

- Only VGG-16 was used in transfer learning; try different pretrained models and explore with fine-tuning the whole architecture.
- For CNN, only 32 samples has been added in the test dataset
- For CNN, 6 layers of Conv layers has been used ; Try CNN model with different layers.
- For Saliency Map, even the gradient tape method did not show a clear localization of the tumor. Both methods could be run again on a more localizable dataset.