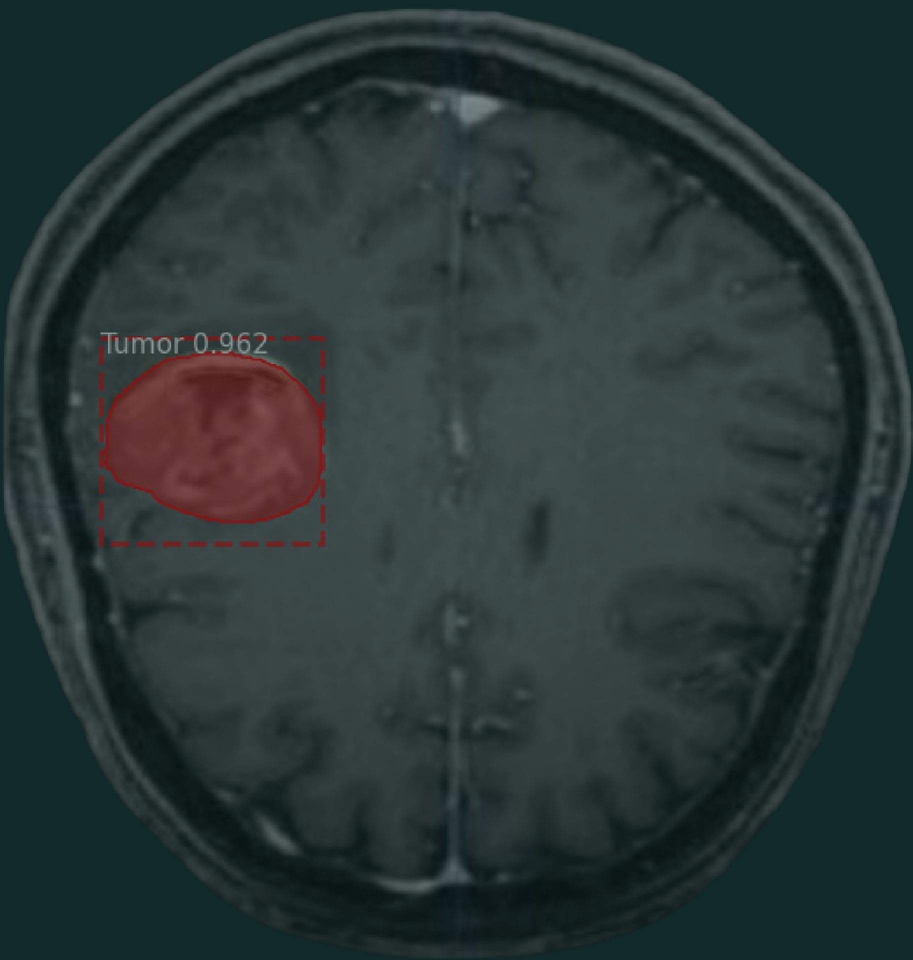
BRAIN, BREAST RADIOLOGY CLASSIFICATION & SEGMENTATION



PREPARED BY:

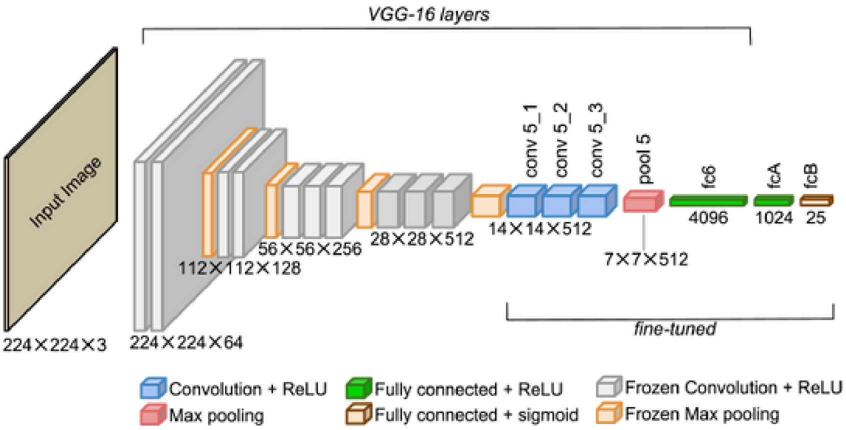
Alzahraa Mohamed Shaheen
Manal Gamal Elsarawy
Ahmed Rabie Galal Taha
Ehab Nabil Fathy Elfeel
Mahmoud Samir Gooda

Task 1 (Brain-Breast Classification):

- At this task we didn’t find it necessary to do any image preprocessing because the two classes were far away from each other in the image features and the model gave us good results without any preprocessing.

- We used for this task a base model of VGG16 and added a Flatten layer, Dropout layer which drops 0.5 of the weights and a dense layer of 1 neuron for binary classification with 'sigmoid' activation function.

- VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date. VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy when it trained and tested using 'ImageNet' dataset. It is one of the popular algorithms for image classification and is easy to use with transfer learning. Here is the architecture:

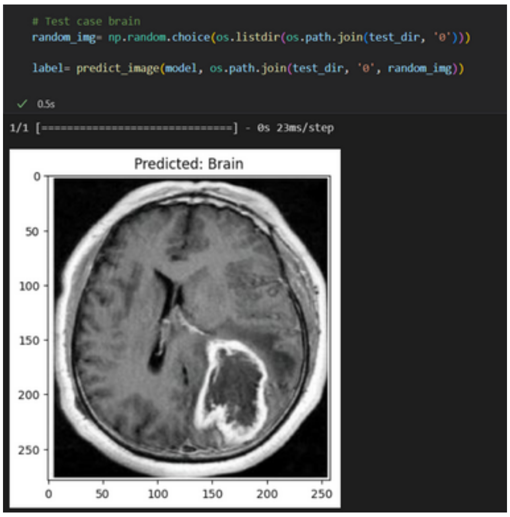


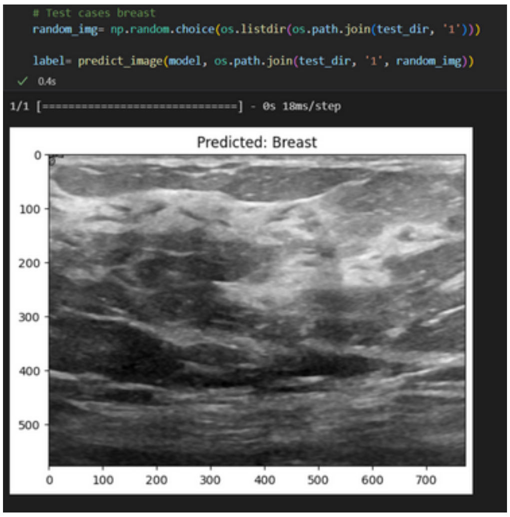
- As we found the data is unbalanced so we used class weights when we trained the model.

- The model trained for two epochs in 57.4 seconds and gave the following scores:
  - Validation loss: 8.366e-05
  - Validation f1-score: 1.00

- In the testing phase, the model tested over the test generator in 3.6 seconds and gave the following scores:
  - Test loss: 8.894e-05
  - Test f1-score: 1.00

- Test cases:





Task 2 (Brain Tumor Classification):

- At this task we didn't do any image preprocessing and the model gave us good results without any preprocessing.

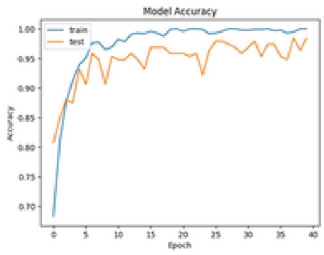
- We explored the data set and found it balanced.

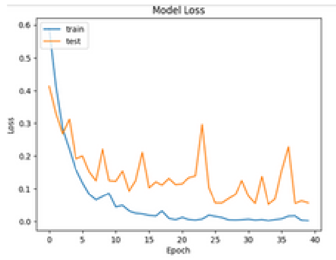
- For data preprocessing step ,we collected the data set in two data frames (train&test) then shuffled them and tried different data augmentations to increase training data and expose our model to a wider variety of training examples, which can help it generalize better to new, unseen data.

- We used for this task a base model of VGG16 and added a Flatten layer, Dropout layer which drops 0.5 of the weights and a dense layer of 1 neuron for binary classification with 'sigmoid' activation function.

- The model trained for 40 epochs in 929 seconds and gave the following scores:

  Test loss: 0.0560
  Test accuracy: 0.9844

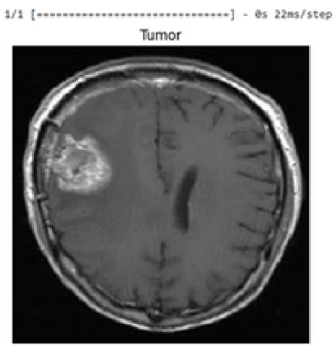


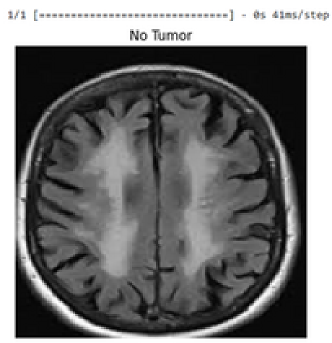


- In the evaluation phase the model tested over the test generator in 1 second and gave the following scores:

  Test loss: 0.0538
  Test accuracy: 0.9850

- Test cases for the used model :





Task 3 (Breast Classification):

- At this task the only image preprocessing we found necessary was to resize all the images to the same size and assure that they all have the same dimensions.

- We found that the data is unbalanced, so we experimented with the data once in an unbalanced state and another time after balancing the data through up sampling and using classes weight for the 3 classifiers (0: benign, 1: malignant, 2: normal).

- Since these are medical images, we performed simple data augmentation by rescaling the images, applying zoom, and applying horizontal flip.

- We tried for this task four base models VGG16, ResNet50, MobileNetV2, MobileNet.

- The models were trained over 25 epocs with learning rate of1e-5.

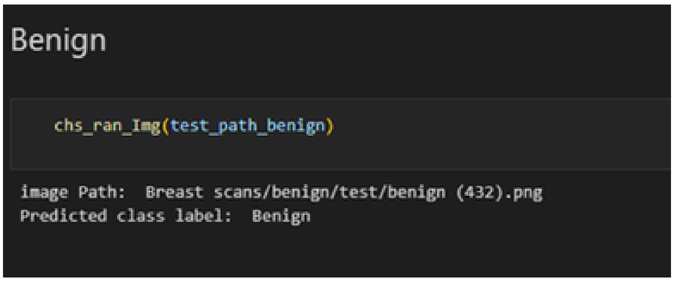
- The best result were obtained using the unbalanced data were with VGG16 with which gave accuracy of 68%**.**

- The best results were obtained after resampling using VGG16 with accuracy of 73%.

- The model trained for 25 epochs in 12m 23s and gave the following metrics:
  - val loss: 0.8262
  - val accuracy: 0.7396

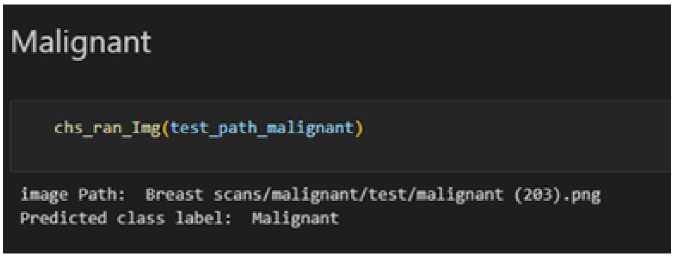
- In the testing phase, the model tested over the test data in 1 second and gave the following scores:
  - loss: 0.9054
  - accuracy: 0.7300

- Test Cases:


Benign

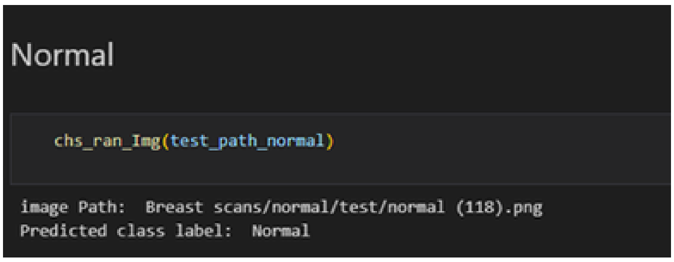
```
chs_ran_Img(test_path_benign)
```

```
image Path:  Breast scans/benign/test/benign (432).png
Predicted class label:  Benign
```


Malignant

```
chs_ran_Img(test_path_malignant)
```

```
image Path:  Breast scans/malignant/test/malignant (203).png
Predicted class label:  Malignant
```


Normal

```
chs_ran_Img(test_path_normal)
```

```
image Path:  Breast scans/normal/test/normal (118).png
Predicted class label:  Normal
```

Task 4 (Brain Tumor Segmentation):

- At this task the only image preprocessing we found necessary was to resize all the images and masks to the same size and assure that they all have the same dimensions.

- For the segmentation process in this task we have used Mask-RCNN pretrained model on 'MS COCO' dataset and retrained it on our data.

- MRCNN is a deep learning architecture for instance segmentation, which involves identifying and segmenting each object instance in an image.

- MRCNN consists of three main components: a backbone network, a region proposal network (RPN), and a mask branch.

- The backbone network is a pre-trained CNN that extracts features from the input image. It shares the feature extraction stage output between the RPN and the mask branch, which makes the architecture more efficient and easier to train.

- The RPN generates a set of object proposals by sliding a small window over the feature map output by the backbone network and predicting whether each window contains an object or not.

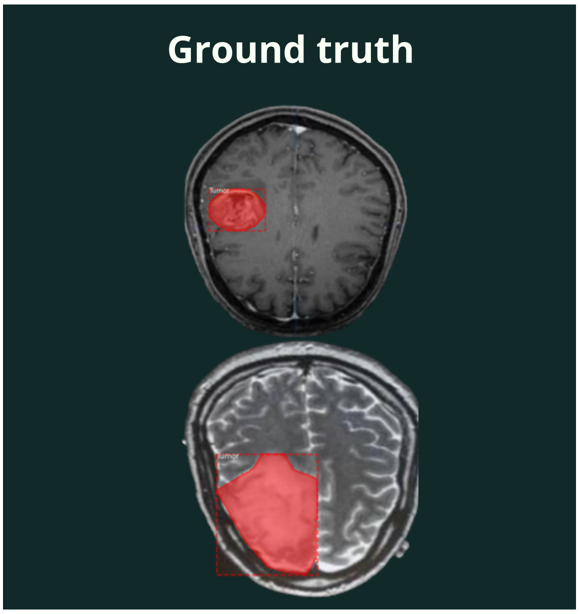
- The mask branch takes as input the features of each proposed object and generates a binary mask for each object instance.

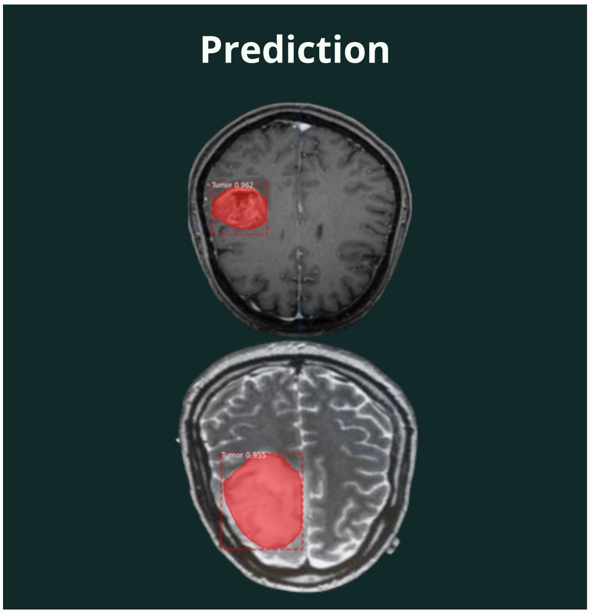
- During training, MRCNN minimizes some loss functions: a classification loss for the RPN, a classification and regression loss for the object detection task, and a binary cross-entropy loss for the mask prediction task.

- The model trained for 5 epochs in 22m 10.6s.

- In the testing phase, the model tested over the test generator in 29.6 seconds and gave the following scores:
  - Average pericesion : 80.4%
  - Average IoU : 63%
  - Average Dice Coefficient: 72%

- Test cases:





Task 5 (Breast Tumor Segmentation):

- For this task, the only image preprocessing we found necessary was to resize all the images and masks to the same size and assure that they all have the same dimensions 256*256 and normalize them.

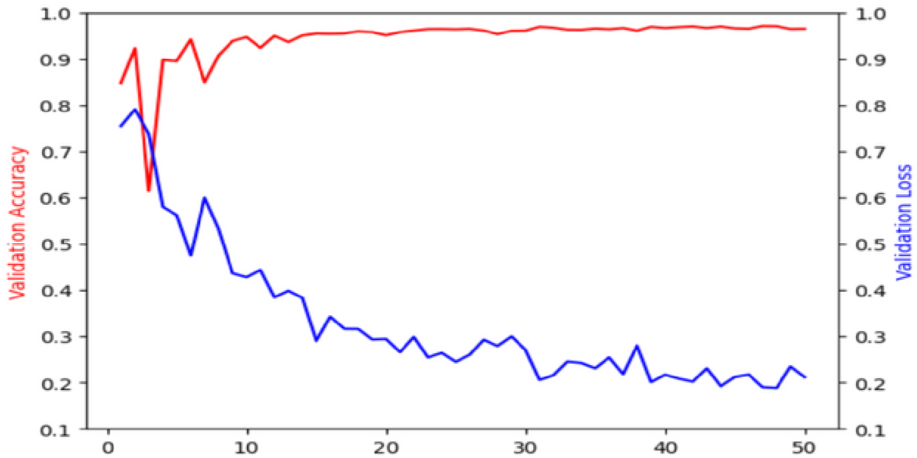
- Also as image processing, we founnd that 'benign' class had 16 images with multiple masks so, we had to merge the two masks into one mask to be the ground truth for the image.

- For the segmentation task we used U-Net model with the fpllowing parameters:
    - Trainable parameters: 31,390,721
    - Non-trainable parameters: 11,776

- The model was trained for 50 epochs with batch size of 16.

- We used the technique of reduced learning rate to update our learning rate during training phase.

- The model trained in 39m 20s in training and gave validation metrics as shown:



- In the testing phase, the model was tested on the testing data in 16.08 seconds, and gave the following scores:
    - IoU: 74.6%
    - F1 Score: 80%
    - Precision: 80.8%