Contents of folder: **Diagnosis_breast_cancer_MRI/**

MRI/ ..... **Page 2**
- **Breast_MRI_001/**
  - T1_axial_slope1.nii.gz
  - T1_axial_slope2.nii.gz
  - T1_axial_01.nii.gz
  - T1_axial_02.nii.gz

Model/ ..... **Page 3**
- CNN_weights.npy

Scripts/ ..... **Page 4**
- model_utils.py
- make_diagnosis_on_MRI.py
- train_model.py

Training_data/ ..... **Page 5**
- Data_Description.csv
- **images/**
  - SUBJECT2_EXAM1_l_14.png
  - SUBJECT4_EXAM1_r_12.png
  - SUBJECT5_EXAM1_r_24.png
  - SUBJECT3_EXAM1_l_33.png
  - SUBJECT1_EXAM1_r_28.png

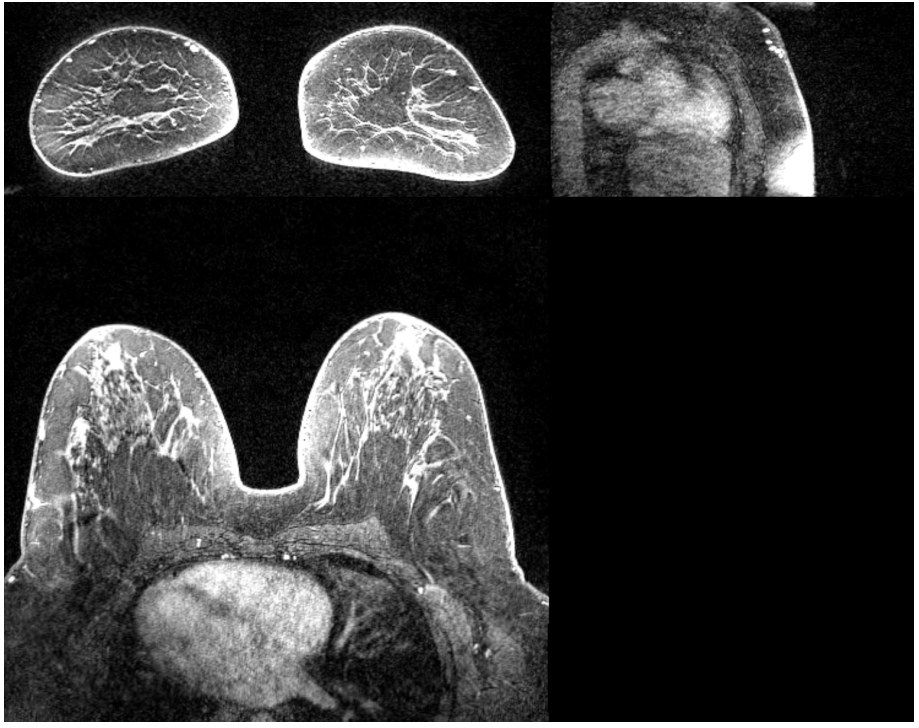
Sessions/ ..... **Page 7**

../MRI/

../Breast_MRI_001/

This folder contains an axial MRI image of a breast. The image is obtained from a public dataset released by Duke University (https://sites.duke.edu/mazurowski/resources/breast-cancer-mri-dataset/)

Nifty format, recommended viewer: MRIcro (https://people.cas.sc.edu/rorden/mricro/mricro.html)



../model/

This folder contains the weights of a pre-trained neural network model used for breast cancer detection, called CNN_weights.npy

The format is of a numpy array

How to load:

```
model_weights = np.load( '/../model/CNN_weights.npy' , allow_pickle=True)

from model_utils import FocalLoss, UNet_v0_2D_Classifier

model = UNet_v0_2D_Classifier(input_shape =  (512,512,3), pool_size=(2, 2),initial_learning_rate=1e-5,
                              deconvolution=True, depth=6, n_base_filters=42,
                              activation_name="softmax", L2=1e-5, USE_CLINICAL=True)

model.set_weights(model_weights)
```

../scripts/

This folder contains three Python scripts: Note that there is no patient or image information of any kind in these files.

a) load_model_and_predict.py

This script loads the pre-trained model weights and an MRI image. It then generates a prediction of breast cancer from the whole volume based on the loaded model.

b) train_new_model.py

This script is used to train a new model on training data. It takes the training data, including images and relevant information, and trains a new model for breast cancer detection.

c) utility_functions.py

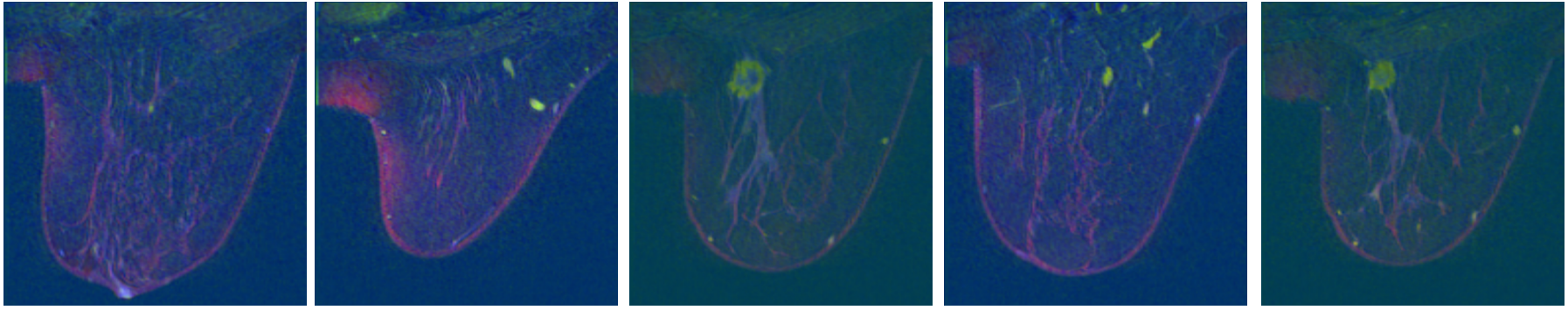
This script contains utility functions used in the other two scripts.

../Training_data/

Data_Description.csv
../images/
SUBJECT2_EXAM1_l_14.png
SUBJECT4_EXAM1_r_12.png
SUBJECT5_EXAM1_r_24.png
SUBJECT3_EXAM1_l_33.png
SUBJECT1_EXAM1_r_28.png

Description: This folder contains the necessary files for a new training session. It includes a spreadsheet with information on the training data and a folder with images to train the network. These images are examples taken from the same public dataset from Duke University. They are saved as PNG images for easier visualization. Note that due to the RGB color format, the images may appear unusual.

The naming convention for the images is: Subject_Exam_Side_SliceNumber. The examples have fake names to serve as an example.



Spreadsheet: Data_Description.csv

The specific data here was generated at random and only serves the purpose of demonstrating the expected data format:

DE-ID: De-identified patient identifier. In this case just a number

Exam: Identifier for exam when image was taken

Scan_ID : Identifier of a specific scan: Constructed from DE-ID + Exam + side (in case of a saggital image)

Partition : Describes whether image is part of the training or validation set

Image : Points to location of the specific file containing the image of a scan. Name is constructed using DE-ID + Exam + Scan_ID + slice number of image.

BIRADS : From clinical information of exam

Pathology : From clinical information of exam

Age : From clinical information of exam

Family Hx : From clinical information of exam

Ethnicity and Race : From clinical information of exam

../sessions/

This folder is used to store the output of each new training session. It saves the trained models, evaluation metrics, and any other relevant information generated during the training process.

By the moment of sharing this folder will be empty.