**DATASET:**

It was collected from,

<https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=68550661>

**DATA FORMAT:**

1. Ultrasound Data in DICOM format
2. MRI Data in DICOM format

**DATA DEFINITION:**

“Prostate-MRI-US-Biopsy-XXXX-SURFACETYPE-seriesUID-YYYY.STL" where "XXXX" is the anonymized patient number and "YYYY" is the series instance UID of corresponding DICOM images.”

After extracting the dataset,

1. “Rendering” Holds Ultrasound Data
2. “MRI” holds MRI Data

**DATA LOGIC:**

Benign : UCLA Score < 2

Malignant: UCLA Score >= 2

**DATA DISTRIBUTION:**

|  |  |
| --- | --- |
| Benign | **131** |
| Malignant | **1490** |

**FRAME DISTRIBUTION:**

|  |  |
| --- | --- |
|  |  |

**DATA PROCESSING, FORMATTING and INDEXING**:

1. Ultrasound: Take an equal number of images in both sides of the middle of the no of Frames and dump it as PNG format for both benign and malignant.
2. MRI: Dump the image as JPG format for both benign and malignant format.
3. USE PREFIX as US\_BENIGN|MALIGNANT\_NUMBER for Ultrasound Images

USE PREFIX as MRI\_BENIGN|MALIGNANT\_NUMBER for MRI Images

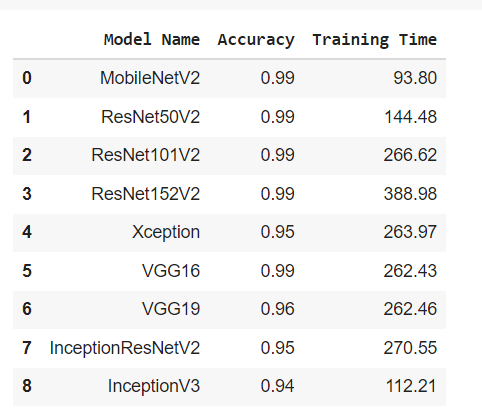
**DATA SPLITTING:**

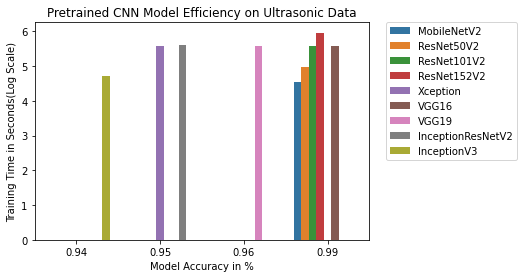
1. Make a batch of images by the orientation of BENIGN and MALIGNANT images.
2. Generate TF Records/Tensorflow Records from the batches where each TF record contains 253 images of BENIGN and MALIGNANT images.
3. Split it 80% TF Records for Training and 20% Records for Testing.

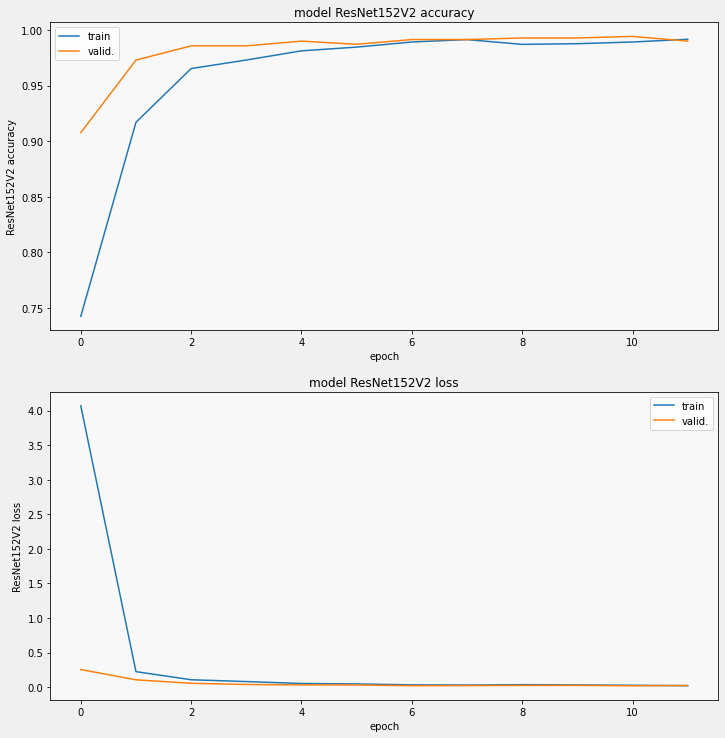
**MODEL DEVELOPMENT AND TRAINING:**

We used pretrained models with transfer learning strategy and measured its validation accuracy.

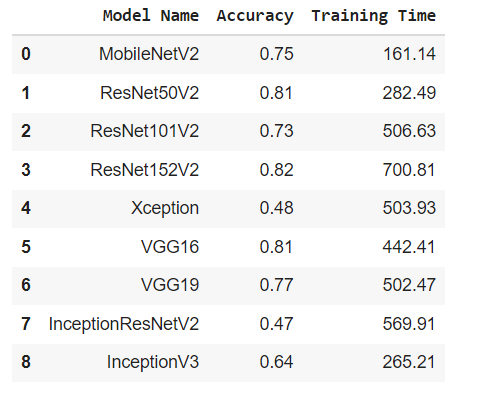
**Model Performance on Ultrasound Data**

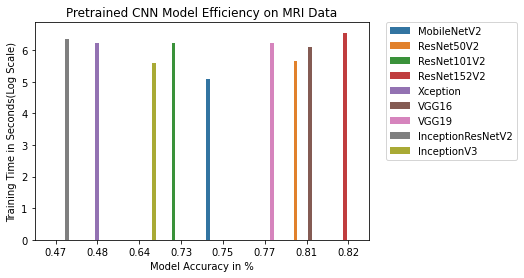
****

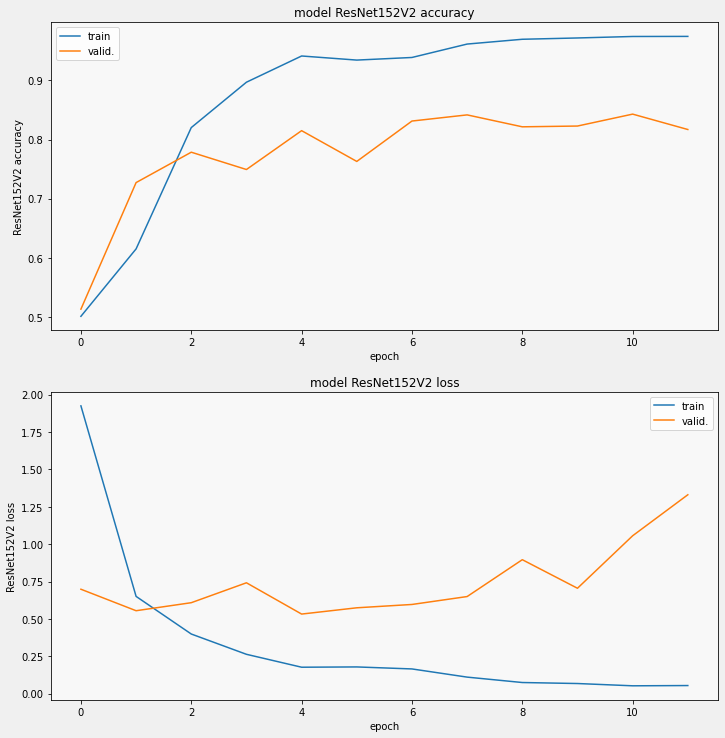




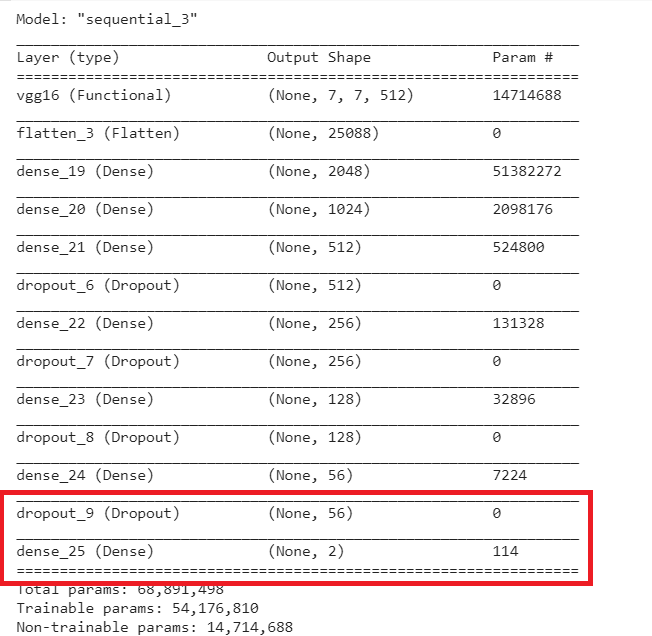
**Model Performance on MRI Data**

****





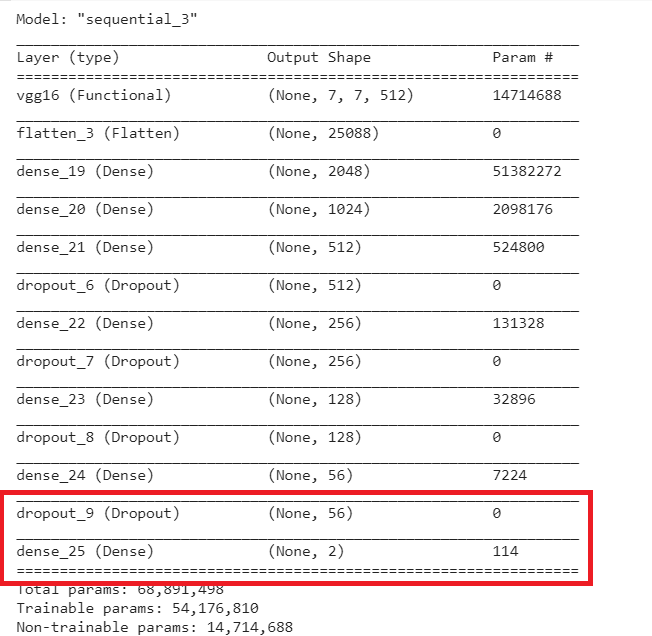
The neural structure of each model for both MRI and ULTRASOUND was:



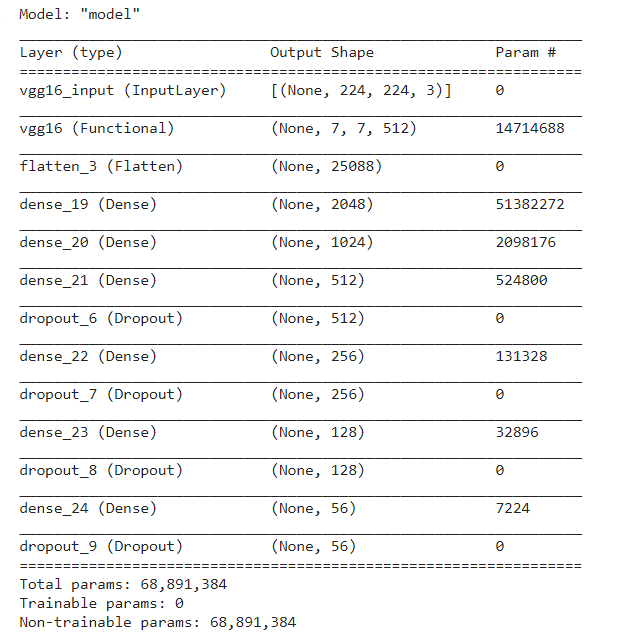
**Improvement strategies of the MODEL made by MRI DATASET:**

**Strategy 1:**

1. Load the saved Model



1. Remove the top layers



1. Extract the features
2. Run the classification algorithm using “KNeighborsClassifier”, “RandomForestClassifier” and “GradientBoostingClassifier”

|  |
| --- |
| KNeighborsClassifier |
|  |
| RandomForestClassifier |
|  |
| GradientBoostingClassifier |
|  |

**Strategy 2:**

1. Load the Fresh Pretrained Model VGG16
2. Remove the top layers
3. Extract the features
4. Run the classification algorithm using “RandomForestClassifier”(n\_estimators = 50, random\_state = 42)

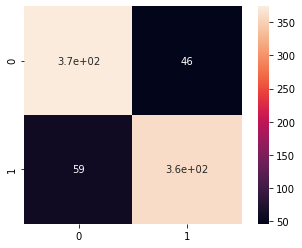
**Accuracy = 88%**

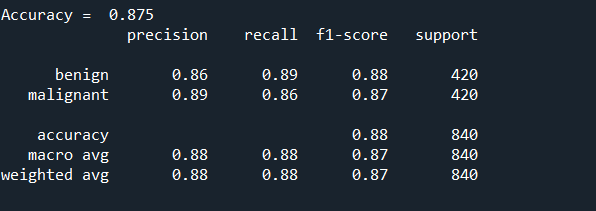
**[[374 46]**

**[ 59 361]]**

**The prediction for this image is: ['malignant']**

**The actual label for this image is: malignant**

****

****

Run the classification algorithm using “KNeighborsClassifier”(n\_neighbors=5, n\_jobs=-1)

Accuracy - 87%

