**My Approach to Classifying the images**

For this classification problem, I’ve tried transfer learning with VGG16 as a base architecture. I’ve included only the convolutional layer from VGG16 and trained ‘imagenet’ weights. Some Fully Connected and Final Softmax layer is added for classification.

**Data Preparation:** The images from **mias** database is downloaded and the images with ‘B’ and ‘M’ labels are extracted, using python script separate\_benign\_and\_malignant.py, for preparing the train, validation, and test set. The total images with ‘B’ and ‘M’ labels were divided into approximately (70-15-15) % for training, validation, and testing.

**Training / Validation:** The VGG16 model with few modifications was trained with/validated against the **mias** training/validation images. The training was done with 20 epochs and optimized with Adam optimizer (lr=1e-5). After the training is completed the training script (train.py) saves the complete trained model (model.h5) in ‘trained\_models’ directory which can be used for inference. The training can be performed by running the following command.

* python train.py

**Testing:** The trained model is then loaded by the testing script (test.py) to test the performance of the trained model. The testing can be done with following command.

* python test.py

***Utils:*** The training and testing parameters can be found in utils.py and can be changed to play around with the parameters during training and testing.

**Dependencies**: keras, tensorflow, matplotlib, PIL