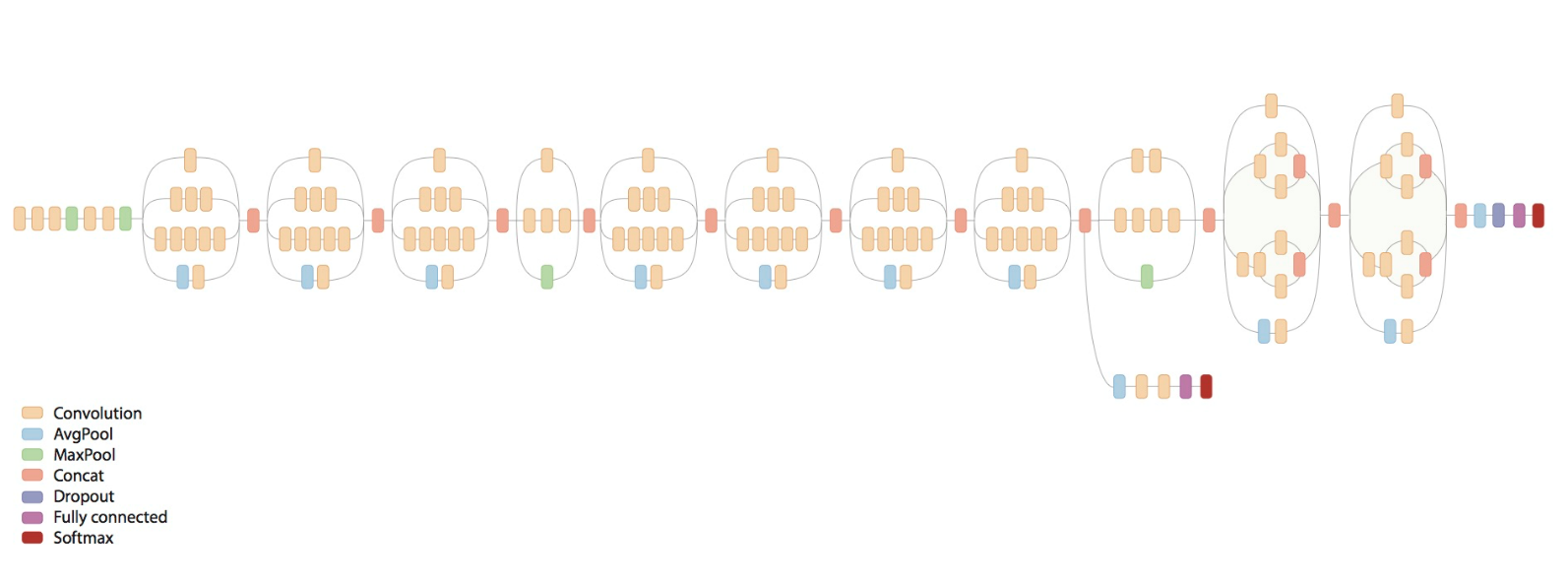
***Team Info***

Team Name: ggwp  
Team Members:   
1. Ang Xuan Yin Joel (1001075)  
2. Sng Han Jie (1001074)  
3. Muhammad Hafiz (1000944)  
4. Huang Jing Jie (1001309)

***Model***

We used the Inception v3 model (model architecture shown below). The model was pretrained on the ImageNet Large Visual Recognition Challenge using data from 2012. We performed retraining on the last layer of the model so that it is able to classify benign and malignant cancer images. The model was trained using 4000 epochs with batch size of 100 and with a learning rate of 0.01. Data augmentation was performed on training and validation sets. Similarly, we performed augmentation on test images and averaged over the predictions for the augmented images to obtain the final prediction for each test image.



***Instructions***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Description** | **File** | **Cmd (first cd to folder containing these scripts)** | **Output** |
| Extract Images | Extract images using txt files and classifies them by their labels for training purposes. | extractimages.py | python extractimages.py <magnification> <split>  e.g.  python extractimages.py 100X split1  (ensure that images are located in same directory of this project folder. i.e. “breakhis\_data/train\_val\_test\_60\_12\_28/non\_shuffled/…” is in the same folder as this project folder. | Generates two subfolders with images classified under their labels (benign and malignant)  These two folders will then be placed in a parent folder named <magnification>\_<split>, e.g. 100X\_split1.  Names of images will by appending with \_train, \_val or \_test respectively. |
| Augment Images | Augment both train and validation sets in the image folder. | augment\_images.py | python augment\_images.py <image\_directory>  e.g. python augment\_imges.py 100X\_split1 | Generates augmented images of train and validation images in the benign and malignant folders. |
| Retrain Inception model | Retrains the last layer of the Inception V3 model using train and validation images in the image directory. | retrain.py | python retrain.py \  --bottleneck\_dir=<bottleneck directory> \  --model\_dir=<model directory> \  --summaries\_dir=<summaries directory> \  --output\_graph=<output graph name> \  --output\_labels=<output labels name> \  --image\_dir=<image directory>  e.g.  python retrain.py \  --bottleneck\_dir=bottlenecks \  --model\_dir=inception \  --summaries\_dir=training\_summaries \  --output\_graph=retrained\_graph.pb \  --output\_labels=retrained\_labels.txt \  --image\_dir=100X\_split1 | Generates an output network that has its last layer retrained to classify the new types of images. The important files that are required for prediction are output\_graph and output\_labels. |
| Testing |  | testing\_with\_augment.py and augment\_images\_test | python testing\_with\_augment.py <image directory> <output graph name> <output labels name>  e.g.  python testing\_with\_augment.py 100X\_split1 retrained\_graph.pb retrained\_labels.txt | Generates an output file named test\_output.txt that contains accuracy of model by prediction on test images in image directory. |

***Results***

Magnification: 100X

|  |  |
| --- | --- |
| **Split** | **Accuracy on Test set (rounded to 2 d.p.)** |
| 1 | 86.48% |
| 2 | 82.59% |
| 3 | 85.22% |
| 4 | 79.71% |
| 5 | 82.89% |
| Average | **83.38%** |