Breast cancer histopathology detection using an end-to-end deep-neural-network.

Introduction:

Breast cancer starts when cells in the breast begin to grow out of control. These cells usually form a tumor that can often be seen on an x-ray or felt as a lump. The tumor is malignant (cancer) if the cells can grow into (invade) surrounding tissues or spread (metastasize) to distant areas of the body. Breast cancer occurs almost entirely in women, but men can get breast cancer, too.[1] In the United States, breast cancer is the most commonly diagnosed type of cancer and the third leading cause of cancer-related deaths among women.[2] In the U.S., the five-year survival rate for breast cancer is 89.7%.[3] An early diagnosis and treatment of this cancer can significantly increase the chance of survival. For the purpose of early diagnosis, proper histopathological analysis of tissue samples is very important. During the diagnostic work-up, suspected tissue samples are evaluated for both overall and region specific tissue organization using whole-slide and microscopic images respectively. However, the large amount of data and complexity of these images make this task time consuming and non-trivial.[4] Therefore, the development of an automated breast cancer histopathology classification tool is essential for the field, to further improve the breast cancer survival rates.

Methodology:

- Convolutional neural network using the Incpetion version 3 architecture.[5]
- Transfer learning using model weights trained on the Imagenet dataset.[6]
- End-to-end deep-learning architecture (from feature extraction to final classification).
- Added stacked fully connected dense layer architecture on top of the Inception layers.
- A set of custom drop-out layers to reduce over-fitting.
- The training data (N=400) was subjected to image augmentation.
- Resultant 16x increase in sample size.
- Augmented data (N=6400) split into 5600 images for training (87.5%).
- The remaining 800 images were used for validation (12.5%).
- Final accuracy, sensitivity & specificity, calculated using a separate test data (N=32).[7]

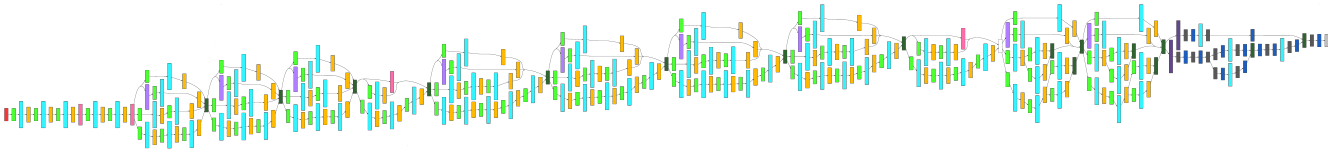

Figure 1: Overview of the Inception version 3 network and the multi-layered, branching perceptron architecture at the top. The top layer architecture was optimized for solving the breast cancer histopathology classification task.

Author : Dr. Rahul Remanan
Contact : rahul@nanoso.me

Results:

Peak training accuracy (%)	Peak validation accuracy (%)
81.50 (Batch size = 32, N=5600)	92.01 (Batch size = 32, N=800)
Least training loss function	**Least validation loss function**
1.350	0.4580

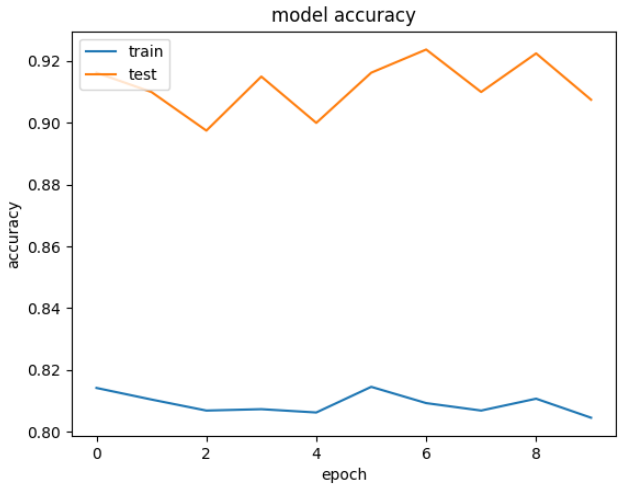

Figure 3: Plots showing the training accuracy and validation accuracy for the final 10 epochs of the neural network training session.

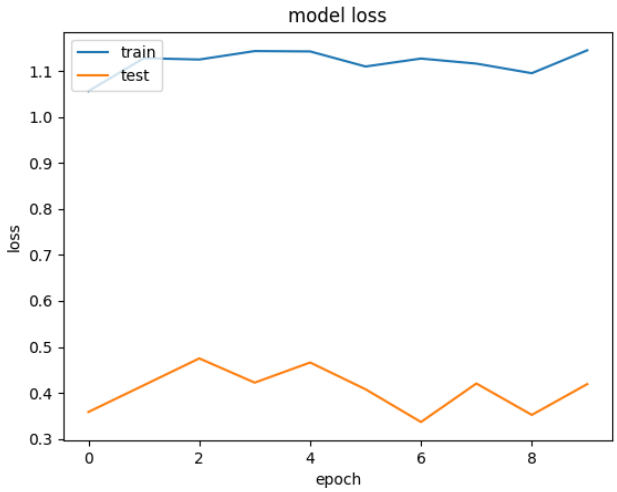

Figure 4: Plots showing the training and validation loss function for the final 10 epochs of the neural network training session.

Classification Accuracy (%)	Sensitivity (%)	Specificity (%)
88.88 (32/36)	94.44 (17/18)	83.33 (15/18)

Conclusion:

An end-to-end deep-neural network based on the Inception version 3 architecture was successfully trained using transfer learning, to classify breast cancer histopathology images. This model achieved a classification accuracy of 88.89%, sensitivity of 94.44% and specificity of 83.33% when tested using a dataset (N=32) that was not exposed to the model during training. The model described here has achieved a better accuracy, sensitivity and specificity than a previous approach, where, Araújo et.al combined a convolutional neural network feature extractor with a support vector machine classifier.[8] Both models: the older heterogeneous architecture by Araújo et.al and our newer end-to-end deep-learning approach, were tested against the same test dataset. An increased classification performance

Author : Dr. Rahul Remanan
Contact : rahul@nanoso.me

demonstrated by our approach is due to the use of a state-of-the-art, highly optimized, end-to-end deep-learning network.

Instructions to access the API for independent testing:

Contact the author to receive a testing server id and schedule a testing window. Before the images can be submitted to the server, the test user need to provide an IP address of the machine used for testing the model. The IP address serves as a layer of authentication to prevent unauthorized access to the test server. Once the test user receives a server id, images can be submitted using the three examples below. These examples assume that the test user is familiar with a unix terminal. For optimum results, the author recommends using Ubuntu Linux 17.10.

(Note: The images need to be submitted in either jpg or png format.)

Example 1 – Submitting a single local image file to the test server:

curl -F 'file=@/home/remanan_rahul/notebooks/HIMA/examples/ICIAR_2018/test/18.jpg' http://server_id

Example 2 – Submitting a single web image to the test server using Google API:

http://server_id/goo.gl/kjRAUf

Example 3 – Batch submit an entire folder of images to the test server:

for f in ./*.jpg; do curl -F 'file=@$f' http://server_id; echo "Predictions for $f"; done

(Note: Do not submit the batches using parallel threads. The test server supports only one API query at a time.)

Author : Dr. Rahul Remanan
Contact : rahul@nanoso.me