Title: Classification of Tumor Types from Histological Structures in Breast Biopsy Image Dataset

By: Alie Antar

Project Option: Research

Date: December 9th, 2022

I. Introduction/ Background

The burden of classifying tumor types from tissue samples can be a daunting task for healthcare professionals around the world. With the advancement of technology, and an increase in research in the field of machine learning, this burden can be lifted. In this paper, I implement a convolutional neural network (CNN) that attempts to accurately classify tumor types from breast tissue biopsy samples as either benign or malignant. Cancer is the uncontrollable growth of cells during the cell cycle. It typically occurs when cells fail to commit apoptosis (programmed cell death). Cancer cells can spread (metastatic) to other parts of the body and are able to trick the immune system into keeping them alive [1]. Early and accurate detection of these cancer cells can mean a drastic change in the outcome of patients. Convolutional Neural Networks (CNNs) utilize the technique of convolution to extract important information from an image, which are then stored in what's known as a feature map. These are then fed through the layers that define the model.

II. Data

The data obtained for this paper is from a Kaggle dataset [2]. This is a sample from a dataset that was obtained from a restricted database, known as Breast Cancer Histopathological Database (BreakHis) [3]. There are 9,109 breast tumor tissue images from 82 patients with the following magnifications: 40X, 100X, 200X, and 400X. The data with dimensions 700 X 460 pixels, 3-channel RGB, 8-bit depth in each channel, and PNG format, is split into a total of 2,480 benign tumors and 5,429 malignant tumors. The train data provided by Kaggle contains 371 benign tumor samples, and 777 malignant tumor samples; while the test data contains 176 benign tumor samples, and 369 malignant tumor samples. Clearly, there is a class imbalance that was difficult to remedy since access to more data is restricted. Since the data was fixed to training and testing, the only preprocessing that was needed was creating a validation set. From the 1,148 images in the train data, I saved 85% of those for training and 15% for validation. To get the data ready for use, the Keras Image Data Generator was used, and the training data was augmented by applying horizontal flips to create some diversity in the data. The batch sizes that were 8, 16, and 32.

III. Methods

The model is trained on the dataset described above and is used to determine if a tumor sample is malignant or benign. This implies a binary classification problem that was approached using a sequential CNN. All convolution layers consisted of 3x3 kernels; all max pooling layers consisted of a 2x2 pool size. All convolutions and all but the final dense layer, used a Rectified Linear Unit (ReLu) activation function. The layers that defined the model consisted of a convolutional layer with 32 filters, and the input of the image dimensions of (460, 700, 3). This was then followed by another convolution with 64 filters. A max pooling layer was then applied to extract the prominent features in the image to create a feature map. Another convolution was applied with 128 filters. This was then followed by another max pooling layer. One final convolution is applied with 256 filters. Max pooling was performed again to extract those high features. Then, a dropout layer with a dropout rate of 25% was applied followed by a flattening layer. After flattening, a dense layer was applied with the unit parameter set to 64, and another dropout layer with a rate of 25% was used. Finally, a dense layer with the units parameter set to 1 (because we're outputting either a 0 or 1), and a sigmoid activation function was applied. The metrics collected in the process were accuracy, precision, recall, true positives, false positives, true negatives, and false negatives. The loss function used was a binary cross entropy loss function, with the Adam optimizer.

Furthermore, early stopping callbacks and checkpoints were defined to prevent the model from continuing training if no improvements are made, thereby preventing the model from overfitting. The validation accuracy was used as the monitor for improvements or lack thereof, with a patience of 5 (i.e., if the model doesn't improve after 5 epochs, save the best-performing model). The model was then fit on the training and validation data for 30 epochs. Then, I loaded the model and tested it on the test data, while saving reports for each trial of the train and test data. In addition, I trained, validated, and tested the data using transfer learning on three different convolutional neural network models obtained from TensorFlow Hub. These models were Inception ResNet V2, Efficient Net V2, and Inception V3. The results for all the methods described will be outlined and discussed further into the paper.

IV. Results

Initially, I trained the model with the layers defined above with a batch size of 32. The model achieved the highest training accuracy of 83.90%, and a loss of 0.45 at epoch 6. The highest validation accuracy achieved by the model was 87.86% with a validation loss of 0.41 at epoch 8, with a precision of 0.91, and a recall of 0.91. For the test data metrics, the model achieved an 82.39% accuracy value, a 0.49 loss value, a precision of 0.84, and a recall of 0.92.

Next, I trained the model with a batch size of 16. It achieved the highest training accuracy of 87.08%, with a loss value of 0.38 at epoch 13. At epoch 9, the model achieved its highest validation accuracy of 89.6%, a loss of 0.38, a precision of 0.92, and a recall of 0.93. For the test data, it achieved an accuracy of 86%, a loss of 0.42, a precision of 0.88, and a recall of 0.91. For the next trial, I trained the model with a batch size of 8. The highest accuracy it was able to achieve was 76.41% with a loss value of 0.53 at epoch 10. It achieved its highest validation accuracy of 83.82% with a loss value of 1.00 at epoch 5. At this epoch, it achieved a precision value of 0.81, and a recall value of 0.99. For the test data, the model achieved an accuracy of 82.02%, a loss of 0.99, a precision of 0.80, and a recall of 0.97. From these results, it is clear that training with a batch size of 16 created the best results since I was monitoring for validation accuracy. However, this isn’t the only metric that will be used in comparison with other classification models.

V. Comparison

For this section, all models were trained with a batch size of 16 to keep it consistent with the model that I built, that achieved the highest validation accuracy. I first trained the data using the Inception ResNet V2 (IncResV2) model [4]. This model achieved the highest training accuracy value of 89.95% with a loss value of 0.26 at epoch 20. It achieved its highest validation accuracy of 91.91% with a loss of 0.25 at epoch 16. For the validation at this epoch, the model achieved a precision value of 0.91, and a recall of 0.97. For this model, the test data achieved an accuracy of 86%, a loss of 0.33, a precision of 0.88, and a recall of 0.92. Next, I trained on the Efficient Net V2 (EffNetV2) model [5]. This model achieved its highest training accuracy of 88.72% with a loss of 0.27 at epoch 9. For the validation, it achieved its highest accuracy of 86.13% with a loss of 0.34 at epoch 4. At this epoch, it obtained a precision value of 0.88, and a recall of 0.92. When using the test data, the model achieved an accuracy of 84%, a loss of 0.36, a precision of 0.85, and a recall of 0.93. Finally, I trained the data on the Inception V3 (IncV3) model [6]. This model achieved its highest training accuracy of 88.92%, with a loss value of 0.29 at epoch 8. For the validation, it obtained its highest accuracy of 92.49% with a loss value of 0.27 at epoch 6. This was the best performing model in terms of validation accuracy. It outperformed the model that I built, and the pre-trained models. At this epoch, the model achieved a precision of 0.91, and a recall of 0.98. For testing, it achieved an accuracy of 88%, a loss of 0.33, a precision of 0.88, and a recall of 0.96. The results from the best performing model for each trial are summarized in Table 1. This includes the results from both the model I built, and the pre-trained models for the validation data. For simplicity, I will refer to my model in the table as *"Batch16"*.

	Batch16	IncResV2	EffNetV2	IncV3
Accuracy	89.60%	91.91%	86.13%	92.49%
Loss	0.38	0.25	0.34	0.27
Precision	0.92	0.91	0.88	0.91
Recall	0.93	0.97	0.92	0.98

Table 1- A Comparison of the Metrics for all the Models

From Table 1, we can see that IncV3 is the highest performing model when considering accuracy, along with a low loss of 0.27 (although not the lowest loss). The second highest performing model is the IncResV2 with an accuracy of 91.91% and a loss of 0.25. This is followed by the third highest performing model, which is ours with an accuracy of 89.60%, and a loss of 0.38. The fourth highest performing model is the EffNetV2 with an accuracy of 86.13%, and loss of 0.34. Figure 1 shows a visual comparison of the validation accuracies, and validation losses of the models used in this experiment.

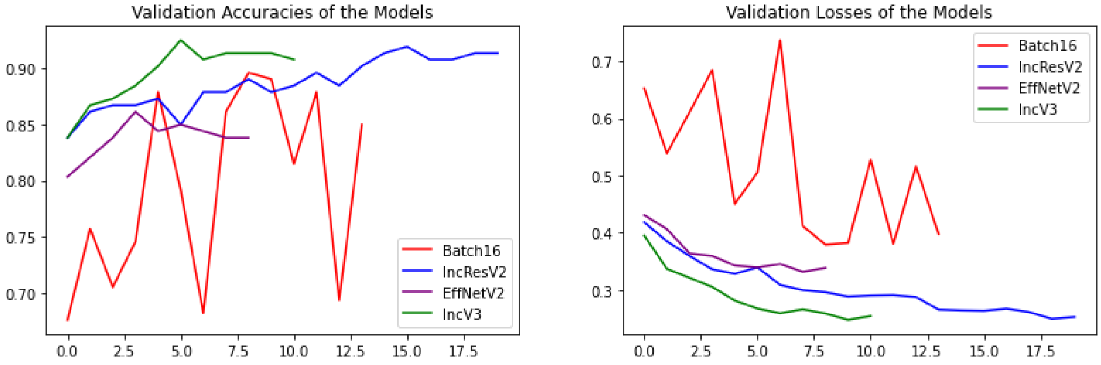

Figure 1- A Comparison of the Validation Accuracies and Validation Losses of the Models

VI. Discussion

It is worth noting that accuracy and loss are not the only metrics that define whether a model performs well or not. Initially, I based the strength of the models on the validation accuracy; however, this was not the only metric that should have decided how well our model was performing. To elaborate, since we are dealing with a class imbalance, it is good practice to focus on precision and/or recall in determining the best model [7]. Misclassifications are an inevitable outcome of predictive models; because we are dealing with a healthcare application, we would rather a benign tumor be misclassified as a malignant tumor rather than a malignant tumor be misclassified as a benign tumor. In other words, it’s better to be safe than sorry. Therefore, in our case where I would prefer to have misclassifications mostly occur in benign samples, I would aim for a better recall. This then changes the basis of the strength of our model. It puts training with batch size of 8 at a higher rank, since that was able to achieve a recall of 0.99. However, this would mean that the model would predict a majority of tumors as malignant, where the root issue of the class imbalance comes from. Therefore, the overall best performing model would still be IncV3 because it still achieves very desirable accuracy, precision, and recall values. In addition, we can observe from Figure 1 that our model, although produces a decent accuracy and recall value, it exhibits a lot of fluctuations. This could be an indication of our model being highly sensitive to variations in the data, or sensitive to noise. If we look at the pre-trained models, we can see a much steadier incline in the accuracy, and a much steadier decline in the loss. Furthermore, all the models produce a lower test accuracy when compared to training and validation, which could indicate some overfitting.

VII. Conclusion

Through trials, I was able to find a model that was able to produce fairly good results in terms of accuracy, loss, precision, and recall. Although the work was supposed to focus more on recall, the model was able to produce a good recall value of 0.93 despite monitoring for validation accuracy. For future improvement on this, I need to shift focus on this metric due to imbalance data; or just find more data that is accessible. In conclusion, we saw how several pre-built models are able to handle variations in data, and how that’s something worth considering for our model as well. Overall, the experiment was a big step in the direction of creating a good model, and definitely worth improving for future use.

VIII. Sources

[1] *What is cancer?* National Cancer Institute. (n.d.). Retrieved December 3, 2022, from https://www.cancer.gov/about-cancer/understanding/what-is-cancer

[2] Muzaki, K. (2020, July 29). *Breakhis 400X*. Kaggle. Retrieved October 21, 2022, from https://www.kaggle.com/datasets/forderation/breakhis-400X

[3] FA Spanhol, LS Oliveira, C. Petitjean and L. Heutte, "A Dataset for Breast Cancer Histopathological Image Classification," in IEEE Transactions on Biomedical Engineering, vol. 63, no. 7, pp. 1455-1462, July 2016, doi: 10.1109/TBME.2015.2496264.

[4] *Inception ResNet V2*. Tensorflow hub. (n.d.). Retrieved December 3, 2022, from https://tfhub.dev/google/imagenet/inception_resnet_v2/classification/5

[5] *Efficient Net V2*. Tensorflow hub. (n.d.). Retrieved December 3, 2022, from https://tfhub.dev/google/imagenet/efficientnet_v2_imagenet1k_s/classification/2

[6] Inception V3. Tensorflow hub. (n.d.). Retrieved December 8, 2022, from https://tfhub.dev/google/imagenet/inception_v3/classification/5

[7] Santos, M. (n.d.). *Precision or recall: Which should you use? | towards data science*. Retrieved December 6, 2022, from https://towardsdatascience.com/explaining-precision-vs-recall-to-everyone-295d4848edaf