Breast Cancer Diagnosis through Image Classification – Scientific Report

Michal Wojcik

Simon Schumacher

Virgil Baclanov

Coordinated by **Mauro Castelli** and **Yuriy Perezhohin**

NOVA Information Management School

Index

1. Introduction and general description

Breast cancer is the most prevalent cancer worldwide, killing roughly 600.000 women annually [1]. And with this grim trend on the rise in the last few years [2], it is more important now than ever before to identify cancer in time to be able to combat it for as many patients as possible. This will be the main motivation of the present project.

Time is a critical resource in such a complex undertaking. And this is even more challenging when the quality of the outcome must be kept high. With this in mind, a compact yet efficient pre-trained convolutional neural network, DenseNet121, was chosen and fine-tuned.

Since there are not many detailed samples taken under the microscope and with some types of cancer more prevalent than others, the given images in the dataset were augmented. It turned out to be beneficial in detecting fine details in such a nuanced disease.

Finally, reliability and accessibility had to be accounted for. Training a neural network on more than 7000 samples is a computationally intensive and error-prone task. Using additional computing power is expensive, so a memory-efficient approach must be found. An optimized dataflow in *Tensorflow* was created, guaranteeing fast training times and no crashes, all with no costs involved for using extremely powerful GPUs.

2. Experimental setup

To be able to leverage powerful GPUs to run our files, Google Colab and its library were used. The best models were always pushed in a remote repository as a checkpoint.

The notebook *Preprocess_csv_and_build_file_structure* is responsible for creating the .csv file that will be fed into the model as a DataFrame, and the file structure required for the model. For convenience purposes, the generated .csv file will be attached to the delivery.

The files *Best_binary_model* and *Best_multiclass_model* are meant to be run independently. **Note that the file paths in the code are absolute and must be changed to ensure reproducibility.**

The first step was manipulating the DataFrames, for which *Pandas* were used. Then, *sci-kit learn* facilitated the preprocessing of the DataFrames so that they can be fed into the model. It also was critical for splitting the dataset, computing class weights and, finally, evaluating the model through the confusion matrix. For creating the desired file structure, we create a directory with *os*, use parallel processing with *concurrent.futures* for faster execution, then read, resize and save the images with *PIL*. For the progress bar, *tqdm* was used.

Since we initially ran into memory problems, we had to come up with a clever workaround to use limited computational power. Thus, we temporarily stored the data by copying it fast in the temporary folder with the library *shutil*.

The augmentations of the images could not have been done easier without *Tensorflow*. For the creation of the model, *Keras* alone proved to be the best choice, because it was simple and yielded good results. It was used for everything model-related, from importing the base pre-trained model to defining callbacks. Visualizing the results was done with *Matplotlib* and *Seaborn*.

3. Approaches and final model

Starting with the *.csv* file, unnecessary columns were dropped, relative paths were added to the file names and labels were encoded. Moreover, missing values were checked for and filled accordingly based on the given file name.

The first issue with the given data set was that the original images were too big to be loaded into the memory at once. There were 4GB of raw data that had to be bulk loaded into a *NumPy* array, ultimately requiring more than 16GB of RAM at a given point in time. This led to memory crashes, stalling the continuation of the project. After many trials of scaling down the images had been conducted, it was agreed that the resolution of 256 x 256 would not lose too much information, while solving the memory dump problem and drastically cutting down on loading times. Moreover, the squared ratio makes data augmentation easier.

After it had been discovered that the training times were still too slow to evaluate simple models in a timely manner, methods were found to mitigate the negative effects of in-memory loading. While the *ImageDataGenerator* yielded better results, it still proved to be too slow to be able to evaluate the model for more than a dozen epochs, especially when adding basic augmentation procedures, like rotating all the images by 90 degrees.

Creating a dataset with *Tensorflow* showed drastically better results. With it, the data is loaded faster in batches and prefetched asynchronously to ensure parallelism. The dataset was split into 70% train, 15% validation and 15% test. To further optimize the performance and facilitate the processing of the images later in the same session, the data was temporarily stored in the */tmp* directory, which makes its retrieval much faster than over the network, in cloud. The code for creating the definitive file structure of the dataset was also created.

Upon observation of the confusion matrix after running a very primitive convolutional neural network on the training dataset, the unbalanced distribution of the classes became evident. The number of images of the benign class is less than half the number of images of the malignant class, meaning that the model tends to predict the latter more often, while not receiving high losses. To combat this issue both in the binary and multi-classification problems, class weights were used to penalize the majority class. Another very important step was augmenting only the minority class with a simple flip, effectively doubling its training instances. Stratification was also added to the train-test-split based on the target labels.

Because the data is unbalanced, the F1-score would be the more accurate metric to base the model on. However, since the F1-score cannot be directly displayed during training, trials were made for an in-house F1-score calculation to be hardcoded. However, it proved to be unreliable,

and, in the end, it was agreed upon using progress diagrams with accuracy and loss for the training and validation data. Due to shuffling and batching, accuracy can be jumpy at times. For testing the model, the F1-score is reliably calculated using *sci-kit learning*. Classification reports and confusion matrixes are used on the test data to confirm that the model is working well.

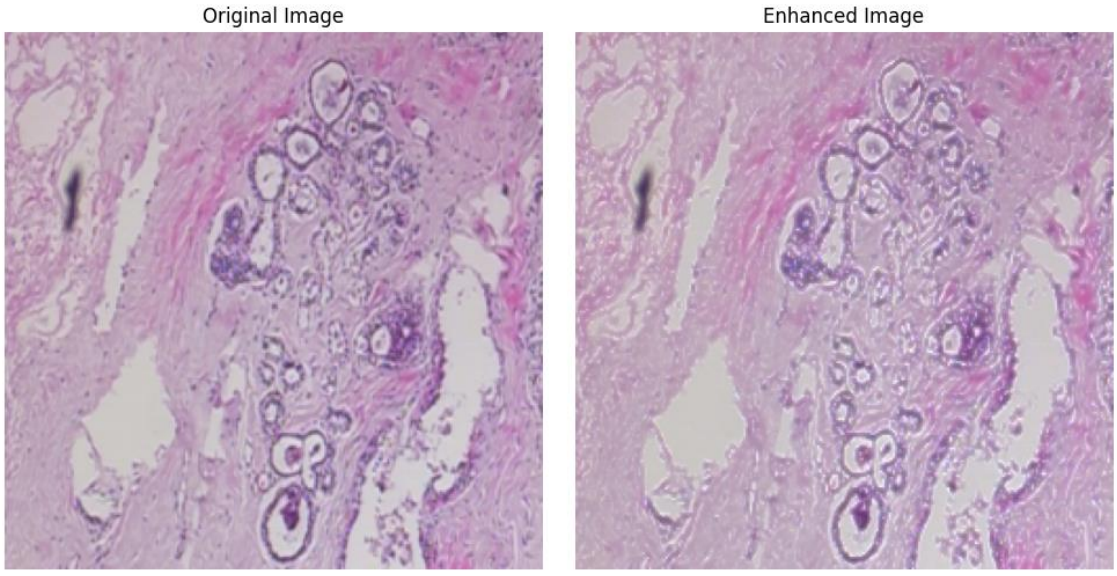
For the training, the all-around fitting ADAM optimizer was used, along with two different callback systems, namely *Earlystopping* and *ReduceLRonPlateau*, which cuts the learning rate in half every time the model starts making little evaluation progress. They both enabled us to faster detect overfitting or poor-performing models.

The primitive multilayer convolutional neural network with the above-mentioned configuration, which served as our base model, posted an F1-score of 85% for the binary classification problem. Other own architectures were implemented, but the F1-score could not top 85%. Additionally, the training times were too high and the desire for better F1-scores led to the decision to implement pretrained models.

The conversion of the *CERBERUS* [3] model from the *PyTorch* framework to *Keras* did not work. Other different pretrained model architectures with frozen layers were tested in *Keras* directly: *EfficientNetB0*, *ResNet50*, *VGG19*, *MobileNetV2*, *NasNetMobile*. However, none of them performed as well as DenseNet121. It posted F1-scores of 91% and 70% in the binary and multi-classification task, respectively.

To further try increase F1-scores, the Reinhard color normalization technique [4] [5] was applied by uniformizing colors over all images based on one reference image from the dataset. However, it is probable that color variation plays an important role in distinguishing between the present breast cancer types, since validation scores decreased. Another explanation could be that the reference image was chosen poorly.

The Laplacian edge detection also yielded worse results. This could boil down to the fact that extremely small cancer spots disappear upon applying the filter, thus losing potentially crucial information (see diagram below). Gray scaling the images led to lower scores as well.

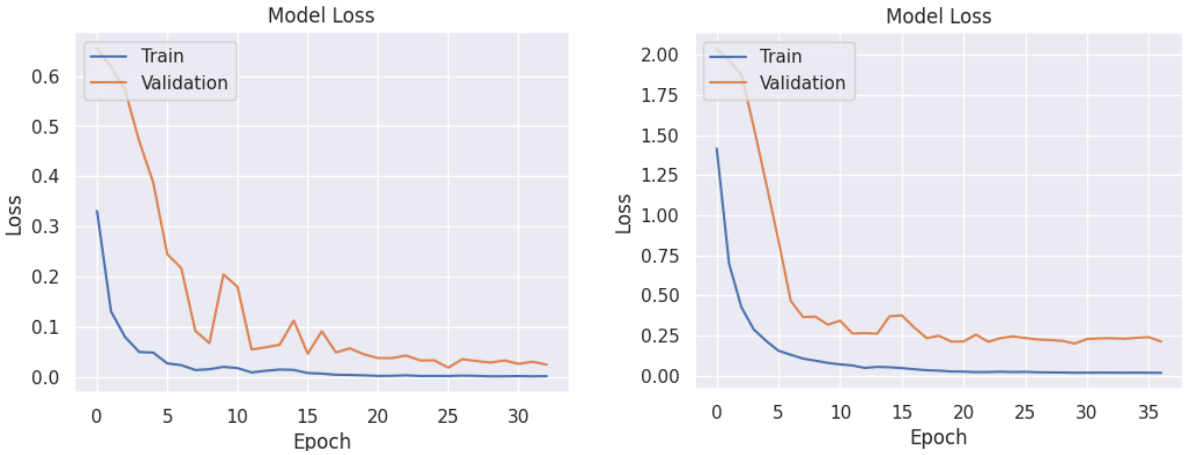


The F1-score saw a tremendous improvement when the layers of the pretrained model were unfrozen, and the initial weights kept in place. This makes sense, because the data to be classified, histological tissue, is different than the data ImageNet was trained on, namely objects observable on a macro-scale. However, these changes caused overfitting, for which solutions have been searched.

Unfreezing only 3 layers still resulted in overfitting and similar scores as to when all layers were unfrozen. However, L2-regularization and the addition of batch normalization and dropout layers after the fully connected layer slightly improved the situation. Likewise, random flips were also beneficial. Augmenting the images with randomly adjusted saturation, brightness and sharpness was to no considerable avail.

The final, best-performing model is a custom, pretrained DenseNet121 with all the layers unfrozen. It was cut after the last convolutional block, with one batch normalization and one dropout layer having been added before flattening. The effective above-mentioned changes for combatting overfitting were kept in place.

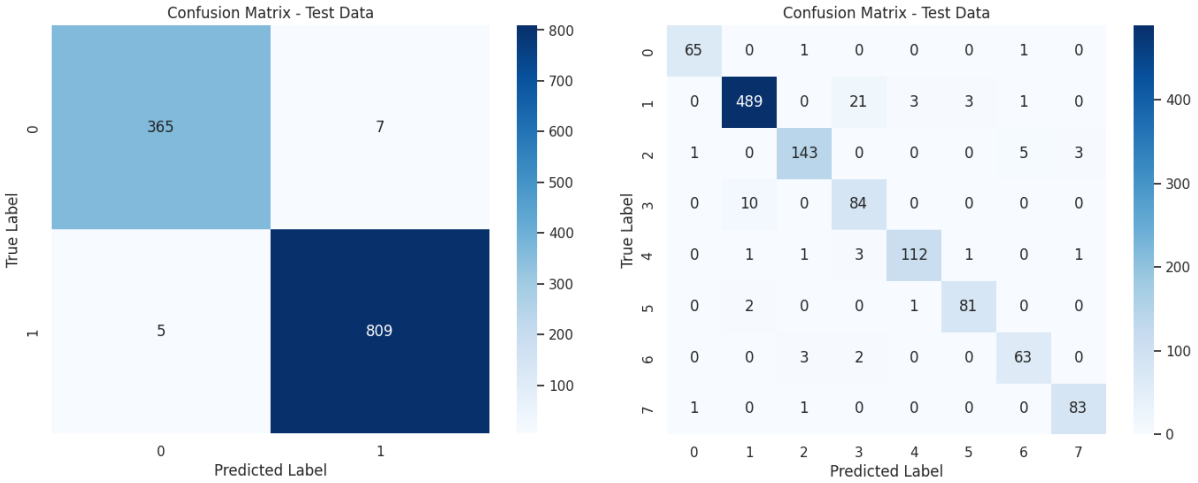
For binary classification (left-hand side), the model had an F1-score of 99% on both the validation and test data, with a training score of 100%. For the final multiclass model (right-hand side), the F1-score was 94% on the validation and test sets, with 98% on the training set.



In the end, with the hope of further boosting the F1-score, and since magnification was not taken as an input for the previous models, a similar multi-input model was created and put to test. However, with an F1-score of just under 70%, the results were not satisfactory. This could be because the magnification does not correlate with the type of cancer that is to be seen in the images. Therefore, the redundant text representing the magnification confuses the model and degrades its performance.

4. Results summary

The binary classification model has an F1-score of 99%, while its multi-classification peer achieves an impressive 94%.



For the multi-classification model, the biggest challenge was determining the differences between *Ductual Carcinoma* and *Lobular Carcinoma*. The precision score for the *Lobular Carcinoma* subclass was the lowest out of all the subclasses, because the model was oftentimes mistaken in classifying it as *Ductual Carcinoma*. Further work should focus deeper on finding out exactly why this happens and implementing an adequate augmentation technique. Harsher penalization of *Ductual Carcinoma* as majority class might have diminished the problem.

5. Future work

Firstly, to push the limits of the resulted model and further contribute to advancing cancer detection, it is critical to bridge the gap between biological and technological expertise. *Advanced interpretability tools*, such as *DeepSHAP* [6], allow doctors to understand actionable insights by making complex neural networks explainable. A complex approach rooted in genetics that could further improve breast cancer detection is *Multimodal Data Integration* [7]. It combines imaging data with clinical information, such as patient history or genetic profiles, to create a comprehensive diagnostic model and uncover meaningful correlations between imaging features and clinical characteristics.

Secondly, on the more technical side, exploring the *Next-Gen Architecture* [8], with hybrid models that combine the strengths of CNNs and Transformers [9], could leverage both local and global feature extraction capabilities. To discover the most useful and effective augmentation operations for the given dataset, Automated Data Augmentation techniques such as *MedAugment* [10] and *RandAugment* [11] might also be beneficial. The conversion of the promising *ADOPT* [12] optimizer into *Keras* could also impact the overall performance of the model.

Bibliography

[1] M. Arnold, E. Morgan, H. Rumgay, A. Mafra, D. Singh, M. Laversanne, J. Vignat, J. R. Gralow, F. Cardosoc, S. Siesling and I. Soerjomataram, "Current and future burden of breast cancer: Global statistics for 2020 and 2040," 02 September 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9465273/#:~:text=With%20over%202.3%20million%20new,most%20commonly%20diagnosed%20cancer%20worldwide.&text=Most%20cases%20occur%20in%20transitioned,share%20of%20breast%20cancer%20deaths..

[2] A. N. Giaquinto, H. Sung PhD, L. A. Newman MD, R. A. Freedman MD, R. A. Smith PhD, J. Star MA, P. A. Jemal DVM and R. L. Siegel, "Breast Cancer Statistics 2024," American Cancer Society Journal , 01 October 2024. [Online]. Available: https://acsjournals.onlinelibrary.wiley.com/doi/full/10.3322/caac.21863.

[3] C. Scribano, G. Franchini, I. Olmedo and M. Bertogna, "CERBERUS: Simple and Effective All-In-One Automotive Perception Model with Multi Task Learning," 03 October 2022. [Online]. Available: https://arxiv.org/abs/2210.00756.

[4] E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley, "Color Transfer Between Images," IEEE Xplore, 07 August 2002. [Online]. Available: https://ieeexplore.ieee.org/document/946629.

[5] S. Roy, S. Panda and M. Jangid, "Modified Reinhard Algorithm for Color Normalization of Colorectal Cancer Histopathology Images," 2021. [Online]. Available: https://eurasip.org/Proceedings/Eusipco/Eusipco2021/pdfs/0001231.pdf.

[6] H. Chen, S. Lundberg and S.-I. Lee, "Explaining Models by Propagating Shapley Values of Local Components," 27 November 2019. [Online]. Available: https://arxiv.org/abs/1911.11888.

[7] Z. Yao, F. Lin, S. Chai, W. He, L. Dai and X. Fei, "Integrating Medical Imaging and Clinical Reports Using Multimodal Deep Learning for Advanced Disease Analysis," 23 May 2024. [Online]. Available: https://arxiv.org/abs/2405.17459.

[8] S. Roy, G. Koehler, C. Ulrich, M. Baumgartner, J. Petersen, F. Isensee, P. F. Jaeger and K. Maier-Hein, "MedNeXt: Transformer-driven Scaling of ConvNets for

Medical Image Segmentation," 02 June 2024. [Online]. Available: https://arxiv.org/abs/2303.09975.

[9] R. Azad, A. Kazerouni, M. Heidari, E. Aghdam, A. Molaei, Y. Jia, A. Jose, R. Roy and D. Merhof, "Advances in Medical Image Analysis with Vision Transformers: A Comprehensive Review," 05 November 2023. [Online]. Available: https://arxiv.org/abs/2301.03505.

[10] Z. Liu, Q. Lv, Y. Li, Z. Yang and L. Shen, "MedAugment: Universal Automatic Data Augmentation Plug-in for Medical Image Analysis," 30 June 2023. [Online]. Available: https://arxiv.org/abs/2306.17466.

[11] E. Cubuk, B. Zoph, J. Shlens and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," 14 November 2019. [Online]. Available: https://arxiv.org/abs/1909.13719.

[12] S. Taniguchi, G. Minegishi, S. C. Jeong, T. Iiyama, Y. Iwasawa, K. Harada, Y. Oshima, G. Nagahara, M. Suzuki and Y. Matsuo, "ADOPT: Modified Adam Can Converge with Any β2 with the Optimal Rate," 22 November 2024. [Online]. Available: https://arxiv.org/pdf/2411.02853v3.