NOVA IMS
Information Management School

DEEP LEARNING PROJECT

BSc DATA SCIENCE 24/25

Breast Cancer Classification

Group 7

João Capitão | 20221863
Maria Rodrigues | 20221938
Nuno Leandro | 20221861
Vidhi Rajanikante | 20221982
Yehor Malakhov | 20221691

Abstract

This project aims to detect whether a tissue sample is malignant or benign and to predict the specific type of cancer, both, based on high-resolution microscopic images of breast tissue. To achieve this, we experimented with various models, applied different image preprocessing techniques, and optimized the best-performing models. The result was two final models: the first one, which predicts whether an image corresponds to malignant or benign cancer, achieved an F1 score of 0.90 on unseen data, while the second one, which predicts the specific type of tumor, achieved an F1 score of 0.83 on unseen data. We believe these results provide valuable insights into the detection of breast cancer, a critical health issue that deserves greater attention.

Introduction

Breast cancer is a major health concern and is the most diagnosed cancer among women in the U.S. Each year, approximately 30% of all newly diagnosed cancers in women are breast cancer (source: BreastCancer.org).

Due to its high prevalence, advancing medical research is crucial. For doctors to provide accurate diagnoses, it is essential to develop models to identify the type of cancer and determine whether it is malignant (cancerous and potentially spreading) or benign (generally not harmful).

Our objective is to create the most effective model for analysing medical cell images and predicting two key factors: whether the cancer cells are malignant or benign and the specific type of tumor.

To achieve this, we utilized Convolutional Neural Networks (CNNs). Our dataset, the BreaKHis (Breast Cancer Histopathological Database) – Laboratório Visão Robótica e Imagem, contains 7909 high-resolution microscopic images of breast tissue captured using the SOB method. The dataset includes four distinct types of benign breast tumors: *adenosis* (A), *fibroadenoma* (F), *phyllodes tumor* (PT) and *tubular adenoma* (TA); as well as four types of malignant tumors: *carcinoma* (DC), *lobular carcinoma* (LC), *mucinous carcinoma* (MC) and *papillary carcinoma* (PC).

The dataset comprises varying numbers of images for each tumor type and different magnifications for each specific tissue image (40X, 100X, 200X, 400X).

We applied various preprocessing techniques, image transformations, and CNN architectures, and explored transfer learning to improve model performance.

Exploration

Our first step was to organize the images into new folders that aligned with our objective. We created two main folders corresponding to the two stages of the project, each containing a structured hierarchy with subfolders for training, validation and testing, all stratified by the target variable. This organization allowed us to efficiently loop through the images or visualize them for each specific purpose.

The next step consisted of visualizing the images. We started by comparing the differences between benign and malignant cells to see if anything noticeable could be detected with the naked eye. By looping through some of the images, we observed some interesting patterns. Specifically, some images of malignant tissue appeared to have black dots[1], something that did not occur in the images of benign tissue. Although we did observe some shadows in benign tissue images[2], they were more greyish and less prominent.

It's important to note, however, that these observations were made by someone without medical expertise, so the presence of black dots may not be directly related to whether the tissue is benign or malignant.

We wanted to extract additional features from the images, so we proceeded with Texture Analysis using the Gray Level Co-occurrence Matrix (GLCM). The GLCM represents the spatial relationship between pixel values in terms of distance and angle within specific sub-regions of the image. This allowed us to visualize areas in the images with the greatest contrast[3].

Given the significant role of color in the images, the distribution of the 3 color channels (Red, Green and Blue) was also analysed. We compared these distributions between benign and malignant types[4] and across the different types of cancer[5].

Preprocessing

To prepare the images to feed into the models, some preprocessing steps were implemented. First, we looped through all the images and found that almost all of them had the same size (the images varied by 4 pixels in height). However, we still needed to resize the images to ensure that our machines would have enough RAM to process them. After some 'trial and error', we decided to resize the images to 128x128 pixels.

To improve the model's generalization, we decided to implement data augmentation. This way, the model would see new variations of images at each epoch, making it less likely to overfit. We experimented with different image transformations using both *ImageDataGenerator*[6] and TensorFlow's Image Processing API[7]. These augmented images were then fed into the model.

Images were then normalized, as neural networks typically perform better when input features are scaled. Data was also shuffled to ensure variation of training examples. Finally, the arrays were converted into tensors to prepare them for input into the model.

Model

The project was divided into two stages. The first stage aimed to predict whether an image corresponds to a benign or malignant type of cancer. The second stage focused on predicting the specific type of cancer.

In both stages, data was divided into training, validation and test sets. The training set was used to help the Artificial Neural Network, ANN, identify patterns, while the validation set evaluated the model's ability to generalize these patterns. To avoid data leakage, a completely independent test set was reserved, containing data unseen during both training and validation. A 60-20-20 split was opted for, which provided sufficient data for training whilst ensuring the validation and test sets were large enough to reliably assess performance and generalization.

Due to the large size of the dataset and the lack of resources, a Hold-Out method was also performed for its evaluation. We recognize that cross-validation is usually more robust but it was less feasible, given our circumstances.

We ensured stratification across all three sets because the classes were imbalanced. Specifically, there were 5429 malignant images and only 2480 benign images. This imbalance was also present in the multiclassification problem, where *ductal carcinoma* was by far the most frequent type of cancer. Without stratification, the model could produce biased results, favoring the majority classes while neglecting the smaller ones.

Stage 1

We initially aimed to create a baseline model, so we started with a simple *Sequential* model from Keras TensorFlow. This model consisted of just two convolutional layers followed by two dense layers. This setup allowed us to establish a baseline and measure the improvements made from that point onward.

Since this is a binary classification problem (0: benign, 1: malignant), we used a *sigmoid* activation function in the final layer and *binary crossentropy* as the loss function.

To evaluate our model, we considered two main factors: the plot of the loss value for both the training and validation sets, and the F1 score. The loss was monitored because it is tightly correlated to the training process, allowing us to track progress over epochs and identify any signs of overfitting (when training loss is much lower than validation loss). The F1 score was used as the evaluation metric for predictions on the test set due to the imbalanced dataset. The F1 score accounts for both precision and recall, ensuring that the model's performance on the minority class is not overlooked, unlike accuracy, which can be inflated by the majority class.

Additionally, the importance of recall in medical diagnosis is recognized as false negatives are generally more detrimental than false positives in medical contexts and recall helps minimize the false negatives.

Our initial model achieved an F1 score of 0.34 and was only predicting one of the classes. Additionally, the validation loss was lower than the training loss, so we decided to increase the number of dense layers and the number of epochs. Our approach also involved progressively increasing the dimensionality of the output in the convolutional layers while decreasing the number of neurons in the dense layers to enhance the model's learning capacity.

After implementing these changes and additional adjustments, such as incorporating *callbacks*, we achieved a model with an F1 score of 0.89.

The goal now was to increase the performance even more. To achieve this, we utilized Keras *Hyperband* to optimize the hyperparameters of the model. Unlike grid search, which exhaustively tests all possible combinations in the specified hyperparameter space, *Hyperband* is much faster. It intelligently navigates the search space, making it significantly more efficient than random search while still exploring a wide range hyperparameter combinations.

With the Hyperband search, we managed to improve the F1 score to 0.90 on the test set and achieved a higher recall, which, as previously mentioned, is a critical metric in this context. The slight increase in performance indicates that the hyperparameters used in the intermediary model were already quite good.

Following this, we experimented with data augmentation, specifically using *ImageDataGenerator* (adjusting brightness, rotation, zooming) and additional transformations. However, the results remained similar to those of the previous model.

We also experimented with Macenko Normalization, a method for normalizing and enhancing the contrast of microscopic images to improve the visibility of cell structures, but it also didn't give great results. An example of this normalization is provided here[8].

Transfer learning was then implemented using pretrained models, including VGG-16, ResNet-50 and Xception. Despite efforts such as freezing and unfreezing layers, the model did not learn efficiently as the loss value remained constant over the epochs.

As a result, we chose the final model without augmentation and transfer learning, which had been optimized through the Hyperband search since it had a great performance.

After developing the model, a question hovering in our minds was finally answered: Does the ANN focus on the black dots, observed initially by the naked eye, when classifying as benign or malignant? To answer this, we implemented the Grad-CAM visualization, which revealed what the model focuses on during classification. Interestingly, the model did indeed look at the black dots. The comparison between the images and the Grad-CAM visualization shows that the black dots, identified earlier, are more prominently highlighted in red tones, in the Grad-CAM visualization[9].

Stage 2

The second stage involved a multiclass classification problem. Given this, we used the softmax activation function for the last dense layer and sparse categorical crossentropy as the loss function. As in the first stage, a baseline model was created to assess performance. This initial model achieved an F1 score of 0.39.

Afterward, a Hyperband search was performed again, to optimize the hyperparameters. This time, BatchNormalization was also incorporated, which applies a transformation to maintain the mean output close to 0 and the output standard deviation close to 1 thus helping to stabilize the learning process and improve training efficiency. Transfer learning was also incorporated again, with the same three pretrained models mentioned earlier. The results were not ideal as a weighted average F1 score of 0.36 was achieved.

A different approach was then opted for. Functional API was implemented, taking advantage of the CSV file that contained the paths to the images and their corresponding labels (benign or malignant). We believed this could help the model by narrowing down the prediction task. If the model already knew whether an image was benign or malignant, the classification task for tumor type would be reduced to four types instead of eight.

The scores on the test data improved significantly, with a weighted average F1 score of 0.68. However, we encountered another problem: the model was overfitting. While the training loss dropped to 0.16, the test loss remained high at 0.92. To mitigate this, we reduced the batch size and incorporated dropout layers into the Hyperband search. These adjustments helped somehow but didn't resolve the issue.

What truly made a difference was when we stopped using transfer learning and pre-trained models. This was primarily because our dataset is relatively small, which increases the risk of overfitting when using pre-trained models. With limited data, the model might end up memorizing the examples rather than learning to generalize. Additionally, the pre-trained models used were not specifically designed for medical tissue images. As a result, the extracted features might not have been relevant or meaningful for our task, leading to worse performance.

By executing a new Hyperband search without the pretrained models, we achieved excellent results. We obtained a weighted average F1 score of 0.83, with only minimal overfitting. Considering that this was a multiclass classification problem with 8 classes, this was regarded as a significant achievement.

To push this score even further, the training and test sets were combined into a single dataset and the model was trained again. However, the F1 score dropped slightly to 0.74. This decline might be attributed to the fact that, although we had more data, the model’s hyperparameters were optimized for the original dataset size and the adjustment decreased its performance.

Results

In stage 1, the best results were achieved through a *Hyperband* search applied to our robust strong sequential model. The optimized model consisted of a neural network with 6 convolutional layers and 2 dense layers. The results and the model architecture are presented here[10]. The confusion matrix reveals that the false negatives are smaller than false positives, which is positive in a medical context, as missing malignant cases (false negatives) can have more severe consequences.

In stage 2, the best results were obtained using a *Functional API* with two inputs: the images and a CSV file containing information about whether each image was benign or malignant. The optimal hyperparameters for this stage were also found through a *Hyperband* search. The results and the model architecture are presented here[11]. The confusion matrix reveals that the model performs well for most classes; however, it struggles with the Phyllodes tumor and the Papillary carcinoma, with only 58 and 69 correct predictions, respectively, with many misclassifications across other classes. This suggests these classes are harder to distinguish for the model, possibly due to shared features with other categories.

Conclusion

By developing this project, a deeper understanding of the various approaches that can be utilized when building a Neural Network architecture was gained. Techniques such as transfer learning and the functional API were applied. Additionally, this project provided us with valuable experience in working with medical images, a crucial field in the present day and arguably one of the most important applications of AI. We learnt how to analyse and transform medical images and we also explored how deep neural networks perceive these images to detect patterns.

Future Work

With more time and resources for this project, a stratified k-fold crossvalidation would have been applied, as it is more robust for evaluating models. Additionally, a pretrained model, specifically trained on medical images, would also have been implemented. Although we searched for such models, we were unable to find one. Ultimately, the most commonly employed pre-trained models were used; however, their lack of specialization in our specific context made them ineffective. Further transformations would also have been explored, as the ones applied did not lead to a significant increase in performance.

Lastly, to further optimize the model, the Adopt optimizer, the latest released optimizer, would have been implemented, as we attempted to use but couldn't due to the compatibility issues between package versions.

References

- Breast Cancer Organization. (2024). *Breast cancer statistics.* BreastCancer.org. (Breast Cancer Facts and Statistics | Breast Cancer Organization)
- TensorFlow. (2023). *Keras Guide.* TensorFlow. (Keras: The high-level API for TensorFlow | TensorFlow)
- Hayes, J. (2007). *Image Classification: Gray Level Co-Occurrence Matrix (GLCM)*. Portland State University. Image Classification: Gray Level Co-Occurrence Matrix (GLCM) | Portland State University)
- Macenko, M., Niethammer, M., Marron, J. S., Borland, D., Woosley, J. T., Guan, X., Schmitt, C., & Thomas, N. E. (2009). *A method for normalizing histology slides for quantitative analysis*. University of North Carolina. (A Method for Normalizing Histology Slides for Quantitative Analysis | University of North Carolina)
- Keras. (2021). *Grad-CAM class activation visualization.* Keras. (Grad-CAM class activation visualization | Keras)
- Deep Learning Course Theoretical PowerPoints.
- Deep Learning Course Practical Notebooks.

Appendix

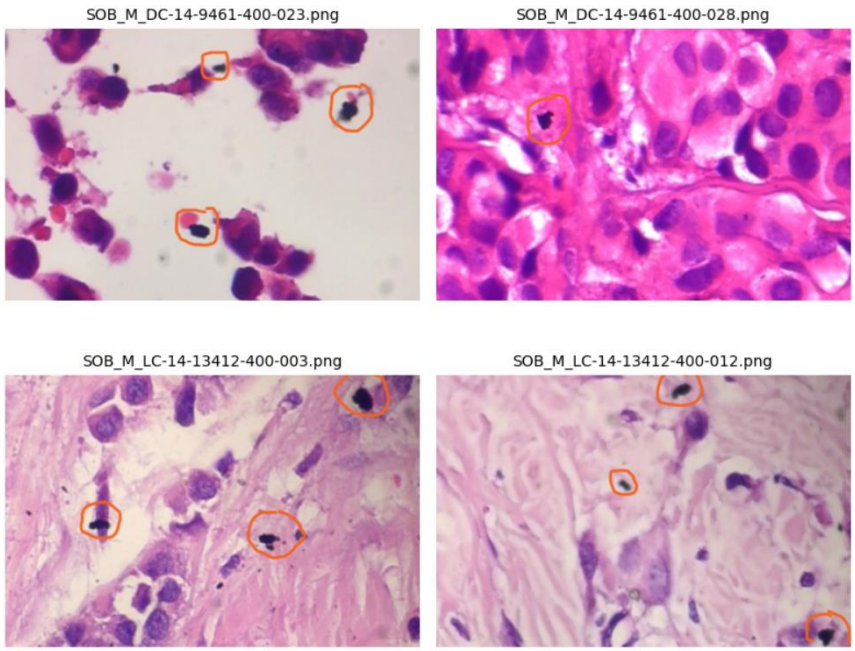


Figure 1: Black dots detected by us in certain images of malignant breast cancer

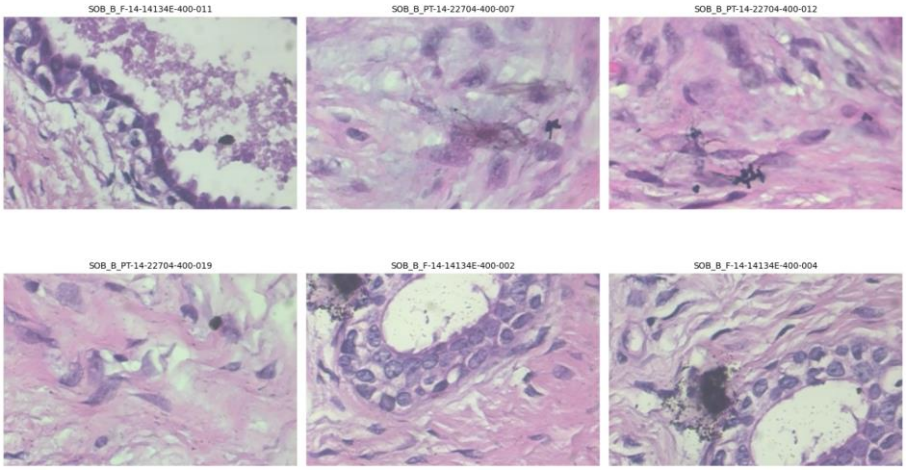


Figure 2: Grey shadows detected by us in certain beningn images

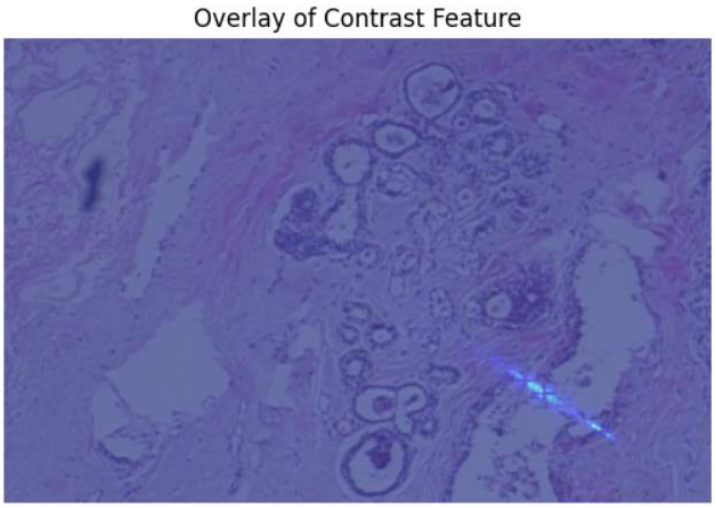


Figure 3: GLCM Contrast Feature Overlay on an example image

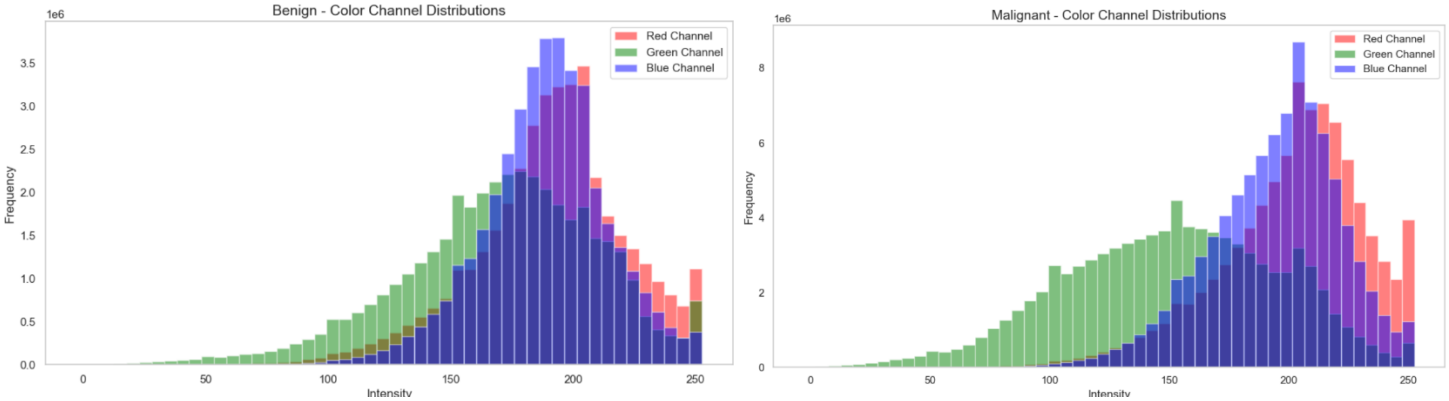


Figure 4: Color Channels (RGB) distribution of beningn and malignant images

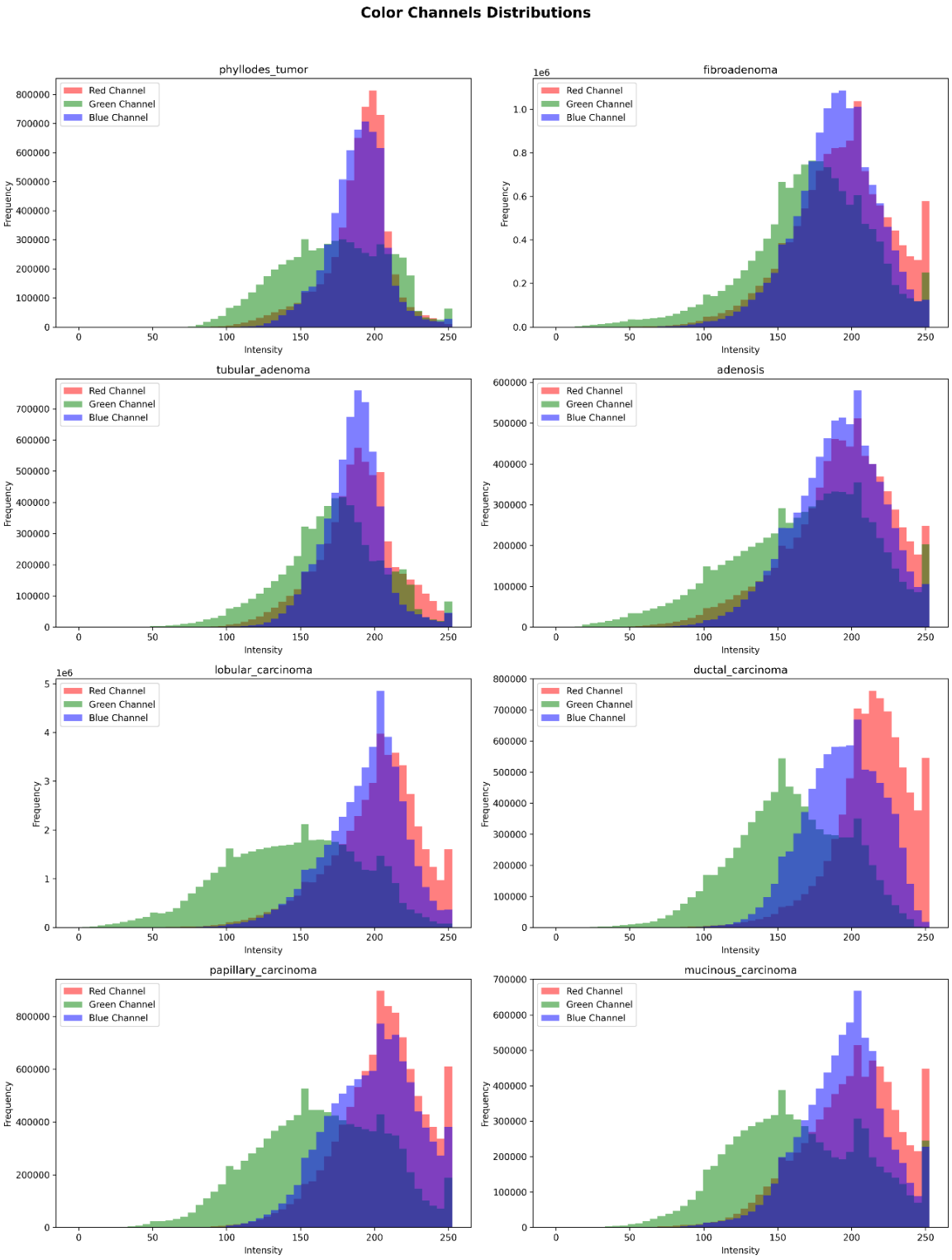


Figure 5: Color channels (RGB) distribution for the different types of cancer

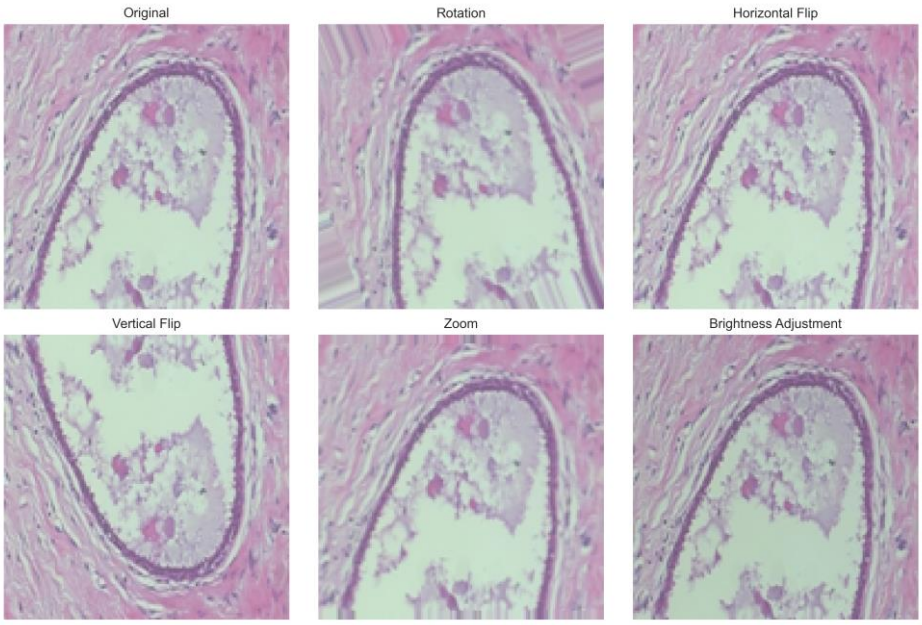


Figure 6: Some image transformations using ImageDataGenerator

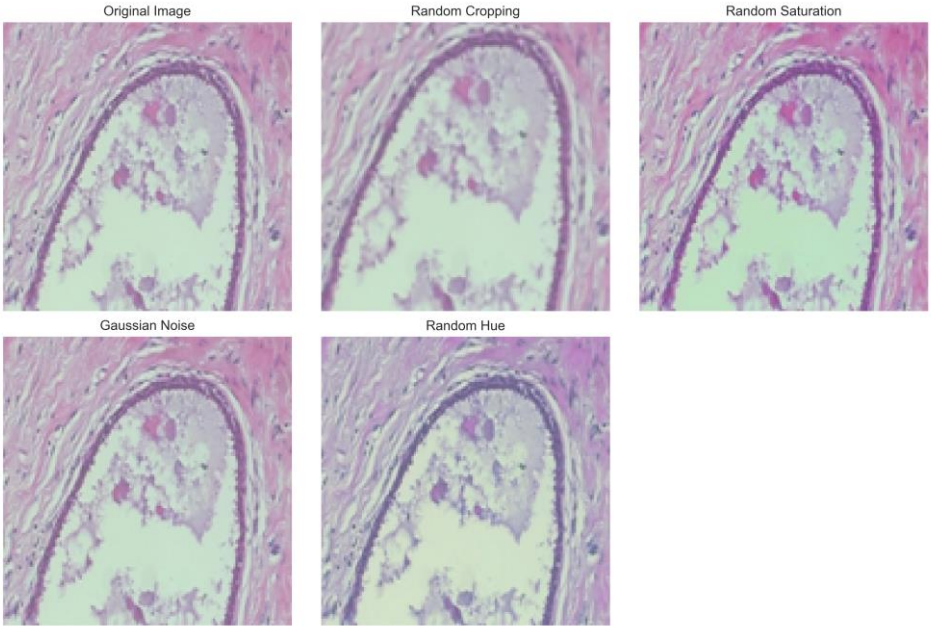


Figure 7: Some image transformations using TensorFlow's Image Processing API

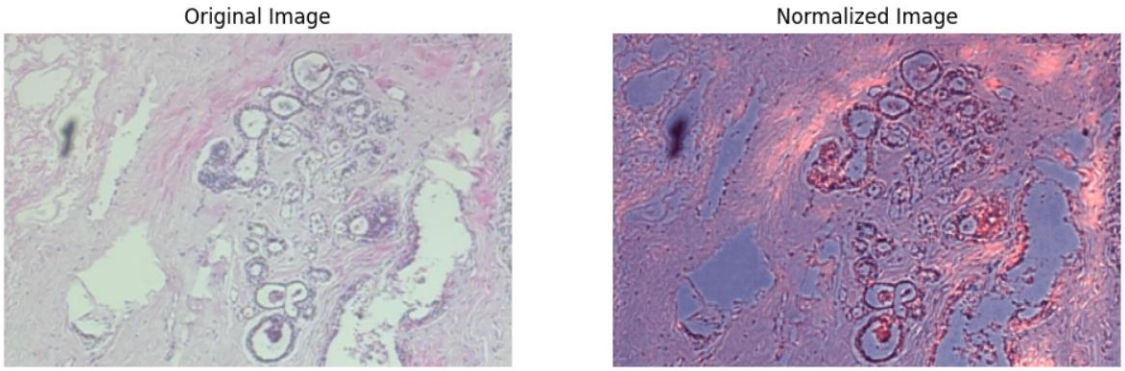


Figure 8: Example of Macenko Normalization

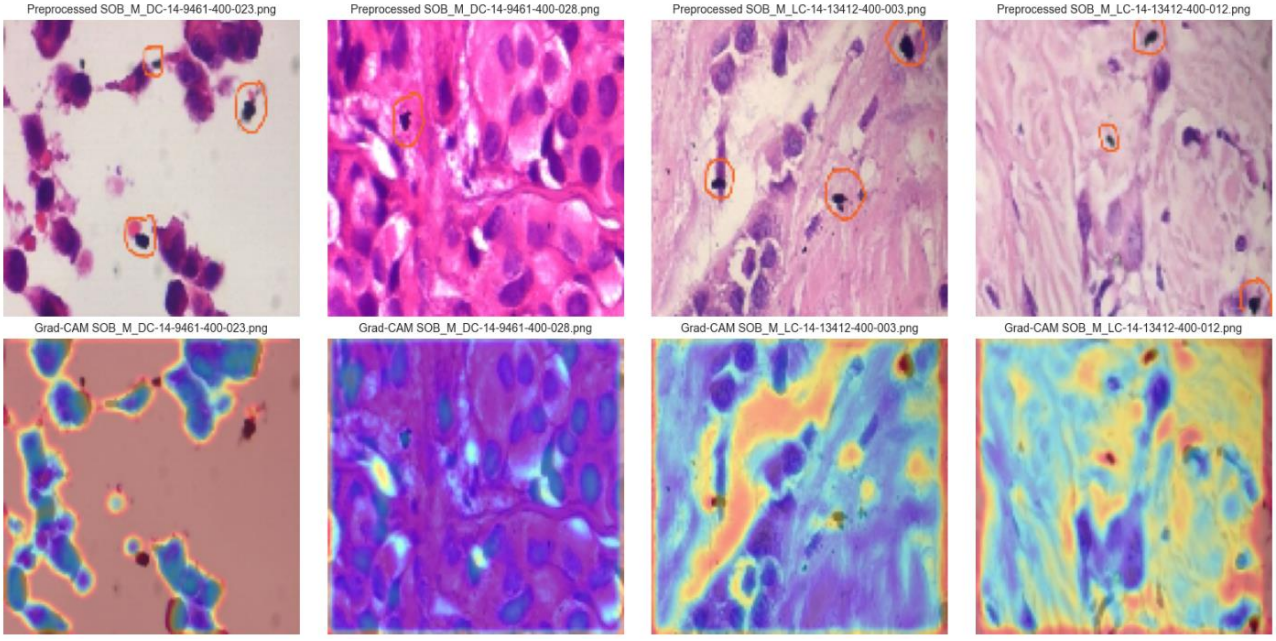
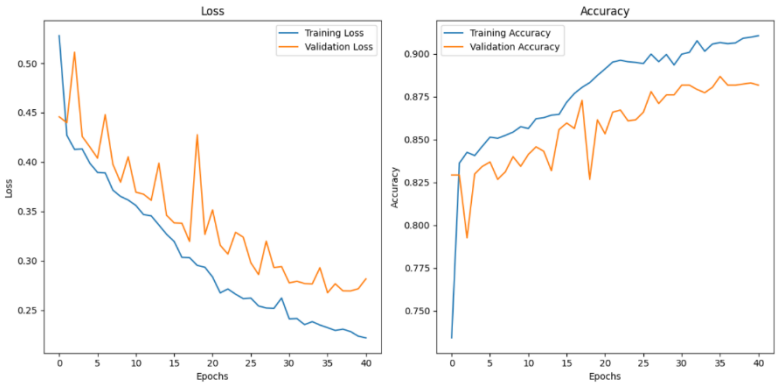


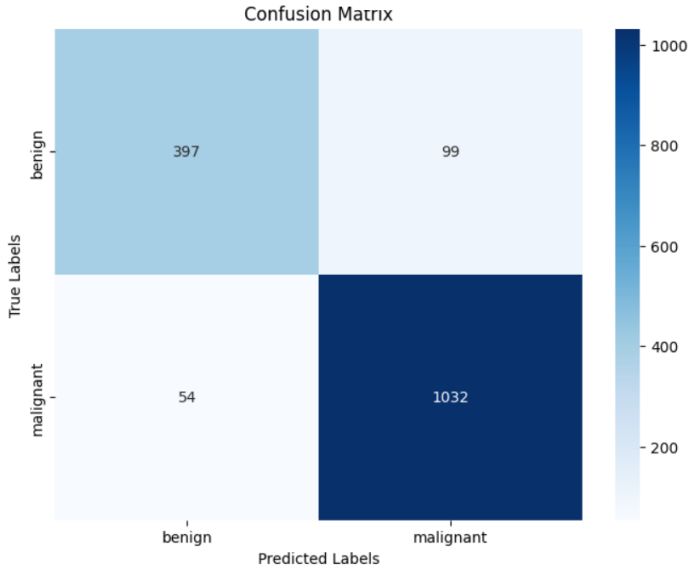
Figure 9: Grad-CAM visualization for the images where we had previously found black dots



(a)

	precision	recall	f1-score	support
0.0	0.88	0.80	0.84	496
1.0	0.91	0.95	0.93	1086
accuracy			0.90	1582
macro avg	0.90	0.88	0.89	1582
weighted avg	0.90	0.90	0.90	1582

(b)



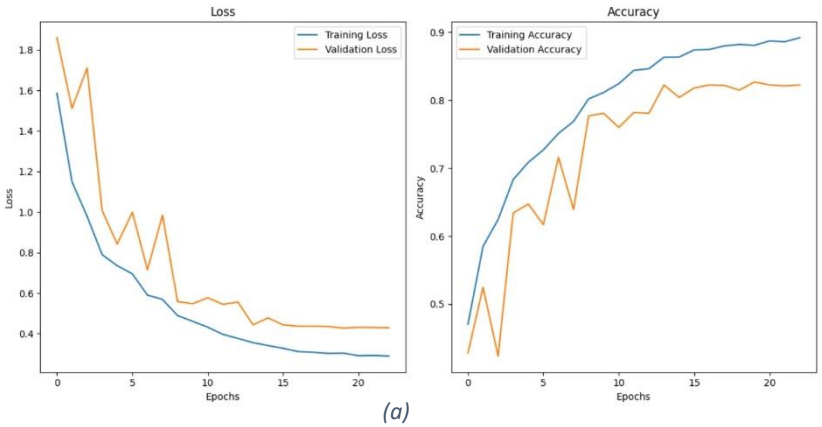
(c)

Layer	Filters/Units	Dropout
Conv Layer 1	40	N/A
Conv Layer 2	48	N/A
Conv Layer 3	16	N/A
Conv Layer 4	24	N/A
Conv Layer 5	32	N/A
Conv Layer 6	40	N/A
Dense Layer 1	176	0.1
Dense Layer 2	208	0.0
Dense Layer 3	48	0.1
Dense Layer 4	16	0.3
Dense Layer 5	80	0.2
Optimization Parameters		
Learning Rate	0.0001	
Optimizer	Adam	

(d)

Figure 10: Results from the best model in stage 1.

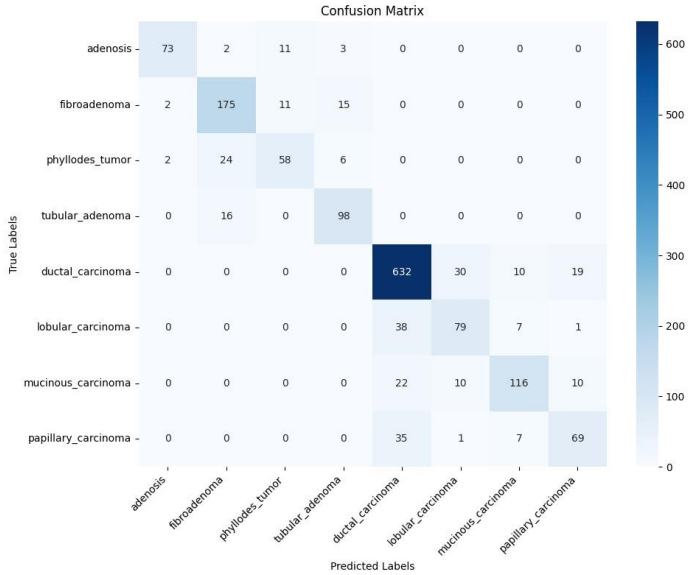
- *(a) Plots of the accuracy and loss on training and validation sets on the model training process*
- *(b) Classification report of the model*
- *(c) Confusion matrix of the model*
- *(d) Model's architecture*



(a)

	precision	recall	f1-score	support
0.0	0.85	0.92	0.88	89
1.0	0.82	0.82	0.82	203
2.0	0.70	0.69	0.70	90
3.0	0.87	0.83	0.85	114
4.0	0.88	0.90	0.89	691
5.0	0.66	0.62	0.64	125
6.0	0.77	0.73	0.75	158
7.0	0.68	0.69	0.68	112
accuracy			0.82	1582
macro avg	0.78	0.77	0.78	1582
weighted avg	0.82	0.82	0.82	1582

(b)



(c)

Layer	Filters/Units	Kernel Size/Dropout
Conv Block 1	48	2
Conv Block 2	32	5
Conv Block 3	16	5
Conv Block 4	80	5
Conv Block 5	80	5
Conv Block 6	48	5
Dense Layer 1	484	0.0
Dense Layer 2	484	0.2
Dense Layer 3	304	0.1
Dense Layer 4	364	0.2
CSV Layer 1	35	0.0
CSV Layer 2	66	0.1
Optimization Parameters		
Learning Rate	0.001	
Optimizer	Adam	

(d)

Figure 11: Results from the best model in stage 2.

(a) Plots of the accuracy and loss on training and validation sets on the model training process
(b) Classification report of the model
(c) Confusion matrix of the model
(d) Model's architecture