Cancer Image Classification Project Group 1

Dinis Fernandes nº20221848 Dinis Gaspar nº20221869
Inês Santos nº20221916 Luis Dávila nº20221949
Sara Ferrer nº20221947

Abstract

Even in a modern world with extremely impressive advances in a wide array of medical fields, cancer still kills millions worldwide every year. The biggest issue is that it's often detected too late and after it has spread to vital organs, drastically decreasing the likelihood of recovery. This is why advances in the field of cancer detection are needed, this is an area where Deep Learning can be extremely valuable. In this project we will attempt to develop Deep Learning models for the purpose of cancer detection from microscopic images of tumors.

1 Introduction

In this report, we aim to detail the process of obtaining models for cancer image classification for both binary (Benign or Malignant) and multiclass (Cancer Type) problems, using data from BreakHis dataset [1]. We will explain the different stages of our process with some visualizations for additional context.

2 Exploratory Data Analysis

As with any machine learning project, the first step is to obtain a good base of knowledge of the data, we use pandas to import our data in a dataframe format and produce visualizations using matplotlib and seaborn. In this case, the problem of missing values was not major, as they were only present in very few rows of the excel provided, and using the directory structure we could very easily solve the problem. We have a total of 7909 images.

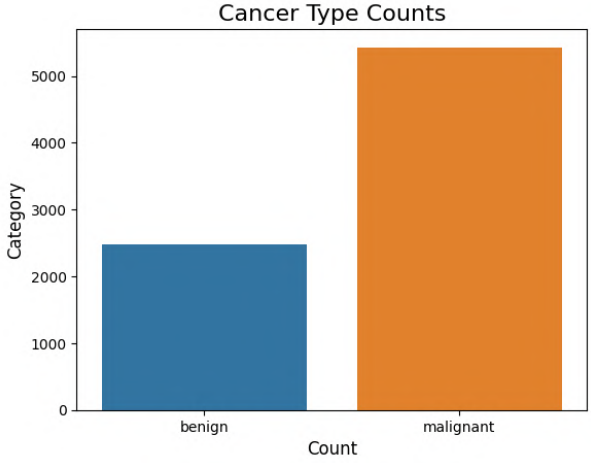

Figure 1: Count of Benign Vs Malignant.

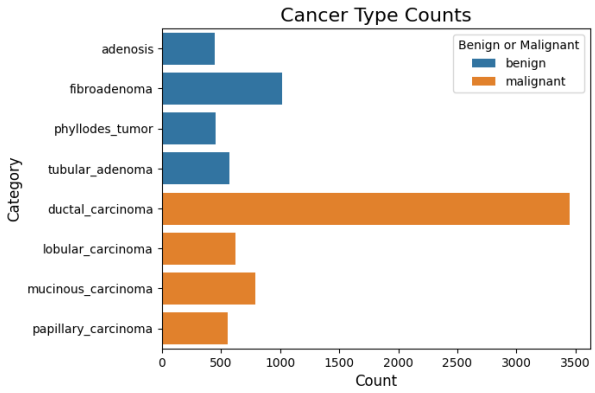

Figure 2: Count of Cancer Type.

As with any classification problem, it is crucial to check the class distribution in the dataset. In this case, we can see that both problems are clearly unbalanced. For this reason, techniques to mitigate this must be considered, such as, stratification, using class weight dictionaries during model training and using F1-score, which is built into Keras, for evaluation (in the case of equal F1-scores we will see the losses to compare results, since that is the metric that is going to be optimized).

We then performed an analysis comparing the different cancer types and the magnification of images, we could see that the malignant cancers (except mucinous carcinoma) had a strong bright pink tone, where benign cancers have less vibrant tones. Also the images had different magnitudes (40x, 100x, 200x and 400x) and different amounts of images for each magnitude, we will take that into consideration when we do the splits to maintain a balanced representation of classes and magnitudes across train, validation and test data.

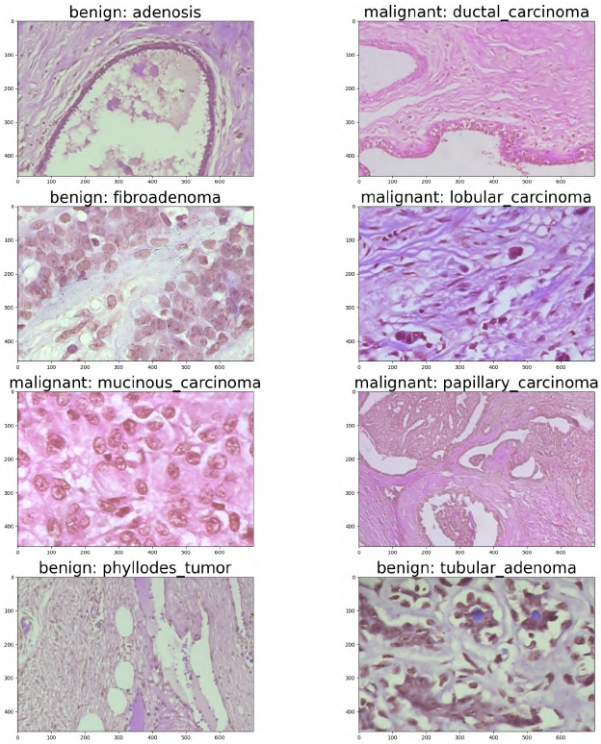

Figure 3: Cancer Types.

To input our images into the models, we must ensure that they are all of the same dimensions.

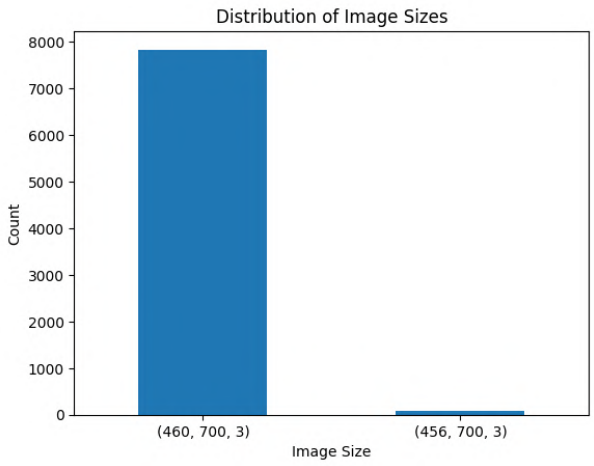

Figure 4: Image size distribution.

We can see that a very small number of images have a negligible difference in image size, this will be corrected by resizing images, which is needed for all images anyway, although no additional considerations are needed for these images as the difference is essentially irrelevant. We also tested resizing all images to a square shape (460x460, for testing purposes), and it seems that no information was lost, thus, we will always resize to square formats going forward. Finally, we checked the pixel value distribution across the original data.

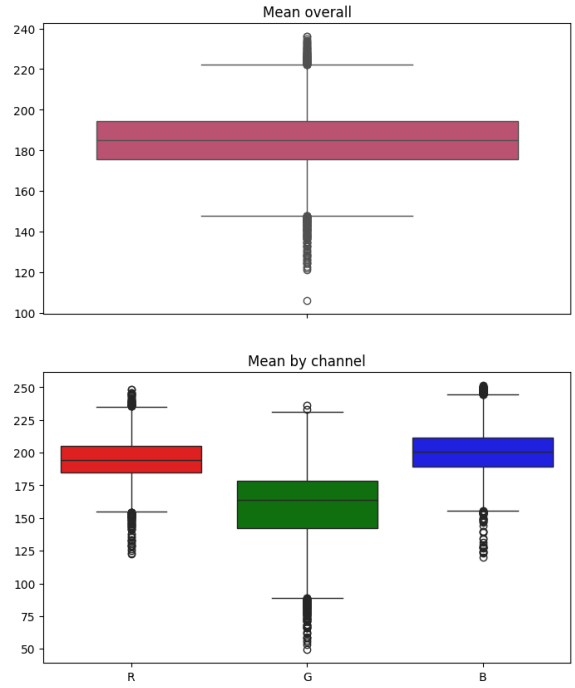

Figure 5: Outlier Detection.

We can see that there are not many outliers since the distribution using the mean and median leads to very similar results, and there is a clear dispersion in pixel representation, this should be good, as the differentiation across images should help models find and learn patterns.

3 Preprocessing

With a good understanding of the data having been obtained, it is now time to test different image processing methods, to do this we obtain an image for each of our cancer types so we can ensure our conclusions hold across them.

We tested methods for adjusting image brightness and contrast, image segmentation (K-means) and noise removal from the CV2 library [2] as well as the autocontrast method from the lmageOps module of the PIL library [3].

We found that image segmentation and noise removal does not have any positive impact on our images, as such they are not used. We were also able to conclude that extreme values for brightness and contrast basically delete the images, turning them completely white.

With this knowledge in mind, we performed a grid-search, using the ParameterGrid class from scikit-learn, for the preprocessing parameters (brightness and contrast settings, PIL autocontrast and even a gray colormap) using some simple models, relying on the assumption that bad data would yield worse results than decent data even on basic models. We also chose to resize all of our images to 150x150, because from some testing, even 100x100 or 200x200 provided problems (bad results and memory issues, respectively, and 150x150 still maintains an acceptable representation of the image Annex 1 - Figure 10). As the results for the test in the multiclass portion were so low (F1 score less than 0.1), they will be ignored and

the best combination from the binary model will be used for both problems going forward. This was a combination that only resized the images and did not actually transform them. Once we have our best models, we will once again look at preprocessing parameters.

To train and test our models did the split by stratifying on cancer type and magnitude (using a compound label), 20% (1582 images) went for test, the rest 80% (6327 images) went for train and validation, 80% (5061 images, 64% of the total) for train and 20% (1266 images, 16% of the total) for validation respectively.

4 Setting a benchmark

Before diving too deep into model development we will set ourselves some benchmark scores, we will then develop models attempting to beat these scores.

The benchmark model is a multi-output model developed using the Keras functional API, meaning it outputs predictions for both the binary and multiclass problems. Furthermore, the predictions of the binary classification part serve as input for the multiclass part. The model takes as input the image itself and its magnification level. The model also contains an inception module, which "aligns with the intuition that visual information should be processed at various scales and then aggregated so that the next stage can abstract features from different scales simultaneously" by performing different feature extraction methods in parallel. More detailed explanations can be found in the *Going deeper with convolutions* [4] article, which served as the base for our implementation. More details on our specific implementation can be found in the Benchmark.ipynb notebook.

As this is a benchmark, the results are the most important take-away, although we will leave a deeper analysis of the results for the results of our final model. We will present the classification report and confusion matrix for our test set for both problems, both were produced using scikit-learn and explanations can be found at Scikit-learn metrics [5]. The loss for this model is more weighted towards the multiclass problem, as it is more complex.

For the Binary problem, the result on the test set was an average F1-score of 0.85:

	Precision	Recall	F1-score	Support
Benign	0.75	0.87	0.81	496
Malignant	0.94	0.87	0.90	1086
Accuracy	-	-	0.87	1582
Macro Avg	0.84	0.87	0.85	1582
Weighted Avg	0.88	0.87	0.87	1582

Table 1: Benchmark Binary Classification Report.

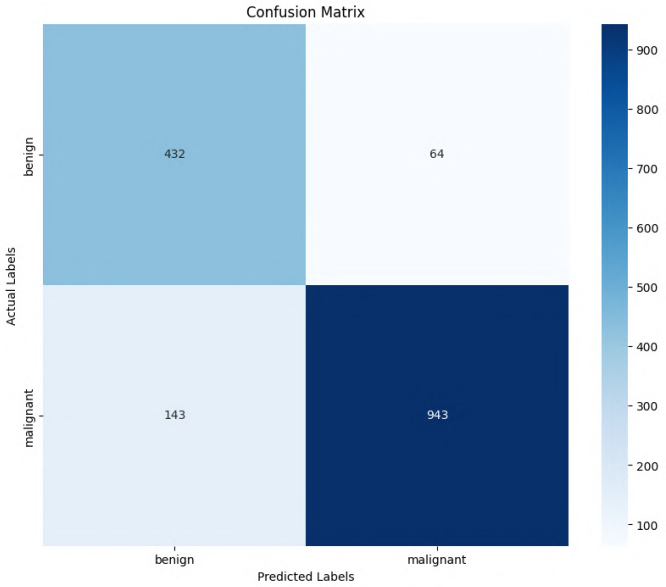

Figure 6: Benchmark Binary Confusion Matrix.

For the multiclass problem, the result on the test set was an average F1-score of 0.38:

	Precision	Recall	F1-score	Support
Adenosis	0.38	0.60	0.47	89
Ductal Carcinoma	0.82	0.44	0.58	691
Fibroadenoma	0.50	0.18	0.26	203
Lobular Carcinoma	0.27	0.72	0.39	125
Mucinous Carcinoma	0.55	0.21	0.30	158
Papillary Carcinoma	0.20	0.53	0.29	112
Phyllodes Tumor	0.19	0.29	0.23	90
Tubular Adenoma	0.42	0.66	0.52	114
Accuracy	-	-	0.43	1582
Macro Avg	0.42	0.45	0.38	1582
Weighted Avg	0.58	0.43	0.44	1582

Table 2: Benchmark Multiclass Classification Report.

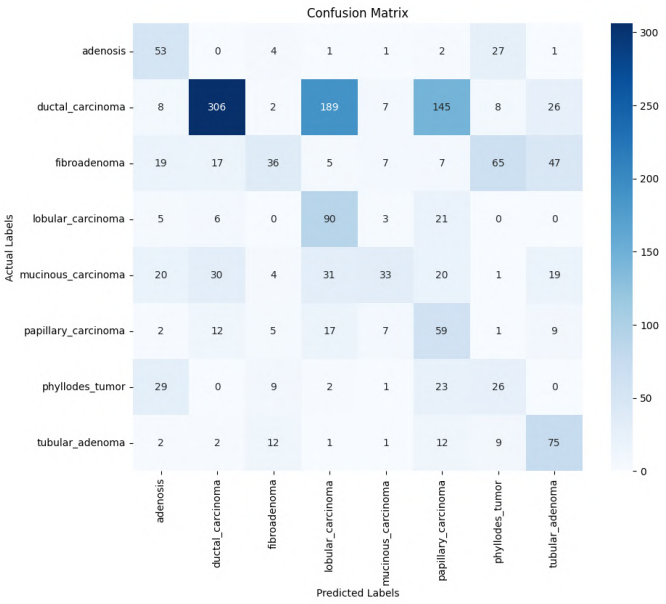

Figure 7: Benchmark Multiclass Confusion Matrix.

Note: Results will only be provided in the same format for our final models, as given the extensive testing we performed, showing all of the results would be tiresome.

5 Model Development

The process of model development is the exact same for binary and multiclass, and obviously performed in Keras for both. We start by obtaining an overfitting and an underfitting model, to try and find the boundaries of our hypothesis space, we then perform a gridsearch, using Keras tuner, attempting to find the ideal combination of parameters in that space; afterwards, evaluate that model with cross-validation (we can already say that these models did not achieve good results). Subsequently, we will attempt to use some pre-trained models to achieve better results, through some fine-tuning. Finally, we will experiment with including the magnification level as input for the model as well as using Image Data Generator. Image Data Generator did not improve results.

In the case of **binary classification**, as none of these techniques had better results than the benchmark model, we retrained only the benchmark model keeping only the binary part, which helped with convergence, as in the benchmark model, the multiclass portion has a lot more weight in the model loss, this led to an F1-score on the test set of 0.89. We were able to avoid overfitting with this model. Details on the final architecture can be found in the last section of the Binary_classification.ipynb notebook.

In the case of **multiclass classification**, we also try a different loss function (categorical focal crossentropy) trying to see if the problem of imbalance data could be solved in a better way, but it leads to even worse results, so as final implementation we use the normal categorical cross entropy loss. We were able to finetune the last convolution of a DenseNet201 model (trained on the ImageNet dataset), taking the magnification level as input, it had no overfitting and produced much better results than our benchmark, achieving an F1-score of 0.58 on the test set. Details on the final architecture can be found in the last section of the Multiclass_classification.ipynb notebook. To see all the intermediate results see the table in Annex 4 - Table 5 .

6 Experiments

As mentioned earlier, now that we have our final models we once again use a gridsearch (again, using the ParameterGrid class from scikit-learn) to test some more preprocessing parameters, this included some less extreme values for brightness and contrast settings, PIL autocontrast and a couple more colormaps (original (RGB), HSV (Hue-Saturation-Value), and the BGR) with the same resize configuration (150x150). Unfortunately, none of these tests improved our results.

7 Final Results

In regards to image processing, our final solution only resizes images to 150x150, without performing any image transformations. We will now look at the final results for both of our problems and analyze them a little bit.

7.1 Binary Problem

Our final binary model uses the inception module described in the benchmark section, taking the magnification level as input, but is a standalone model, meaning it does not have the multiclass part of the original model.

	Precision	Recall	F1-score	Support
Benign	0.81	0.89	0.85	496
Malignant	0.95	0.91	0.92	1086
Accuracy	-	-	0.90	1582
Macro Avg	0.88	0.90	0.89	1582
Weighted Avg	0.90	0.90	0.90	1582

Table 3: Final Model Binary Classification Report.

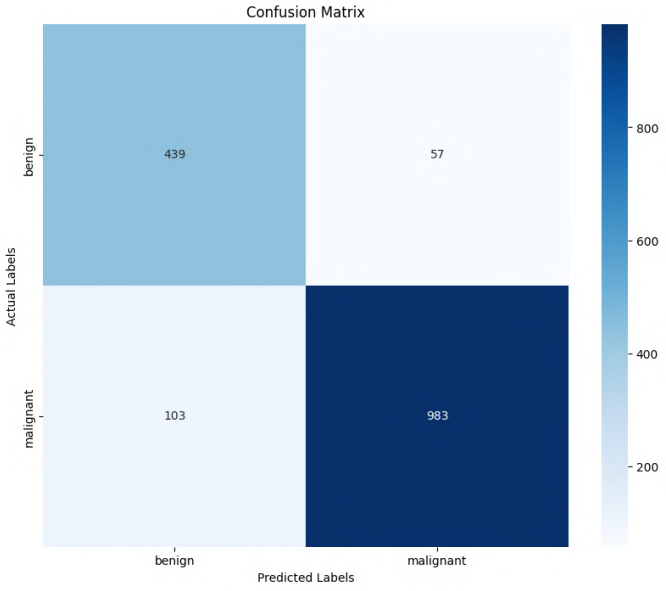

Figure 8: Final Model Binary Confusion Matrix.

Testing our best binary model on the test set, we achieved an F1 score of 0.89, and a weighted one of 0.90.

Obviously the weighted average is slightly higher, because, as expected, the class our model finds easiest to classify is the majority class (malignant), which inflates the weighted average when comparing to the macro average as it is $\tilde{2}$ times more common than the benign class.

Interestingly, the model predictions have very few false positives (only 57 cases of 1582, $\tilde{3}.6\%$ of cases), in a real context this is good since predicting a benign cancer as malignant can cause the patients to get a false, and much more severe, diagnosis.

On the other hand, the model predictions contain 103 false negatives, this is not ideal, as in a real context this means these malignant cancer cases would go undetected.

However, 103 out of 1582 is only $\tilde{6}.5\%$, which means this is not an extremely common scenario.

7.2 Multiclass Problem

Our best multiclass model is a fine tuned DenseNet201 model which takes the magnification level as input.

	Precision	Recall	F1-score	Support
Adenosis	0.53	0.70	0.60	89
Ductal Carcinoma	0.84	0.64	0.72	691
Fibroadenoma	0.58	0.64	0.61	203
Lobular Carcinoma	0.40	0.65	0.49	125
Mucinous Carcinoma	0.57	0.60	0.58	158
Papillary Carcinoma	0.43	0.56	0.49	112
Phyllodes Tumor	0.47	0.42	0.45	90
Tubular Adenoma	0.65	0.68	0.66	114
Accuracy	-	-	0.62	1582
Macro Avg	0.56	0.61	0.58	1582
Weighted Avg	0.66	0.62	0.63	1582

Table 4: Final Model Multiclass Classification Report.

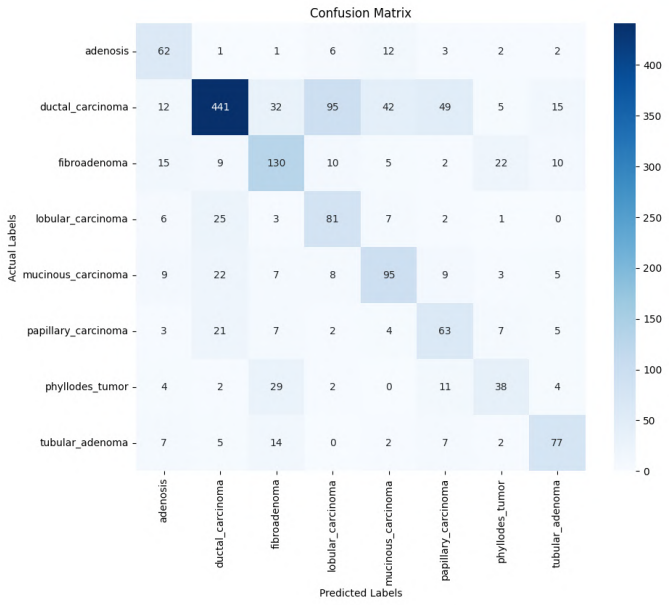

Figure 9: Final Model Multiclass Confusion Matrix.

Testing our best model on the test set, we achieved an F1 score of 0.58, and a weighted one of 0.63.

Obviously the weighted average is slightly higher, because, as expected, the class our model finds easiest to classify is the majority class (ductal_carcinoma), which inflates the weighted average when comparing to the macro average as it is $\tilde{3}.5$ times more common than the second most common cancer type (fibroadenoma).

Interestingly, the minority class (Adenosis) is one the classes with the highest F1 score, this is likely due to this cancer type presenting unique and distinct characteristics. On the other hand, the second least common class (phyllodes_tumor) achieves the lowest F1 score across all classes, despite being essentially as common as the class mentioned previously.

We clearly see that our model is having problems with the ductal carcinoma class, predicting it into lobular, mucinous and papillary carcinoma. These predictions make sense since ductal, lobular and papillary carcinomas have similar colors, going even deeper, in 400x ductal and papillary have very similar shapes, and the same occurs with ductal and mucinous carcinoma but in 100x. There is also a problem between phyllodes tumor and fibroadenoma, but they are not alike, so the error could be because of the existence of few samples of phyllodes tumor. All of these leads to predictions errors for not capturing the differences between classes so well Annex 2 - Figure 11.

8 Future work

We believe that with more time or computational resources, it would've been interesting and potentially beneficial to test more grid search combinations of preprocessing and network architecture, to unfreeze larger portions of the pre-trained models (in an attempt to specialize the models to a certain extent) or even to have a pre-trained model trained with cancer cell image data (trained by ourselves or with the weights saved to do transfer learning, something that we could find). Finally, having the images with a larger size to maintain more quality (min 300x300 images Annex 3 - Figure 12).

References

[1] BreakHis Dataset: Breast Cancer Histopathological Database. [Online]. Available: https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-breakhis/.

[2] OpenCV library documentation. [Online]. Available: https://docs.opencv.org/4.x/index.html

[3] ImageOps module of the Pillow (PIL) library documentation. [Online]. Available: https://pillow-wiredfool.readthedocs.io/en/latest/reference/ImageOps.html

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", 2014. Available: https://arxiv.org/pdf/1409.4842.

[5] Scikit-learn metrics documentation [Online]. Available: https://scikit-learn.org/1.5/modules/model_evaluation.html

9 Annex

Annex 1:

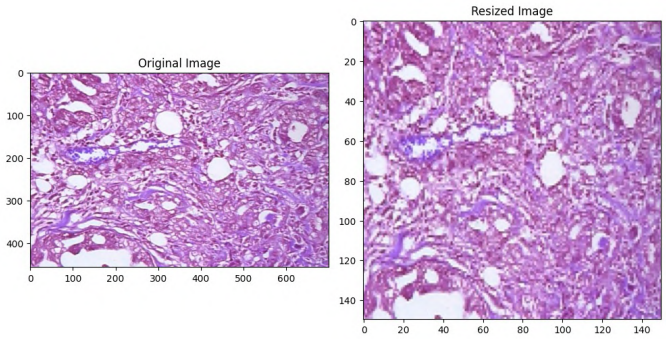

Figure 10: Resizing 150X150.

Annex 2:

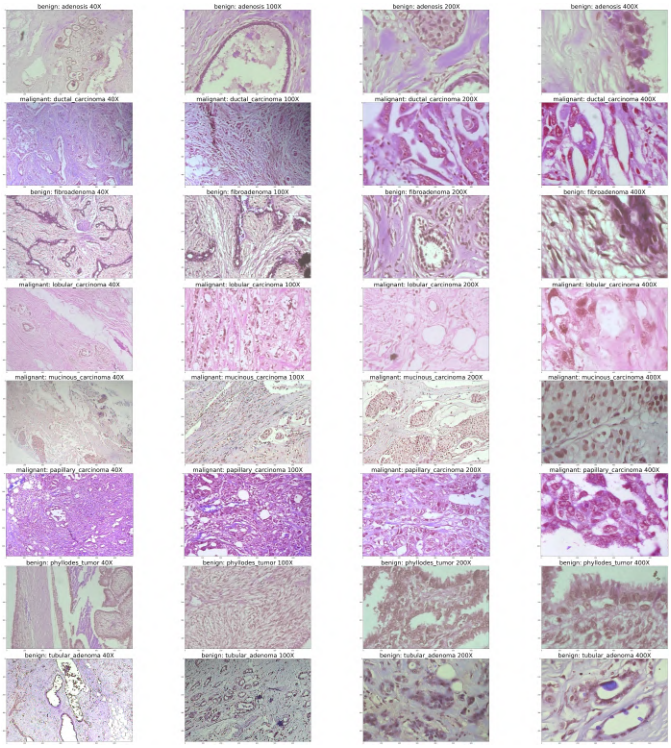

Figure 11: Cancer Types with Different Magnifications.

Annex 3:

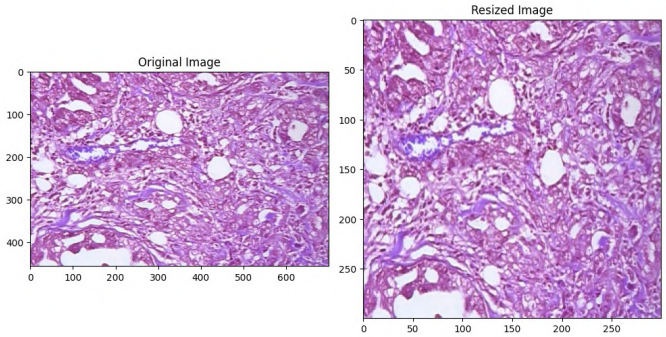

Figure 12: Resizing 300X300.

Annex 4:

F1-Scores	Validation		Test	
	Binary	Multiclass	Binary	Multiclass
Benchmark	0.88	0.37	0.85	0.38
Gridsearch CV	0.81	0.39	-	-
Best Pre-Trained Model(Densenet)	0.87	0.52	-	-
Best Model with Categorical Focal Crossentropy	-	0.45	-	-
Best Model with ImageData Generator	0.86	0.42	-	-
Best Model with Magnitude	0.87	0.55*	-	0.58*
Benchmark Model (Only Bianary Part)	0.91*	-	0.89*	-

* Final Model

Table 5: Model Scores