NOVA IMS
BSc in Data Science
Deep Learning 2024/2025

Breast Cancer Detection

Image Classification

Students:
Laura Matias (20221836)
Marta Aliende (20241453)
Marta Almendra (20221878)
Matilde Casimiro (20221940)
Teresa Simão (20221873)

Professors:
Mauro Castelli
Yuriy Perezhohin

1 Abstract

Breast cancer is a major health issue for women globally, and early detection is critical to improve patient outcomes. This study relies on deep learning methods to classify images of breast tumors, contributing to more automated diagnostic processes. Using the BreaKHis collection of high-resolution breast tissue microscopic images, the project includes:

- **Binary Classification:** Distinguishing between benign and malignant tumors.
- **Multi-Class Classification:** Identifying tumor types (Adenosis, Ductal Carcinoma, Fibroadenoma, Lobular Carcinoma, Mucinous Carcinoma, Papillary Carcinoma, Phyllodes Tumor, Tubular Adenoma).

Key steps include data pre-processing, image transformation, and the development of convolutional neural networks (CNNs). Methods like data augmentation, and transfer learning were also employed, to provide robust predictions and overall assist healthcare professionals in making faster and more reliable decisions.

2 Background

Several Python libraries were used, the most significant ones being **numpy** and **cv2** for data handling and image processing, and **tensorflow.Keras** used to create, train, and evaluate deep

learning models. **Kerastuner** was used for hyperparameter optimization, **seaborn** and **matplotlib** for visualizations and **pandas** for managing metadata and data manipulation.

3 Methodology

3.1 Metadata Preprocessing

During our initial exploration, we found and dealt with **missing data** found in 4 observations. Since the missing information was regarding the image's paths, we imputed those values manually. **Data transformation** techniques such as label encoding were used on the dataset. As a result, two columns were added to the dataset: one for binary classification, with numeric values for benign (0) and malignant (1) tumors, and one for multi-class classification, which converted cancer types labels into values from 0 to 7 for multi-class classification.
Next, we created multiple data visualizations, which allowed us to better understand the data we were dealing with. We concluded that the dataset was **imbalanced**, with the majority of cases corresponding to malignant tumors (Annexes Fig.1). There was also a significantly greater amount of Ductal Carcinoma tumor type among cancer types, however, this imbalance reflects real-life prevalence, as this is the most common type of breast cancer. When splitting the data, stratified sampling was used to preserve the original distribution of benign and malignant classes across the training and testing sets.

3.2 Image Preprocessing[3][4]

To prepare data for training and testing, we performed **resizing** of all images to a consistent size (50x50) and normalized pixel values.
We performed various color transformations with the goal of enhancing model robustness and performance. We experimented with **grayscale** (focus on intensity and simplify inputs) (Annexes Fig.2), **RGB scaling** (modifying the red, green, and blue components of an image to enhance its colors) (Annexes Fig.3), **contrast adjustment** (increasing contrast by brightening light areas and darkening dark areas) (Annexes Fig.4) and **Laplacian filtering** (highlighting regions of rapid intensity change for edge detection) (Annexes Fig.5). Duplicate images were found and eliminated, in order to prevent any possible data leakage. **Data augmentation** was also implemented to increase variability in the training dataset. This step was done with ImageDataGenerator from the Keras library, which performs image transformations to artificially expand the dataset, reducing overfitting, and improving model generalization.

3.3 Step 1: Binary Classification [2][5][6]

In this stage, the goal was to build a deep learning model to classify tumor images as either Benign or Malignant. Having this in mind, we experimented with Convolutional Neural Network (CNN) and transfer learning, using pre-trained models like **VGG16** and **ResNet50**, as these have been proven to work well with medical data.
We started by building a **Convolutional Neural Network** (CNN) from scratch, using binary cross-entropy as the loss function and the Adam optimizer, which we choose due to its adaptive learning rate properties. We also specified that the model should track the **Area Under the Curve** (AUC) for precision-recall (PR) curves during training. AUC-PR is particularly useful metric for imbalanced binary classification tasks because it focuses on the performance of the model across different decision thresholds. Then, we built on the base models VGG16 and ResNet50 with the same specifications. We also incorporated **Callbacks** to help improve training by monitoring validation loss and stopping early if the model stopped improving.
All the models were then tuned with **HyperBand**, a hyperparameter tuning algorithm designed to quickly find the best set of hyperparameters for a model. It combines bandit-based strategies with early stopping to allocate resources to promising hyperparameter configurations while discarding underperforming ones early in the process.
For the models built from scratch, we tested and tuned training with all the different preprocessing techniques mentioned in the section before and data augmentation. However, it was explicit right away that some color transformations performed worse, so, with the pre-trained models, which take longer to tune and train, we used only the original resised images and the images with RGB scaling. During training, we monitored the AUC-PR metric in the validation set, however, to compare all the models' performances, we considered the **F1 Score**, **Precision** and **Recall** of the predictions made on the test set. These metrics are particularly important given the imbalanced nature of our data and the high cost of a false negative (failing to identify a malignant tumor).

3.4 Step 2: Multi Class Classification [2][5][6]

Next, we extended the task at hand to a multi class classification model to determine the particular tumor type.
Having in mind the results obtained when building the binary model, we discarded the ResNest50 pre-trained model, and the Laplacian transformed images for this section. Applying the same approach as before, we used a a **CNN built from scratch**, a **pre-trained model** (VGG16) and a **functional API**, all with variations in terms of RGB scalling, contrast adjustments, and class weights, a very common technique used to balance the model's sensitivity to each class in an

imbalanced dataset. The weights for the eight classes, were calculated to be inversely proportional to the class frequencies, meaning that less frequent classes get higher weights.
The models had similar architectures as in binary classification, with the main difference being the use of sparse categorical cross-entropy loss, and the output layer modified to have eight neurons (one for each class) and a softmax activation function. The Functional API approach required a slightly different architecture, since this model takes as input not only the images, but also the binary labels, using both to try to classify the tumor type.
Callbacks were also used, effectively improving training by incorporating early stopping. Once again, we explored the hyperparameter space and selected the best parameters for each model using **HyperBand Tuner**. The AUC-PR metric defined in the keras library is not made to deal with multiple classes, so in this part of the project, the metric we monitored was **accuracy** during the validation and tuning, and once again, the F1 Score, Recall and Precision when testing.

4 Results

4.1 Binary Classification Results [7]

We analyzed the results of the binary classification and observed that the models with the highest validation AUC were the ones without scaling, the model with contrast adjustment, and the VGG16 RGB model, while the grayscale model had the highest validation loss (Annexes Fig.6). Regarding test scores, the basic **RGB scaled model** had the highest weighted F1-score (of 85 percent), recall (86 percent), and precision (85 percent), making it our final choice for **binary classification** (Annexes Fig.7). Training loss gradually decreased with oscillations, whereas validation loss showed some variances in results. The AUC showed constant improvement with slight oscillations, having validation always superior to training scores, proving great results on both training and validation data.

4.2 Multiclass Classification Results [7][8]

Regarding **multiclass classification**, our model tuning exploration revealed that basic models built from scratch were more robust. However, when paired with the class weights, they showed moderately low accuracy and notable loss. Models with RGB scaling had similar results to the previously mentioned, though with a somewhat larger loss. The model combining scaling and class weights worked poorly, with loss outweighing accuracy (Annexes Fig.8). Lastly, the model variations where contrast was adjusted, did not stand out from the rest in terms of accuracy and lost. Regarding VGG16, we found that the RGB scaled outperformed the unscaled one, obtaining

higher accuracy and lower loss. Lastly, we noticed that all API variations outperformed the previous models, having a **Functional API with contrast adjustment** provide the best metrics, with a weighted F1 score of 58 percent, a recall of 63 percent and a precision of 59 percent, consequently being our chosen model (Annexes Fig.9). Training loss decreased steadily, whereas validation loss decreased only with few oscillations, indicating good generalization. Moreover, accuracy improved steadily, with validation accuracy always similar to training.

5 Discussion

5.1 Error Analysis

With only 86 incorrect diagnoses for malignant tumors, the binary model performs well, although it has more difficulty with benign tumors, misclassifying 137 as malignant (Annexes Fig.11). Due to class imbalance or greater learnt features for malignant patterns, this suggests a slight bias towards predicting malignant tumors. While an unbiased model is ideal, a bias toward malignant predictions is more acceptable because detecting malignant tumors is more important for patient outcomes.

The multi-class classification results can be explained by some of the model's difficulties with the data. In fact, Ductal Carcinoma produces a significant number of misclassifications in other categories, and the same can be said for Fibroadenoma. On the other hand, Phyllodes Tumor is rarely correctly classified, with only 9 correct classifications and 92 misclassifications (Annexes Fig.13). Overall, these errors highlight the challenges of class imbalance and overlapping features in the dataset.

5.2 Future Work

Given the imbalanced data in both the multi-class and binary classification tasks, the model tended to learn more from certain cancer types and diagnostics. Had more time been available, we would have explored ensemble methods, such as stacking predictions, to combine outputs from several models into a robust framework. Additionally, we would experiment with more complex base models, that are more time-consuming. These two strategies could improve generalization and reduce bias towards more frequent classes. Furthermore, we would also validate the model on external breast cancer datasets, to ensure that the model generalizes well to new, unseen data from different sources of images.

6 Bibliography

[1] *Cheng, J., Ni, D., Chou, Y., Qin, J., Tiu, C., Chang, Y., Huang, C., Shen, D., Chen, C. (2016). Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans. Scientific Reports, 6(1).*
https://www.nature.com/articles/srep24454

[2] *Jorgecardete. (2024, February 17). Convolutional Neural Networks: A Comprehensive guide. Medium.*
https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175

[3] *OpenCV: OpenCV modules. (n.d.).*
https://docs.opencv.org/4.x/

[4] *TensorFlow: tf.keras.preprocessing.image.ImageDataGenerator, by the TensorFlow team.*
https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

[5] *Team, K. (n.d.). Keras documentation: Hyperband Tuner.*
https://keras.io/keras_tuner/api/tuners/hyperband/

[6] *Kingma, D. P., Ba, J. (2014, December 22). Adam: A method for stochastic optimization. arXiv.org.*
https://arxiv.org/abs/1412.6980

[7] *Kumar, S. (2024, November 26). Metrics to Evaluate your Classification Model to take the Right Decisions. Analytics Vidhya.*
https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

[8] *Metrics for Multi-Class Classification: an Overview. (n.d.).*
https://ar5iv.org/html/2008.05756

7 Annexes

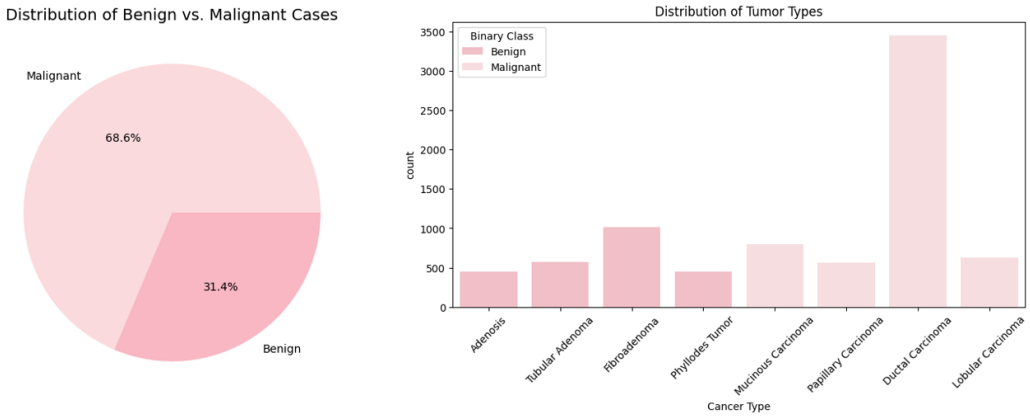

Figure 1: Binary and multi classes distributions

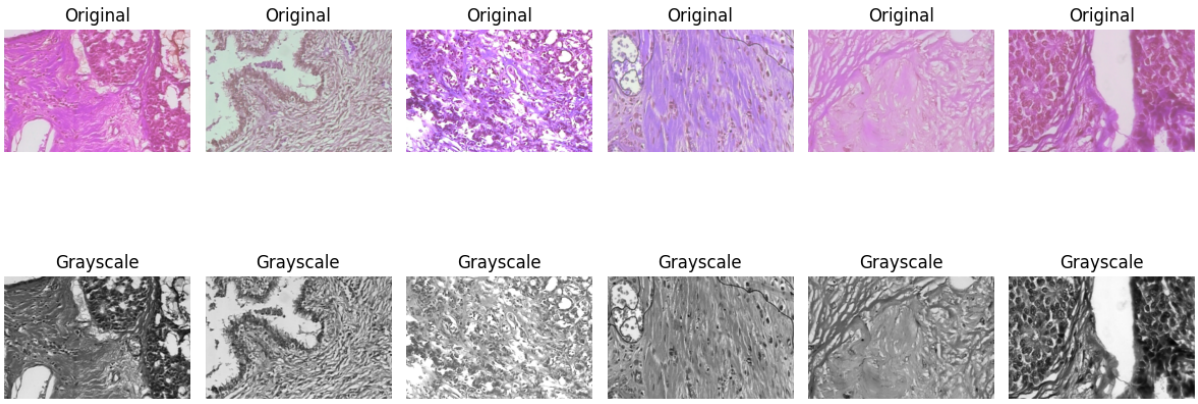

Figure 2: Gray Scaling

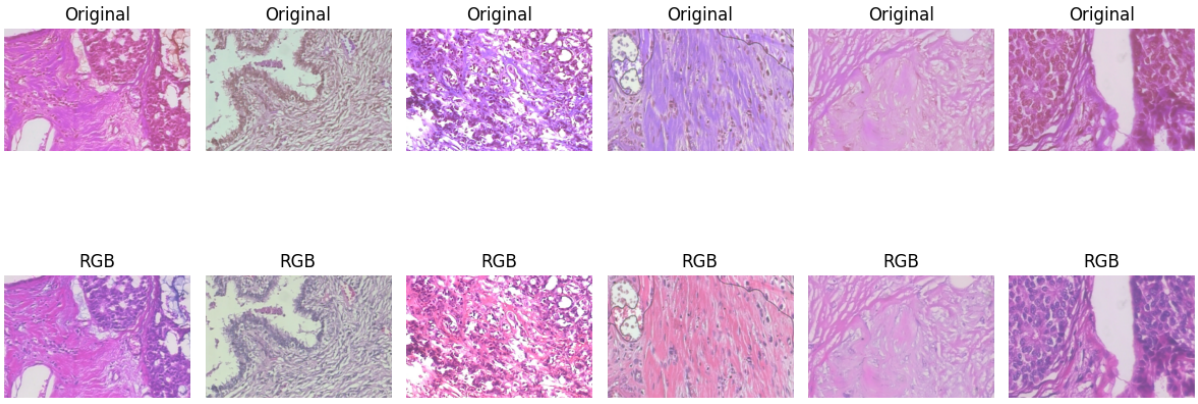

Figure 3: RGB Color Transformation

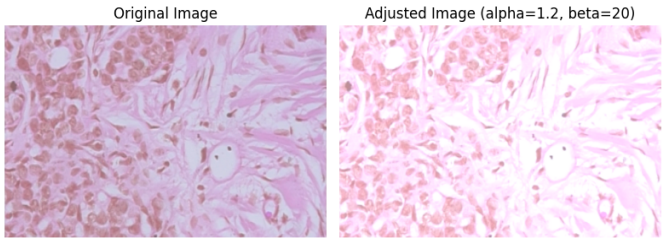

Figure 4: Contrast Adjustment

Figure 5: Laplacian Transformation

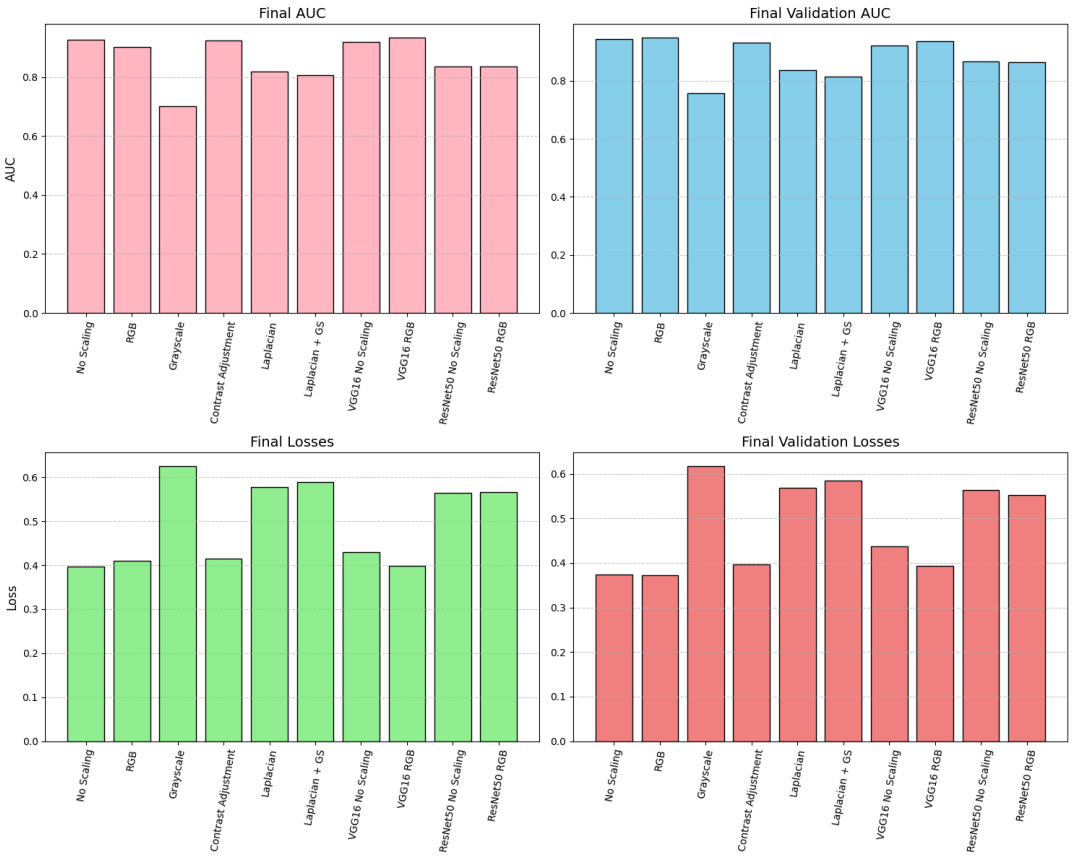

Figure 6: Binary Models Validation Scores

Figure 7: Binary Models Test Scores

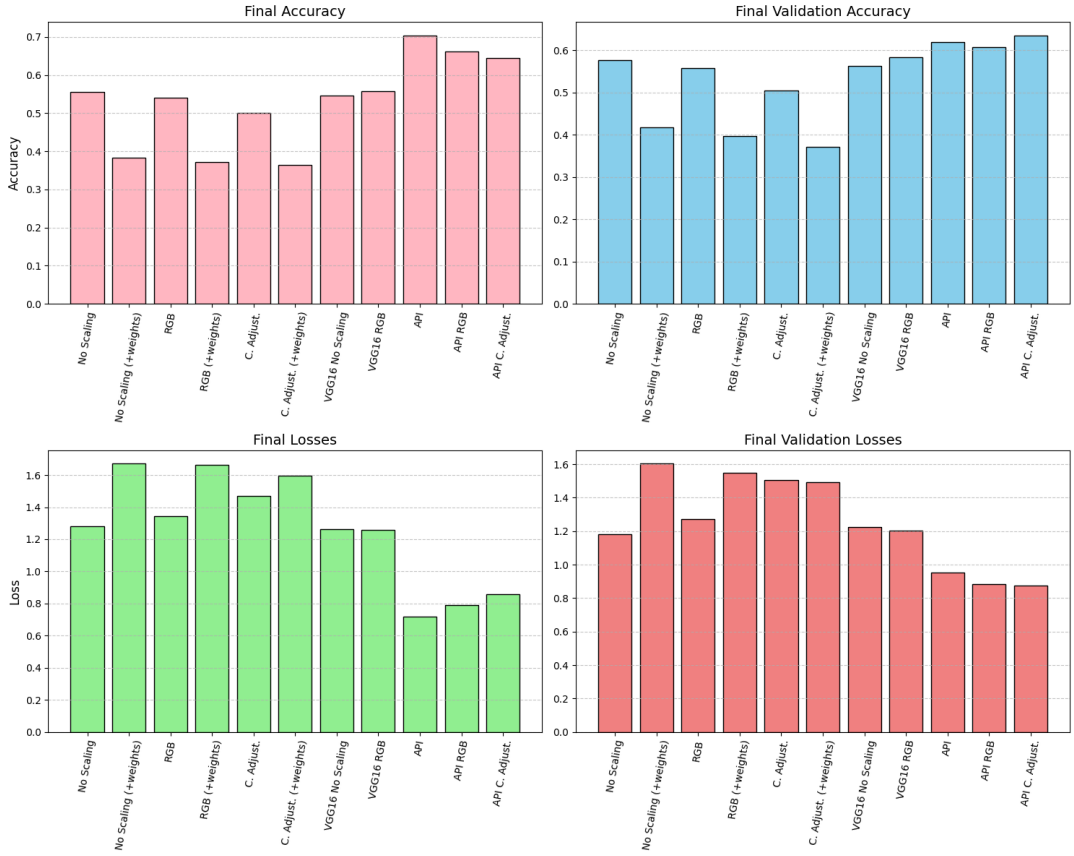

Figure 8: Multi Class Models Validation Scores

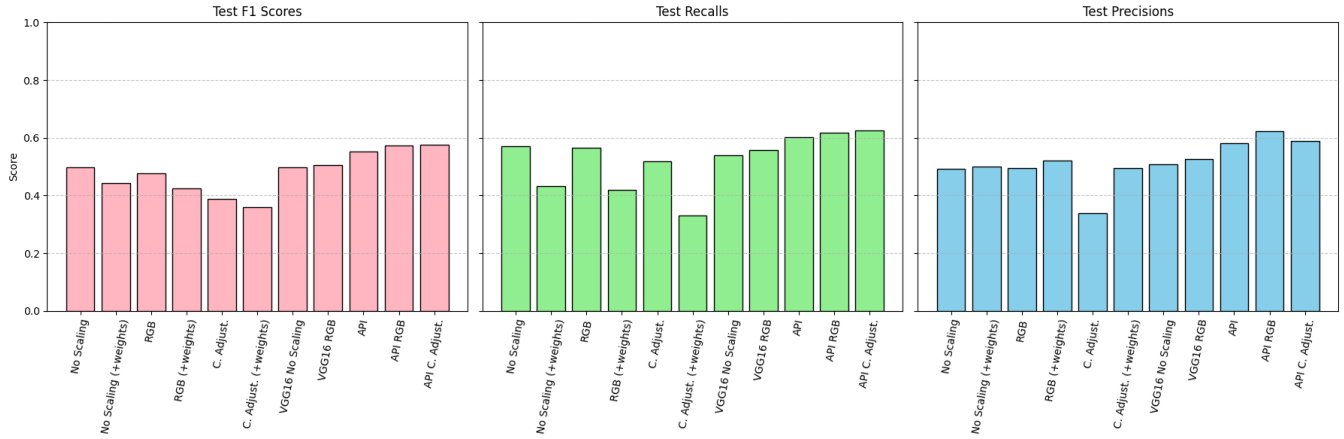

Figure 9: Multi Class Models Test Scores

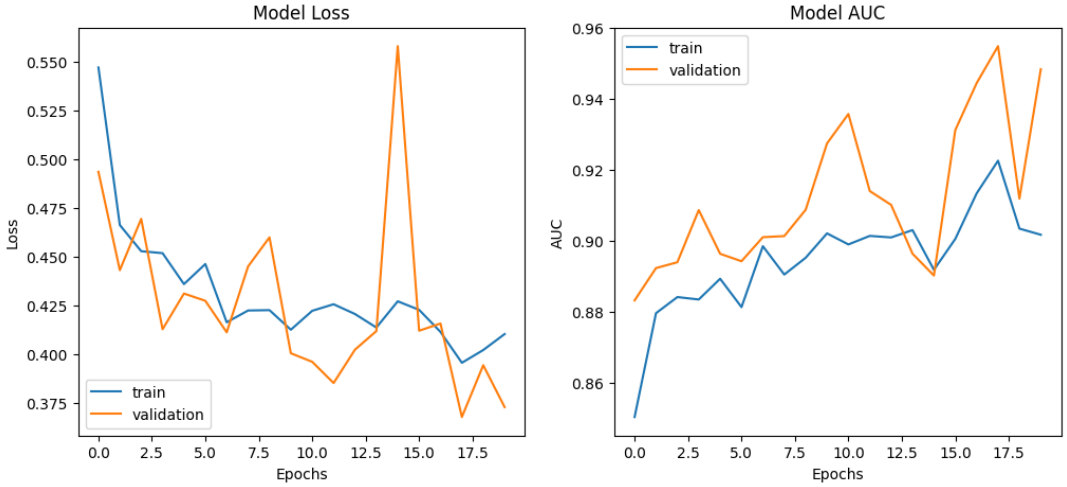

Figure 10: Binary Model Training History

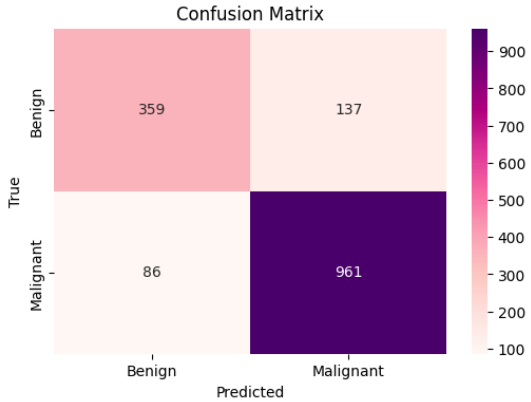

Figure 11: Binary Model Confusion Matrix

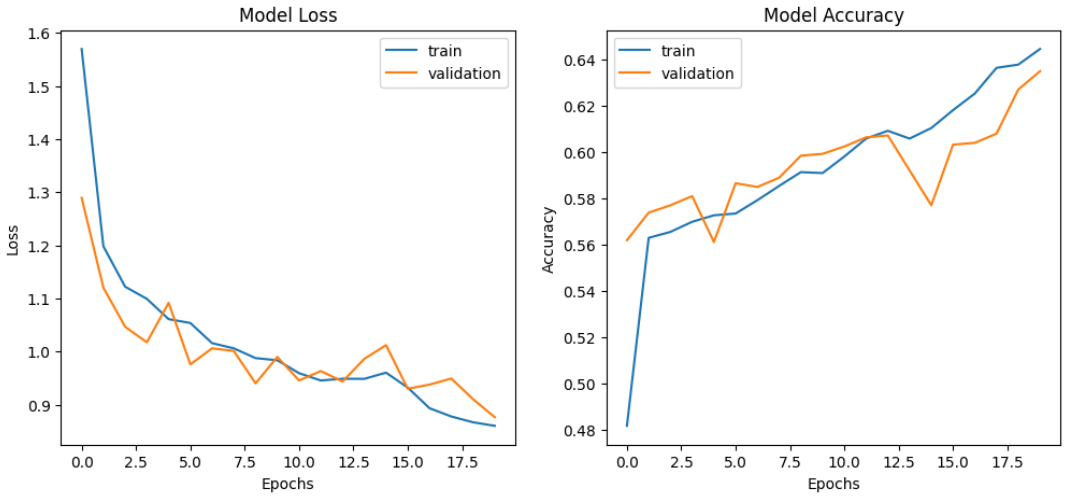

Figure 12: Multi Class Model Training History

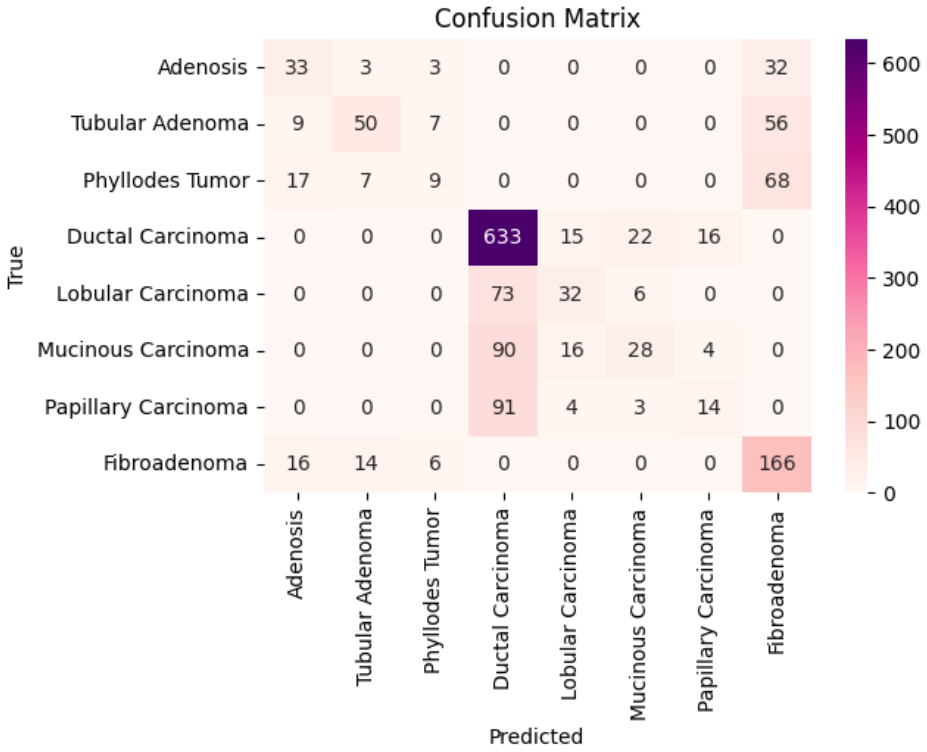

Figure 13: Multi Class Model Confusion Matrix